



Bioestadística en R **para ciencias de la salud**

Guía práctica

Edsaúl Emilio Pérez Guerrero
coordinador



ALFABÉTICA



Rector General

Ricardo Villanueva Lomelí

Vicerrector Ejecutivo

Héctor Raúl Solís Gadea

Secretario General

Guillermo Arturo Gómez Mata

**Coordinadora General Académica
y de Innovación**

María Esther Avelar Álvarez



**Rector del Centro Universitario
de Ciencias de la Salud**

José Francisco Muñoz Valle

Secretaria Académica

Beatriz Verónica Panduro Espinoza

Secretaria Administrativa

Saralyn López y Taylor

**Directora de la División de Disciplinas
Básicas para la Salud**

Norma Alicia Ruvalcaba Romero

Director de la División de Disciplinas Clínicas

Eduardo Gómez Sánchez

**Directora de la División de Disciplinas
para el Desarrollo, Promoción
y Preservación de la Salud**

Gabriela Macedo Ojeda

Coordinador de Innovación Educativa y Calidad

Juan Francisco Flores Bravo

COMITÉ EDITORIAL CUCS

Presidente

José Francisco Muñoz Valle

Secretaria

Edith Oregon Romero

Secretarios Técnicos

Alexis Missael Vizcaino Quirarte
Ma. de los Ángeles Villegas

**Directora de la Colección
de Publicaciones Científicas**

Juliana Marisol Godínez Rubí

Directora de la Colección de Difusión

Georgina Vega Fregoso

Directora de la Colección

de Libros de Texto
Alma Marina Sánchez Sánchez

Director de la Colección de Divulgación

Francisco Javier Turrubiates Hernández

**Directora de Comisión de Análisis,
Impacto y Relevancia Científica,
Curricular, Social e Institucional**
Beatriz Verónica Panduro Espinoza

Profesor Dictaminador Invitado Externo

Iván Manuel Girón Pérez

Bioestadística en R

para ciencias de la salud

Guía práctica

Edsaúl Emilio Pérez Guerrero
coordinador



ALFABÉTICA

La presente obra cuenta con el dictamen
No. CUCS/CE/0006/24, del Comité Editorial del
Centro Universitario de Ciencias de la Salud

ALFABÉTICA

D.R. © 2024, Editorial Innovación

Educativa S. A. de C. V.

Lábaro Patrio 264, Conjunto Patria
45150, Zapopan, Jalisco

Primera edición, 2024

ISBN 978-607-5894-30-0

Coordinador

Edsaúl Emilio Pérez Guerrero

Autores

- › Edsaúl Emilio Pérez Guerrero
- › Alberto Beltrán Ramírez
- › Miryam Rosario Guillén Medina
- › Daniel Osmar Suárez Rico
- › Ana Gabriela Flores Vargas
- › Jesús Jonathan García Galindo

Noviembre de 2024

Agradecemos el invaluable apoyo para la publicación
de este libro a la Universidad de Guadalajara, institución
referente en la promoción de la educación y la cultura.

Impreso y hecho en México / *Printed and made in Mexico*



Este trabajo está autorizado bajo la licencia Creative Commons Atribución-NoComercialSinDerivadas 4.0 Internacional (CC BY-NC-ND) lo que significa que el texto puede ser compartido y redistribuido, siempre que el crédito sea otorgado al autor, pero no puede ser mezclado, transformado, construir sobre él ni utilizado con propósitos comerciales. Para más detalles consúltese <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

Índice

Prólogo	11
CAPÍTULO 1	
Introducción a R y RStudio	13
CAPÍTULO 2	
Laboratorio de R 1	26
CAPÍTULO 3	
Introducción a la Bioestadística	49
CAPÍTULO 4	
Laboratorio de R 2	62
CAPÍTULO 5	
Población y muestreo	80
CAPÍTULO 6	
Laboratorio de R 3	100
CAPÍTULO 7	
Laboratorio de R 4	108
CAPÍTULO 8	
Laboratorio de R 5	119

CAPÍTULO 9

Laboratorio 6 de R	134
--------------------	-----

CAPÍTULO 10

Visualización de datos	144
------------------------	-----

CAPÍTULO 11

Estadística descriptiva	205
-------------------------	-----

CAPÍTULO 12

Distribuciones de probabilidad	251
--------------------------------	-----

CAPÍTULO 13

Distribución normal	279
---------------------	-----

CAPÍTULO 14

Introducción a las pruebas de hipótesis	317
--	-----

CAPÍTULO 15

Pruebas de hipótesis para la comparación de una media	341
---	-----

CAPÍTULO 16

Pruebas de hipótesis para la comparación de variables cuantitativas en dos grupos	377
--	-----

CAPÍTULO 17

Pruebas de hipótesis para la comparación de una variable cuantitativa en más de dos grupos	427
---	-----

CAPÍTULO 18

Pruebas post hoc	447
------------------	-----

CAPÍTULO 19	
Pruebas de hipótesis para la comparación de proporciones	492
CAPÍTULO 20	
Medidas epidemiológicas	511
CAPÍTULO 21	
Correlación	546
CAPÍTULO 22	
Regresión lineal	567
CAPÍTULO 23	
Regresión logística	599
CAPÍTULO 24	
Curvas ROC	622
CAPÍTULO 25	
Análisis de los componentes principales	651
CAPÍTULO 26	
Análisis factorial	671
CAPÍTULO 27	
Introducción al análisis de clústeres	692
CAPÍTULO 28	
Análisis de sobrevida	713
Semblanzas	745

A Lucas, Emiliano y Ale
A mis alumnos

Prólogo

Este libro surgió de la idea de contar con un compendio de ejercicios prácticos para mis alumnos de posgrado en Ciencias de la Salud. Sin embargo, poco a poco evolucionó hasta convertirse en esta guía, cuyo objetivo principal es facilitar la implementación de R en Bioestadística. Este trabajo cubre la necesidad de un recurso accesible y práctico que permita a estudiantes y profesionales de las ciencias de la salud dominar las herramientas bioestadísticas básicas necesarias para analizar datos de manera efectiva. A lo largo de los años, he observado que muchos encuentran desafiante integrar el análisis estadístico con la programación en R, y esta obra busca cerrar esa brecha. La programación, el análisis estadístico y las disciplinas de las ciencias de la salud no tienen por qué ser contrarias entre sí; de hecho, cada día están más interconectadas.

Este libro está dirigido a estudiantes de pregrado y posgrado en Ciencias de la Salud, investigadores y profesionales que deseen aprender a utilizar R para el análisis de datos. No se requiere experiencia previa en programación. El enfoque de este trabajo está en la aplicación de R en Bioestadística, centrándose en los conceptos y técnicas más utilizados en la investigación en las ciencias de la salud.

La estructura del libro sigue una progresión lógica, comenzando con los fundamentos de R y Bioestadística, y avanzando hacia técnicas más complejas, como la regresión logística y el análisis de componentes principales. Cada capítulo incluye ejemplos prácticos y ejercicios diseñados para consolidar el aprendizaje. Dentro del libro, encontrará algunos capítulos titulados “Laboratorio de R”, dedicados exclusivamente al aprendizaje del lenguaje R. Estos conocimientos serán útiles para resolver otros ejercicios que se encuentran en distintos capítulos.

Algunos códigos, sobre todo en los capítulos avanzados, son extensos, por lo que le recomendamos que copie y ejecute el bloque de código en su computadora cuando sea necesario. La versión disponible en línea facilitará este proceso, ya que cada código puede ser copiado con un solo clic. En este enlace usted podrá revisar el código de los ejercicios y ejemplos de este libro: <https://ed-perez-g.github.io/Codigo-Bioestadistica-R-Cs-Salud/>. La intención es que pueda copiar de manera más eficiente y sin errores el código y reproducirlo, aunque también puede copiar directamente el código desde este archivo.

Por otro lado, los ejercicios y ejemplos han sido diseñados utilizando diversas bases de datos. Algunas deberán construirse empleando R, otras se descargarán y algunas se adjuntarán utilizando librerías de R. Si bien en ocasiones utilizamos bases de datos libres y disponibles en internet, en otras empleamos bases de datos propias. Sin embargo, todos los ejemplos y ejercicios son originales. Si bien puede descargar las bases de datos utilizando los links que se encuentran a lo largo del libro, hemos creado un repositorio de GitHub donde estos archivos se encuentran alojados de forma permanente. Tenga este link a la mano: <https://github.com/Ed-Perez-G/Bases-Libro-Bioestadistica-Cs-Salud/tree/4734aba2f45337a23057074839bd3078e369ed2c/Bases>.

Este libro no hubiera sido posible sin la ayuda de muchas personas: agradezco a mis alumnos de posgrado por haber inspirado la creación de este libro. A Miryam, Ana, Katherine, Javier, Juan y Ricardo, quienes ayudaron en la revisión de los ejercicios y ejemplos de este libro. Agradezco especialmente a Katherine, quien fue la mente creativa detrás de muchas de las ilustraciones de este libro. A los otros autores de este libro: Alberto, Miryam, Ana, Jonathan y Daniel. Gracias por su compromiso y dedicación. Finalmente, agradezco a mi familia por su apoyo incondicional y por los sacrificios que hicieron para que este libro se volviera realidad.

Edsaúl Emilio Pérez Guerrero

Introducción a R y RStudio

Edsaúl Emilio Pérez Guerrero

En este capítulo se abordará una primera introducción a R y RStudio, dos herramientas esenciales para el análisis estadístico y la visualización de datos en ciencias de la salud. Comenzaremos explorando qué es R, posteriormente discutiremos por qué R es una opción excelente para los profesionales y estudiantes. Además, enlistaremos algunas de las múltiples aplicaciones de R en diversas áreas.

También cubriremos la instalación de R en diferentes sistemas operativos y presentaremos RStudio, un entorno de desarrollo integrado (IDE) que facilita enormemente el trabajo con R. Exploraremos la estructura de RStudio y cómo aprovechar sus funcionalidades para mejorar nuestra productividad en la programación y el análisis de datos.

Al finalizar este capítulo, usted debería tener una clara comprensión de las capacidades y ventajas de R y de sus aplicaciones en las ciencias de la salud.

¿Qué es R?

R es un lenguaje y un entorno de programación orientado a objetos creado especialmente para estadística y a la creación de gráficos. R es una derivación del lenguaje S que fue desarrollado en Bell Laboratories (antes AT&T) por John Chambers y sus colegas. Sin embargo, R se distribuye bajo una licencia GNU (General Public License) por lo que puede utilizarse sin costo.

R es un lenguaje estadístico extremadamente poderoso, ya que provee una amplia variedad de técnicas estadísticas entre las que se incluyen: modelado lineal y no lineal, pruebas estadísticas clásicas, estimaciones bayesianas, todo tipo de regresiones, análisis de series de tiempo, clasificación y agrupamiento, entre otros. Además, R se utiliza ampliamente para la creación de gráficos y en otras disciplinas, como *machine learning* y *deep learning*. Puede ver algunos ejemplos de gráficos creados en R [aquí](#). Por si esto no fuera poco, en R usted fácilmente podría hacer presentaciones, reportes, análisis bioinformáticos, análisis genómicos, *dashboards*, entre otras cosas. Incluso, escribir un libro como este. La verdadera pregunta aquí es: ¿hay algo que R no pueda hacer?

¿Por qué usar R?

Aunque existen múltiples razones para utilizar R, los autores de este libro nos enfocaremos en las que consideramos más importantes:

1. **R se distribuye de forma gratuita.** R es un lenguaje gratuito, se puede instalar, usar, actualizar y modificar libremente. R es gratis para todos, se puede instalar en distribuciones de Linux, MacOS y Windows. Incluso, algunos entusiastas han desarrollado una versión de R que puede utilizarse en exploradores web (puede ver un poco más de webR [aquí](#)).
2. **Actualizaciones continuas.** Comparado con otros paquetes estadísticos R se actualiza gratuitamente y con mayor frecuencia. Incluso, a diferencia de otros software, en R se brinda soporte prácticamente para todas las versiones, lo cual permite utilizar versiones antiguas y nuevas sin ningún problema.
3. **R es un lenguaje muy utilizado.** En el contexto académico y en la investigación biomédica y de ciencias de la salud, R se está convirtiendo en un estándar. Además, R se utiliza con mucha frecuencia en las universidades y algunas de las instituciones académicas más prestigiosas, como [Stanford](#), [Harvard](#), [Massachusetts Institute of Technology](#) y [Cambridge](#), cuentan con cursos específicamente diseñados para aprender R. En PUBMED utilizando los términos de búsqueda ([r software](#)) OR ([r project](#)) podemos

observar un crecimiento importante de los artículos indexados en esta base de datos. La cantidad de artículos publicados en PUBMED pasó de 3 846 en 2018 a 13 285 en 2023. Además, si se refinan más los términos de búsqueda y nos enfocamos a Ciencias de la Salud utilizando los siguientes términos de búsqueda (([r software](#)) OR ([r project](#))) AND ([medicine](#) OR [biology](#) OR [public health](#) OR [genetics](#) OR [biomedicines](#) OR [pharmacology](#) OR [nutrition](#) OR [genomics](#) OR [molecular biology](#)) la cantidad de artículos pasa de 4329 en 2018 a 9299 en 2023. En cambio, con estos mismos términos de búsqueda en Google Scholar podemos encontrar cerca de [404 000](#) y [228 000](#), para el año 2023.

4. **Res un lenguaje con excelente soporte.** A pesar de ser un lenguaje gratuito, R cuenta con una comunidad académica y de entusiastas encargada del desarrollo, de crear tutoriales, resolver dudas y de invertir gran parte de su tiempo en ayudarnos. Jamás estará solo mientras aprenda R. Resolver cualquier duda de R es muy sencillo. Solo basta con realizar una búsqueda en internet y encontrar un tutorial, video o blog que nos ayude. Algunos de los foros más importantes para la resolución se pueden encontrar en [StackOverflow \(SO\)](#) o [R-bloggers](#). Por otro lado, inteligencias artificiales como ChatGTP o Copilot han sido entrenadas con una gran cantidad de documentación sobre R, de tal manera que nos pueden ayudar con cualquier complicación relacionada con este lenguaje de programación. Algo tan sencillo como copiar un error generado al utilizar R en estas inteligencias artificiales puede ahorrarnos mucho tiempo y resolernos en la mayoría de los casos nuestro error.
5. **Relacionado con estadística, R puede hacer prácticamente todo.** Dado que R es un lenguaje específicamente diseñado para estadística, permite hacer prácticamente todo lo relacionado a ella. R se puede utilizar para estadística frecuentista, bayeseiana, para hacer modelos de predicción, manipulación efectiva de grandes cantidades de datos, entre otros.
6. **R es ideal para las ciencias de la salud.** R es perfecto para las ciencias de la salud, ya que se puede utilizar en Epidemiología, Salud Pública, Nutrición, Farmacología, Genómica, Biología Molecular, Citometría de Flujo, etcétera.

¿Qué puede hacer R?

Se muestra un listado general sobre algunas de los usos de R.

Análisis de datos

1. **Estadística descriptiva:** calcular medidas de tendencia central (media, mediana, moda), medidas de dispersión (varianza, desviación estándar, rangos y otros) y distribuciones de frecuencia .
2. **Pruebas de hipótesis clásicas:** realizar t-tests, análisis de varianza, pruebas hipótesis paramétricas clásicas, pruebas de hipótesis no paramétricas, entre otras.
3. **Regresión:** modelos de regresión lineal y no lineal, regresión logística, modelos de regresión múltiple.
4. **Análisis multivariante:** análisis de componentes principales (PCA), análisis de clústeres, análisis factorial.

Visualización de datos

1. **Gráficos básicos:** histogramas, gráficos de barras, gráficos de dispersión, gráficos de caja y bigotes. [Ejemplos](#).
2. **Gráficos avanzados:** mapas de calor, gráficos de densidad, gráficos de violín, visualizaciones interactivas con paquetes como [ggplot2](#) y [plotly](#).

Manipulación de datos

1. **Limpieza y transformación de datos:** filtrado, selección, agrupamiento y reordenamiento de datos.
2. **Manejo de datos faltantes:** identificación y tratamiento de datos faltantes.
3. **Fusión y unión de datos:** combinar diferentes fuentes de datos utilizando funciones de join.

Programación

1. **Scripting y automatización:** creación de *scripts* para automatizar procesos de análisis de datos.
2. **Desarrollo de paquetes:** desarrollar y compartir paquetes personalizados para funciones específicas.
3. **Integración con otras herramientas:** comunicación con bases de datos, integración con lenguajes como Python y SQL.

Modelado predictivo y machine learning

1. **Clasificación y regresión:** implementar algoritmos de machine learning como árboles de decisión, bosques aleatorios, máquinas de soporte vectorial (SVM).
2. **Clustering:** K-means, clustering jerárquico.
3. **Redes neuronales:** implementar redes neuronales y modelos de aprendizaje profundo.

Bioestadística

1. **Análisis de supervivencia:** curvas de Kaplan-Meier, modelos de riesgos proporcionales de Cox.
2. **Curvas ROC y AUC:** evaluación del rendimiento de modelos clasificatorios.
3. **Estudios de asociación genética:** GWAS, análisis de variantes genéticas.

Gestión y manipulación de bases de datos

1. **Importación y exportación de datos:** leer y escribir archivos en formatos CSV, Excel, JSON, SQL, entre otros.

Documentación y reportes

1. **Generación de reportes:** crear reportes en formato HTML, PDF y Word usando RMarkdown y Quarto.
2. **Dashboards:** crear dashboards interactivos con paquetes como Shiny.

Instalación de R

R está disponible para la mayoría de los sistemas operativos, se puede descargar para Linux, Windows y MacOS. Para descargar R por primera vez diríjase a: <https://www.r-project.org>, a continuación de clic en el apartado de descarga ([download R](#)) y seleccione cualquiera de los repositorios del CRAN que ahí se enlistan. El [CRAN](#) es el *The Comprehensive R Archive Network*, una red en la que se archivan todas las versiones de R base, así como todos los paquetes para R que han pasado por un proceso de revisión riguroso, realizado por el CRAN Team, que se encarga de asegurar su correcto funcionamiento. Aunque la instalación de R en cualquiera de los sistemas operativos es sencilla e intuitiva, se describen algunas peculiaridades a continuación.

Instalar R en Windows

Para utilizar R en Windows solo basta con descargar la versión más reciente del [CRAN](#). El archivo que se descargará será un archivo .exe. En la medida de lo posible trate de instalar la versión más reciente de R. A la fecha de escritura de este libro, 4.4.0. es la versión más reciente. Por otro lado, si se pregunta qué versión instalar de 32 o 64 bits, actualmente, la mayoría de los equipos de computo tienen un procesador de 64 bits. Si usted está utilizando Windows 10 o posterior, es casi un hecho que su procesador es de 64 bits. El siguiente enlace debería descargar la versión [más reciente de R para Windows](#). Una vez descargado, siga los pasos del asistente de instalación. De momento, instale la configuración predefinida.

Instalar R en MacOS

En el caso de MacOS el procedimiento es muy similar a Windows, sin embargo, en este caso descargamos un archivo binario. Actualmente existen dos versiones de R para MacOS: una para arquitectura x86 para computadoras con procesador de Intel y otra para arquitecturas ARM para computadoras con Chip M1 o

más actual. Si usted cuenta con una computadora con arquitectura ARM, puede ejecutar cualquiera de las versiones, pero es altamente recomendable utilizar la versión que corresponde, para correr R de forma nativa. Si su computadora utiliza procesadores con arquitectura x86, descargue R [aquí](#). En cambio, si su computadora posee un procesador con arquitectura ARM, descargue R [aquí](#). Una vez descargado, siga los pasos del asistente de instalación. De momento, instale la configuración predefinida.

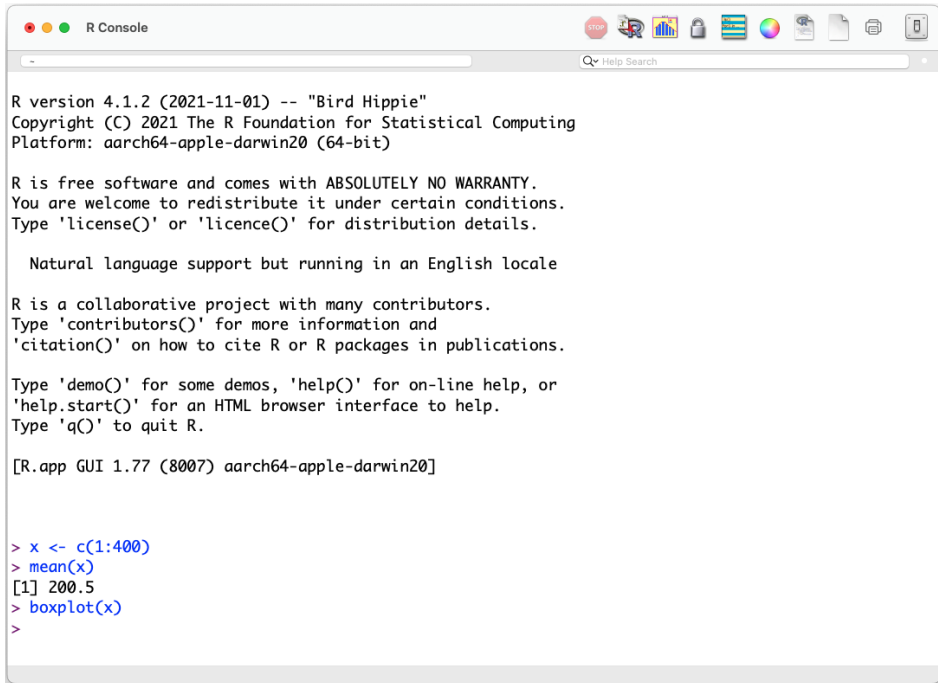
Instalar R en Linux

En Linux usted puede descargar R desde el [CRAN](#) para la mayoría de las distribuciones de Linux. Se encuentran archivos para Debian, Fedora, Redhat, Suse y Ubuntu. Sin embargo, R se puede instalar para la mayoría de las distribuciones desde consola. Si usted utiliza Linux, no deberá tener problema para instalar R.

Consola de R

Cuando usted abre R por primera vez, verá una imagen parecida a la que se muestra en la [figura 1.1](#), tome en cuenta que la visualización dependerá del sistema operativo, pero son muy similares. Esta imagen muestra la consola de R, la consola es la ventana principal donde interactúa directamente con el lenguaje. Es decir, la consola es donde usted escribe instrucciones en forma de código, donde el lenguaje las interpreta y en el mejor de los casos nos devuelve un resultado. En los primeros laboratorios de este libro, el trabajo será directamente en la consola, pero en la medida de lo posible estaremos utilizando *scripts*.

FIGURA 1.1. CONSOLA DE R



```
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.77 (8007) aarch64-apple-darwin20]

> x <- c(1:400)
> mean(x)
[1] 200.5
> boxplot(x)
>
```

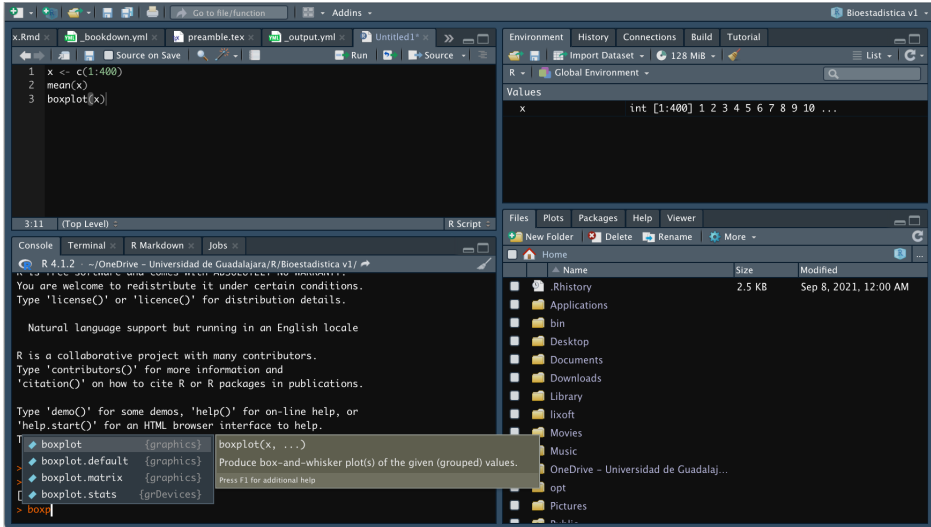
RStudio

RStudio es un entorno de desarrollo integrado (IDE) para R. Incluye una consola, un editor de resaltado de sintaxis que admite la ejecución directa de código, así como herramientas para el trazado, el historial, la depuración y la gestión del espacio de trabajo.

Utilizar un IDE puede facilitar mucho su trabajo y la creación de código. Un IDE ayuda a que la programación sea mucho más fácil y productiva. Entre otras cosas, la sintaxis de su código se resalta con distintos colores, lo que facilita su lectura y escritura. Puede previsualizar los colores de sus gráficos. Además, RStudio puede completar su código de forma mucho más intuitiva

y sin necesidad de memorizar todas las funciones que utiliza. Una comparación de la consola de R y RStudio utilizando el mismo código se muestran en [figura 1.1](#) y [figura 1.2](#).

FIGURA 1.2. CONSOLA DE R UTILIZANDO RSTUDIO

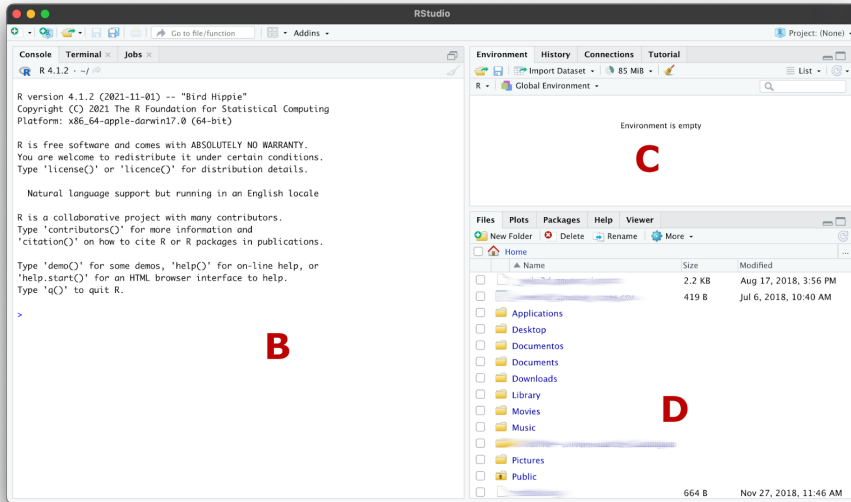


Actualmente RStudio pertenece a la compañía Posit. Aunque existen versiones de pago, si el uso que le damos a RStudio es académico podemos utilizar la versión gratuita de este IDE. Puede instalar RStudio según su sistema operativo desde [aquí](#). La instalación es sencilla, solo siga las instrucciones. El único requisito que necesita es tener R instalado.

Estructura de RStudio

En la resolución de estos ejercicios y en los restantes del libro trabajaremos directamente en RStudio y no en la consola de R. Cuando usted abre por primera vez Rstudio estará observando algo muy similar a lo mostrado en la [figura 1.3](#).

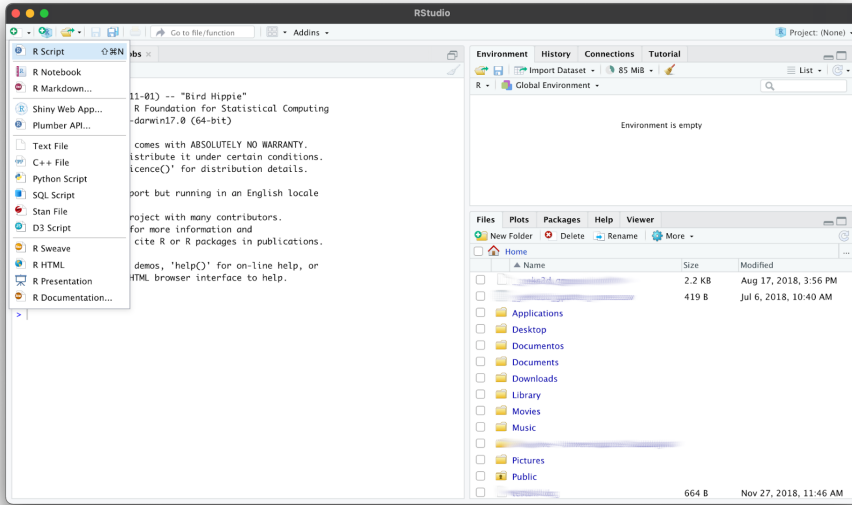
FIGURA 1.3. VISUALIZACIÓN DE RSTUDIO PRIMERA APERTURA



En el apartado B de la [figura 1.3](#) se encuentra la consola de R que corresponde al entorno computacional de este lenguaje (donde se interpretan las ordenes). En la consola podemos escribir código, se realizará la interpretación y nos dará un resultado. En el apartado C se encuentra, entre otras cosas, el ambiente de trabajo. Es aquí donde se irán mostrando los objetos, *Dataframe*, listas, etc., que vayamos creando. En este apartado, en otra de las pestañas también podrá encontrar el historial de nuestro código. Finalmente, en el apartado D, se muestran los archivos que tenemos en nuestra computadora para poder acceder a ellos. Otras pestañas muestran la previsualización de los gráficos que se crean en R (Plots), el listado de los paquetes (Packages), la ayuda de las funciones y paquetes (Help) y un visualizador html (Viewer).

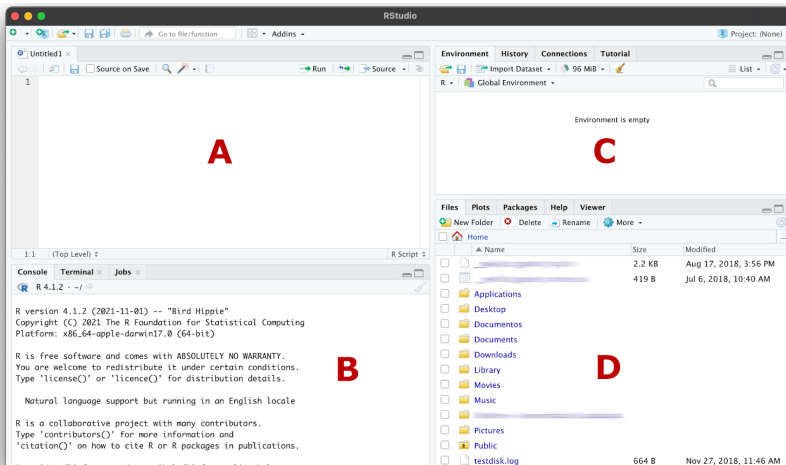
Una de las ventajas de utilizar un IDE (RStudio) es que podemos utilizar scripts para correr nuestro código. Un *script*, es un archivo de texto en el que podemos crear nuestro código para que después sea interpretado por la consola de R. Los scripts tienen la extensión .R, por ejemplo *mi_primer_script.R*. En la [figura 1.4](#) se muestra cómo crear un script.

FIGURA 1.4. CREACIÓN DE UN SCRIPT EN RSTUDIO



Ahora su RStudio lucirá como lo muestra la [figura 1.5](#). En el apartado A se incluyen los scripts. Esta es la versión con la que trabajaremos en el resto de este libro. El resto de los apartados es igual a lo descrito en la [figura 1.3](#). En el siguiente capítulo realizaremos algunos ejercicios que nos permitirán comprender algunos fundamentos básicos de R.

FIGURA 1.5. VISUALIZACIÓN DE RSTUDIO CON CUATRO APARTADOS



Referencias

- Batra, N., Neale, A., Spina, A., Blomquist, P., Campbell, F., Laurenson-Schafer, H., Florence, I., Ndaiye, A. *et al.* (2021). *The Epidemiologist R Handbook*.
- Buuren, S. van y Groothuis-Oudshoorn, K. (2011). Mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3).
- Dai, Y., Xu, A., Li, J., Wu, L., Yu, S., Chen, J., Zhao, W., Sun, X.-J. y Huang, J. (2021). CytoTree: An R/Bioconductor package for analysis and visualization of flow and mass cytometry data. *BMC Bioinformatics*, 22(1), 138. <https://doi.org/10.1186/s12859-021-04054-2>
- Everitt, B. y Hothorn, T. (2011). *An Introduction to Applied Multivariate Analysis with R*. Springer. <https://doi.org/10.1007/978-1-4419-9650-3>
- Ghatak, A. (2017). *Machine Learning with R*. Springer Singapore. <https://doi.org/10.1007/978-981-10-6808-9>
- Hui, E. G. M. (2019). Introduction. En E. G. M. Hui (Ed.), *Learn R for Applied Statistics* (pp. 1-18). Apress. https://doi.org/10.1007/978-1-4842-4200-1_1
- Jeong, H.-W., Hernández-Rodríguez, B., Kim, J., Kim, K.-P., Enriquez-Gasca, R., Yoon, J., Adams, S., Schöler, H. R., Vaquerizas, J. M. y Adams, R. H. (2017). Transcriptional regulation of endothelial cell behavior during sprouting angiogenesis. *Nature Communications*, 8(1), 726. <https://doi.org/10.1038/s41467-017-00738-7>
- Ortutay, Z. y Orturay, C. (2017). *Molecular Data Analysis Using R*. Wiley. <https://www.wiley.com/en-us/Molecular+Data+Analysis+Using+R-p-9781119165026>
- R Core Team.(2024). *R: A language and environment for statistical computing*. <https://www.R-project.org/>
- R Core Team.(2024). *What is R?* <https://www.r-project.org/about.html>
- Shakya, P. R., Melaku, Y. A., Shivappa, N., Hébert, J. R., Adams, R. J., Page, A. J. y Gill, T. K. (2021). Dietary inflammatory index (DII®) and the risk of depression symptoms in adults. *Clinical Nutrition (Edinburgh, Scotland)*, 40(5), 3631-3642.
- Shalit, A., Moon, D. y Starbuck, O. (1996). *The Dylan Reference Manual: The Definitive Guide to the New Object-Oriented Dynamic Language*. Reading, Mass: Addison-Wesley Developers Press.
- Sievert, C. (2020). *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. Hall/CRC. <https://plotly-r.com/>

- Voysey, M., Clemens, S. A. C., Madhi, S. A., Weckx, L. Y., Folegatti, P. M., Aley, P. K., Angus, B. *et al.* (2021). Safety and efficacy of the ChAdOx1 nCoV-19 vaccine (AZD1222) against SARS-CoV-2: An interim analysis of four randomised controlled trials in Brazil, South Africa, and the UK. *The Lancet*, 397(10269), 99-111.
- Wickham, H., Çetinkaya-Rundel, M. y Grolemund, G. (2018). *R for Data Science: Import, Tidy, Transform, Visualize, and Model*. <https://r4ds.hadley.nz/>
- Wickham, H., François, R., Henry, L., Müller, K. y Vaughan, D. (2023). *Dplyr: A Grammar of Data Manipulation*. <https://dplyr.tidyverse.org/>
- Yu, G., Wang, L.-G., Han, Y. y He, Q.-Y. (2012). clusterProfiler: An R package for comparing biological themes among gene clusters. *Omics: A Journal of Integrative Biology*, 16(5), 284-287. <https://doi.org/10.1089/omi.2011.0118>

Laboratorio de R 1

Edsaúl Emilio Pérez Guerrero

En este primer laboratorio de R se abordarán conceptos básicos sobre la consola de R, el uso de R como calculadora, la creación de vectores, creación y llamado de objetos, entre otros.

Los objetivos aprendizaje de este primer laboratorio de R son:

1. Uso de R como calculadora
2. Creación y llamado de objetos
3. Creación de vectores
4. Funciones (introducción básica)
5. Directorio y espacio de trabajo
6. Librerías

Descripción del Dataframe bacteria

Para resolver estos primeros ejercicios vamos a utilizar una muestra de los datos contenidos en *Dataframe bacteria* que se encuentra en la librería MASS. Este *Dataframe* fue modificado para realizar algunos ejercicios del compendio, por lo que se podrán encontrar algunas diferencias del *Dataframe* original. En el apartado “Definición de un Dataframe” profundizaremos sobre lo que es un *Dataframe*, de momento solo es conveniente saber que es una estructura ordenada de datos en el que cada una de las filas son un caso (paciente) y las columnas son una variable.

El *Dataframe* *bacteria* contiene datos en los que el doctor A. Leach probó los efectos de un fármaco en 50 niños con antecedentes de otitis media en el Territorio del Norte de Australia. Los niños fueron asignados al azar al fármaco o al placebo, y también para recibir un estímulo activo para cumplir con la toma del fármaco. Se comprobó la presencia de *H. influenzae* en las semanas 0, 2, 4, 6 y 11; faltaban 30 de las comprobaciones y no se incluyen en este marco de datos. Puede encontrar más información sobre el estudio de Leach [aquí](#). Este conjunto de datos se modificó para los fines de este capítulo. Los variables contenidas en esta muestra de datos son las siguientes:

- ♦ **ID**: es un identificador para cada uno de los pacientes.
- ♦ **y**: indica la presencia ausencia de la otitis. Es un factor con dos niveles *n* que indica la presencia de otitis e *y* que indica su ausencia.
- ♦ **ap**: es un factor con los niveles *a* (*estímulo activo*) y *p* (*placebo*) que indican si el paciente recibió estímulo activo para cumplir con la toma del fármaco.
- ♦ **hilo**: indica el cumplimiento del paciente. *hi* representa el cumplimiento alto y *lo* cumplimiento bajo.
- ♦ **week**: es una variable numérica que indica la semana en la que se realizó la prueba para la detección de *H. influenzae*.
- ♦ **trt**: un factor con niveles de *placebo*, *drug* y *drug+* (recodificación de la variable *ap* e *hilo*).
- ♦ **time**: es una variable numérica que indica el tiempo en días de la infección por otitis. Esta variable no se encuentra en el *Dataframe* original. Esta es una variable ficticia que se agregó solamente para la realización de algunos ejercicios de este compendio. Esta variable solo se encuentra llena para los pacientes que sí presentaron otitis (*y*).
- ♦ **age**: es una variable numérica que indica la de los pacientes. Esta variable no se encuentra en el *Dataframe* original. Esta es una variable ficticia que se agregó solamente para la realización de algunos ejercicios de este compendio.

La [tabla 2.1](#) muestra los datos que necesitaremos para nuestros ejercicios.

TABLA 2.1. MUESTRA DE 12 DATOS DEL DATAFRAME BACTERIA

ID	Y	AP	HILO	WEEK	TRT	TIME	AGE
Patient A	n	a	hi	6	drug+	24	5
Patient B	n	a	hi	2	drug+	18	3
Patient C	n	a	hi	4	drug+	7	3
Patient D	n	a	hi	6	drug+	26	4
Patient E	n	p	hi	0	placebo	26	4
Patient F	y	p	hi	6	placebo	NA	3
Patient G	y	p	hi	11	placebo	NA	4
Patient H	y	a	lo	0	drug	NA	7
Patient I	y	a	lo	2	drug	NA	6
Patient J	y	a	lo	4	drug	NA	3
Patient K	y	a	lo	6	drug	NA	3
Patient L	n	a	lo	2	drug	20	4

En esta sección no utilizaremos el *Dataframe bacteria* para importarlo en R, pero sí para realizar algunos ejercicios que nos permitirán familiarizarnos con R.

R como calculadora

Supongamos que queremos conocer el tiempo en días que los tres primeros pacientes tuvieron de duración de la infección. Para ellos nos dirigimos a la consola de R (el apartado B de la [figura 1.5](#)). Ahí podemos sumar los días de duración de la infección y la consola arrojará el resultado. Para ello solo basta con poner el siguiente código:

```
24+18+7
```

Y presionar *Enter* para obtener el resultado

```
[1] 49
```


Si quisiéramos saber cuál es la diferencia del tiempo de infección entre los pacientes 1 y 2, tecleamos en la consola lo siguiente:

```
24-18
```

Y obtenemos lo siguiente:

```
[1] 6
```

Directamente en la consola podemos hacer múltiples operaciones aritméticas, desde sumas, restas, divisiones, potencias, raíz cuadrada, entre otras. Solo debemos recordar que R sigue la jerarquía de las operaciones y que podemos apoyarnos de paréntesis para separar operaciones. La [tabla 2.2](#) nos muestra los principales operadores numéricos que se utilizan en R.

TABLA 2.2. OPERADORES NÚMERICOS/ARITMETICOS EN R

SÍMBOLO (OPERADOR)	OPERACIÓN	EJEMPLO	RESULTADO
+	Suma	4+3	7
-	Resta	4-3	1
*	Multiplicación	4*3	12
/	División	4/3	1
^	Potencia	4^3	64

Al final del capítulo podrá encontrar otros ejercicios que le permitirán practicar más las funciones de R como calculadora, pero desde ahora, le recomendamos utilizar R como su calculadora de confianza; esto, además de práctico, le permitirá familiarizarse cada vez más con R.

Objetos en R

R es un lenguaje de programación orientado a objetos; esto significa que todo en R, desde los datos hasta las funciones, son objetos y podemos hacer múltiples operaciones con ellos. Es decir, podemos crear objetos, manipularlos, sumarlos,

estimar el promedio, crear gráficas, entre otros. El *Dataframe* *bacteria* es un objeto. Cuando hablamos de un objeto, en R hablamos de cualquier cosa que podemos encapsular y luego hacer operaciones.

Continuando con las operaciones del *Dataframe* *bacteria*, ahora supongamos que queremos conocer el tiempo promedio de duración de la infección (otitis) en los pacientes de la [tabla 2.1](#). Recuerde que el tiempo de duración de la infección solo aplica para los individuos identificados con una *n* en la variable *y*. El primer paso sería sumar el tiempo de todos los niños y luego dividirlo entre el total de niños seleccionados como se muestra a continuación.

```
(24+18+17+26+26+20)/6
```

Nuestro resultado es:

```
[1] 21.83333
```

Sin embargo, podemos también crear un objeto que contenga la suma del tiempo de la infección para después dividirlo entre 6. En este caso nuestro objeto se llamará *suma*:

```
# Creación de nuestro objeto llamado suma
suma ← 24+18+17+26+26+20
# Nuestro objeto tendrá el resultado de sumar 24+18+17+26+26+20
# El texto a la izquierda del símbolo
# es un comentario
```

En el código anterior podemos resaltar varias cuestiones. La primera es que lo que se encuentre a la derecha del símbolo *#* es un comentario y por lo tanto no será leído como código. En el ejemplo anterior podemos encontrar dos comentarios. El primero es *# Creación de nuestro objeto llamado suma*, el segundo sería *# Nuestro objeto tendrá el resultado de sumar 24+18+17+26+26+20*.

Otro punto importante a destacar es el uso del símbolo `<-` el cual en R se utiliza para hacer asignaciones. Es decir, en nuestro ejemplo le asignamos a `suma` el resultado de sumar $24+18+17+26+26+20$. Vea al símbolo `<-` como una flecha que indica que los datos de la izquierda son asignados al nombre de la derecha, es decir, en dirección de la flecha.

Podemos llamar a nuestro objeto `suma` y hacer operaciones con él. Solo basta con teclear en nuestra consola `suma` (cuide el uso de mayúsculas y minúsculas) para obtener el resultado de $24+18+17+26+26+20$ como se muestra a continuación:

```
suma # Teclear en la consola
```

Nuestro resultado es:

```
[1] 131
```

Ahora, una vez creado nuestro objeto podemos hacer operaciones con él. Por ejemplo, calcular el tiempo promedio de duración de la infección:

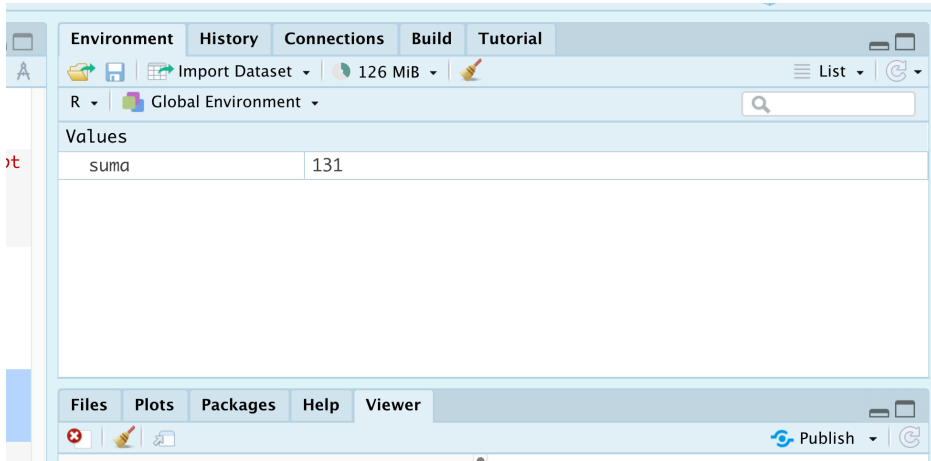
```
suma/6 #Para obtener el promedio de la duración de la infección
```

Nuestro resultado es:

```
[1] 21.83333
```

Cuando creamos objetos en RStudio, en el apartado de ambiente (apartado C de la [figura 1.5](#)) de trabajo podemos ver el objeto en sí y sus características tal cual se muestra en la [figura 2.1](#).

FIGURA 2.1. OBJETOS GUARDADOS EN EL AMBIENTE DE TRABAJO



También podemos crear un objeto llamado resultado que tendrá el promedio de los días de duración de la infección. Podemos utilizar nuestro objeto suma que creamos anteriormente.

```
# Este objeto tendrá el promedio de los días de infección
resultado <- suma/6 # Utilizamos el objeto "suma"
```

Si queremos acceder a este nuevo objeto, solo es necesario teclear resultado en nuestra consola.

```
resultado
```

Nuestro resultado es:

```
[1] 21.83333
```

Hasta ahora hemos revisado cómo crear un objeto que asignábamos al promedio y a la suma del tiempo de duración de la otitis en un grupo de pacientes. Pero ¿qué pasa si queremos asignar a un objeto el identificador de los primeros 5 pacientes? ¿Si queremos guardar un conjunto de números y después hacer operaciones con él?

Texto en los objetos

Para crear un objeto que contenga los identificadores de los primeros 5 pacientes del *Dataframe* *bacteria* utilizamos el siguiente código:

```
Identificador ← c("Patient A", "Patient B", "Patient C",
                  "Patient D", "Patient E")
```

Antes de llamar al objeto, es necesario profundizar en algunas particularidades del código.

Se creó un objeto llamado *Identificador* que contiene los identificadores de los primeros 5 pacientes de la tabla 2.1).

Se utilizó el símbolo `<-` que asigna los identificadores al objeto.

Se utilizaron las comillas para cada uno de los identificadores. Todo el texto o números entrecomillados le indican a R que se trata de caracteres. Si, por ejemplo, en lugar de poner "Patient A" hubiéramos escrito simplemente Patient A, R buscaría un objeto llamado Patient A.

Utilizamos la letra *c*. La letra es una función que se utiliza para combinar diversos componentes en un objeto que se encuentran dentro de paréntesis. [En la sección Introducción a las funciones](#) se hace una introducción a las funciones de R.

Se utiliza la coma para separar cada uno de los componentes de nuestro objeto.

Si llamamos a nuestro objeto *Identificador* obtenemos lo siguiente. Note cómo se conservan las comillas.

```
[1] "Patient A" "Patient B" "Patient C" "Patient D" "Patient E"
```

Vectores numéricos en los objetos

Ahora bien, si es de nuestro interés tener un listado o vector de número con las edades de los 12 pacientes, podemos utilizar el siguiente código en el que creamos un objeto llamado *edades*.

```
edades ← c(5,3,3,4,4,3,4,7,6,3,3,4)
```

Note cómo el código tiene las mismas características explicadas anteriormente, pero no utiliza las comillas. Las comillas no son necesarias para los números, siempre y cuando no quiera que los números se comporten como texto o como factor.

Si llamamos al objeto `edades` obtenemos lo siguiente:

```
[1] 5 3 3 4 4 3 4 7 6 3 3 4
```

Una de las ventajas del lenguaje de programación orientado a objetos es que podemos hacer operaciones con ellos. Por ejemplo, puedo restarle 2 a cada uno de los componentes del objeto `edades` con el siguiente código:

```
edades-2
```

Cuyo resultado sería:

```
[1] 3 1 1 2 2 1 2 5 4 1 1 2
```

Note como R le restó 2 a cada número del objeto `edades`.

Incluso podemos unir dos objetos. En el siguiente código se sumarán dos objetos: el que contiene las edades y otro objeto que tiene las mismas edades, pero al cuadrado. Estos objetos colocarán un objeto llamado `edades_final`.

```
edades2 <- edades^2
# Creación del objeto con las edades al cuadrado
edades_final <- c(edades,edades2)
# Objeto con todas las edades
```

Si llamamos al objeto obtenemos lo siguiente:

```
[1] 5 3 3 4 4 3 4 7 6 3 3 4 25 9 9 16 16 9 16 49 36 9 9 16
```

Puede crear secuencias de números utilizando el símbolo `:`. Por ejemplo, el siguiente código crea una secuencia del 1 al 25.

```
secuencia ← 1:25
```

Si llamamos a nuestro objeto obtendremos:

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

Note cómo no es necesario utilizar la función `c`. Sin embargo, si su objeto tiene más de dos secuencias sí es necesario, por ejemplo:

```
secuencia2 ← c(1:25, 27:50)
```

Si llamamos al objeto anterior obtenemos:

```
secuencia2
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50
```

Nombres de objetos en R

Los nombres de las variables y objetos deben comenzar con una letra, aunque pueden incluir letras, números, puntos y guiones bajos. Si por alguna razón los nombres de los objetos comienzan con un punto, este puede ser utilizado siempre deber ir seguido de una letra.

Una característica importante de los objetos es que deben de tener un nombre para identificarlos y que no podemos asignar el mismo nombre a dos objetos, ya que sobrescribiremos el primer objeto. Algunas recomendaciones para asignarles nombres a los objetos son:

- ♦ No comenzar con número.
- ♦ Es importante que los nombres de sus objetos reflejen claramente su propósito y contenido. Use nombres descriptivos, pero no largos.
- ♦ Siga un sistema. Siempre nombre los objetos de la misma manera. Recomendamos altamente utilizar nombres de objeto que comiencen con

mayúscula cada que inicia una palabra y que los espacios sean un guion bajo. Por ejemplo: “Mi_Primer_Objeto”, “Suma_Edad”, etc.

- ♦ Evite usar palabras reservadas de R como nombres de objetos, ya que pueden causar conflictos y confusiones en tu código.
- ♦ Evite caracteres especiales.
- ♦ Evite nombres largos. Si bien es importante que los nombres sean descriptivos, trate de que los nombres de los objetos no sean largos. Imagine teclear 80 veces el nombre: “Glucosa_En_Ayuno_De_Los_Pacientes”.

En resumen, se recomienda usar nombres cortos, claros, no ambiguos y descriptivos. Esto nos ahorrará muchos dolores de cabeza.

Introducción a las funciones

En R existen funciones que son básicamente un conjunto de operaciones que nos darán uno o más resultados (incluidos resultados numéricos, gráficas y otros). Estos resultados pueden ser de características muy distintas, por ejemplo, pueden ser números, números y texto, gráficas, etc. Las funciones trabajan con argumentos, es decir, aquello que nos permite modificar la forma en la que la función realiza sus operaciones.

La estructura básica de la función es la siguiente: nombre de la función(). Dentro de los paréntesis se incluyen los argumentos y los datos con los que será *alimentada* la función.

Algunos ejemplos de funciones son:

- ♦ `mean()`. Función para obtener el promedio de un conjunto de datos
- ♦ `summary()`. Función para obtener un resumen
- ♦ `c()`. Función que nos permite combinar
- ♦ `quantile()`. Función que nos permite obtener los percentiles de un conjunto de datos
- ♦ `help()`. Función para obtener ayuda de otras funciones

Por ejemplo, si queremos obtener el promedio de duración de la infección del grupo de niños de la [tabla 2.1](#), podemos utilizar el siguiente código:

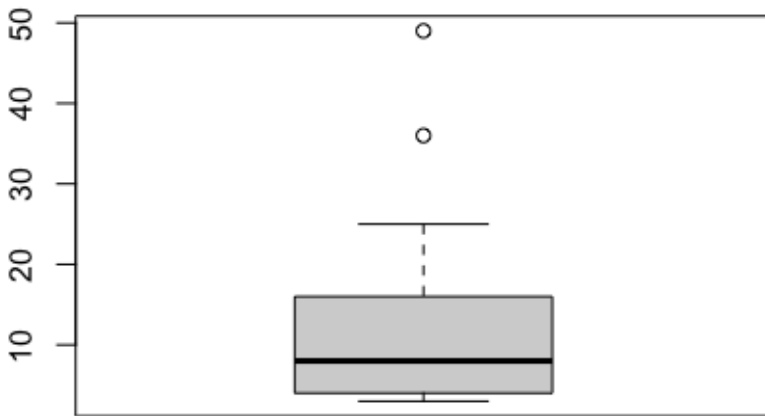

```
mean(24,18,17,26,26,20)
```

Que nos dará como resultado:

```
[1] 24
```

También podemos crear una gráfica:

```
boxplot(edades_final)
```



```
# Se utilizó el objeto creado anteriormente
# Con este código se crea un gráfico de caja y bigotes
```

Función de ayuda

La función mas importante en R es la ayuda `help()`. Esta función nos permitirá encontrar documentación para conocer la estructura de paquetes, funciones y datos. Por ejemplo, puede solicitar ayuda de la función `mean()` con el siguiente código:

```
help("mean")# Note el entrecomillado. Si no lo utiliza R buscará un objeto
```

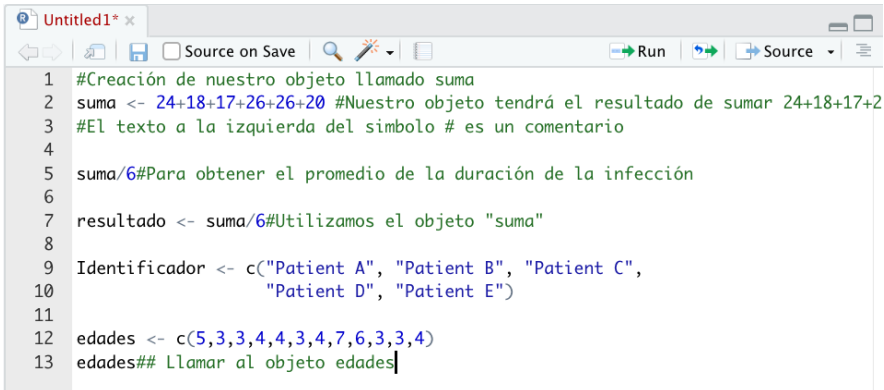
El resultado de la ayuda se mostrará en el apartado inferior derecho de RStudio (ver [figura 1.5](#), apartado D). En la ayuda mostrada, podrá encontrar la estructura de la función (como alimentar la función), los argumentos y su uso, y al final algunos ejemplos. Recomiendo siempre tomar los ejemplos mostrados en ayudas y adaptarlos a nuestras necesidades. Esto facilitará mucho su proceso de aprendizaje de R.

Uso de scripts

Es momento de dejar de utilizar la consola para utilizar los *scripts*. La descripción de un *script* y de cómo abrir uno nuevo se encuentra en el apartado [Estructura de Rstudio](#).

Una de las ventajas de los *scripts* es que podemos organizar nuestro código y guardarlo. Además, en los *scripts* podemos poner comentarios y ver exactamente qué línea de código es en la que estamos trabajando. La [figura 2.2](#) muestra el *script* que contiene código utilizado en este capítulo.

FIGURA 2.2. EJEMPLO DE UN SCRIPT



```

1 #Creación de nuestro objeto llamado suma
2 suma <- 24+18+17+26+26+20 #Nuestro objeto tendrá el resultado de sumar 24+18+17+2
3 #El texto a la izquierda del simbolo # es un comentario
4
5 suma/6#Para obtener el promedio de la duración de la infección
6
7 resultado <- suma/6#Utilizamos el objeto "suma"
8
9 Identificador <- c("Patient A", "Patient B", "Patient C",
10                  "Patient D", "Patient E")
11
12 edades <- c(5,3,3,4,4,3,4,7,6,3,3,4)
13 edades## Llamar al objeto edades
  
```

Para correr alguna parte o todo el código del *script*, puede seleccionar el código que desee evaluar y presionar el botón de *Run* (ver [figura 2.3](#)). Otra opción es colocarse al inicio o al final de una instrucción y presionar `ctrl + enter` en Windows o `cmd + enter` en Mac. Es importante identificar dónde comienza y

dónde termina la instrucción. Hay una sola instrucción en las líneas 9 y 10 de la [figura 2.3](#), pero dos instrucciones en las líneas 12 y 13. Un consejo para identificar las instrucciones tiene que ver con el cierre de los paréntesis y de las sangrías. Note cómo la línea 10 no comienza con la misma sangría que la línea 9, esto es un indicativo de que la instrucción continua.

FIGURA 2.3. BOTÓN DE RUN



Los autores de este libro recomendamos documentar siempre los *scripts*, es decir, añadir en la medida de lo posible comentarios que ayuden a explicar su código. Los comentarios hacen que su código sea más comprensible para usted y para otros (a veces hacemos código para otros). Los *scripts* son su bitácora de trabajo. Por último, no olvide que el código de la línea 9 afectará al código de la línea 10 y así sucesivamente.

Directorio de trabajo

Una de las principales características de R es que podemos trabajar en directorios de trabajos (carpetas). El directorio de trabajo es el lugar de nuestra computadora en el que se encuentran los archivos que estamos trabajando con R. Además, es aquí donde se guardarán todos nuestros archivos como gráficos, informes, etcétera.

Por ejemplo, el directorio de trabajo de este compendio es: Users/carpetas1/carpetas2/BioestadísticaV2

Si le interesa conocer cuál es el directorio de trabajo en el que se encuentra trabajando, utilice el siguiente código:

```
getwd()
```

Puede cambiar el directorio de trabajo usando la función `setwd()`. Esta función requiere alimentarse con el directorio de trabajo que queremos usar. Por ejemplo:

```
setwd("\\otro_directorio")
# No olvide el uso de las comillas
```

Note cómo todas las rutas de archivos dentro de nuestro equipo van entre comillas.

Los pasos anteriormente descritos no son muy útiles si creamos un proyecto directamente en RStudio. Para ello diríjase al cuadrante superior izquierdo y busque el símbolo mostrado en [figura 2.4](#) y [figura 2.5](#).

FIGURA 2.4. CREACIÓN DE UN NUEVO PROYECTO EN RSTUDIO

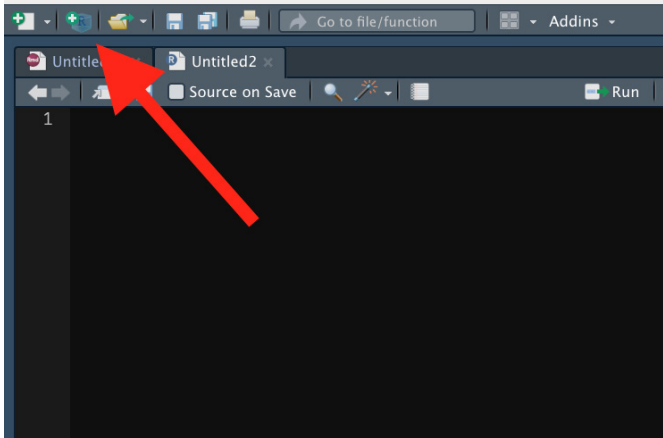
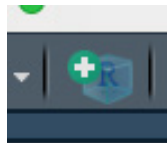


FIGURA 2.5. CREACIÓN DE UN NUEVO PROYECTO EN RSTUDIO



Una vez en el menú anterior podemos crear un proyecto en una carpeta ya existente ([figura 2.6](#)) o crearlo en una carpeta ya creada ([figura 2.7](#)). Para fines prácticos no hay diferencia entre cualquiera de las dos vías.

FIGURA 2.6. CREACIÓN DE UN NUEVO PROYECTO DESDE UN DIRECTORIO NUEVO

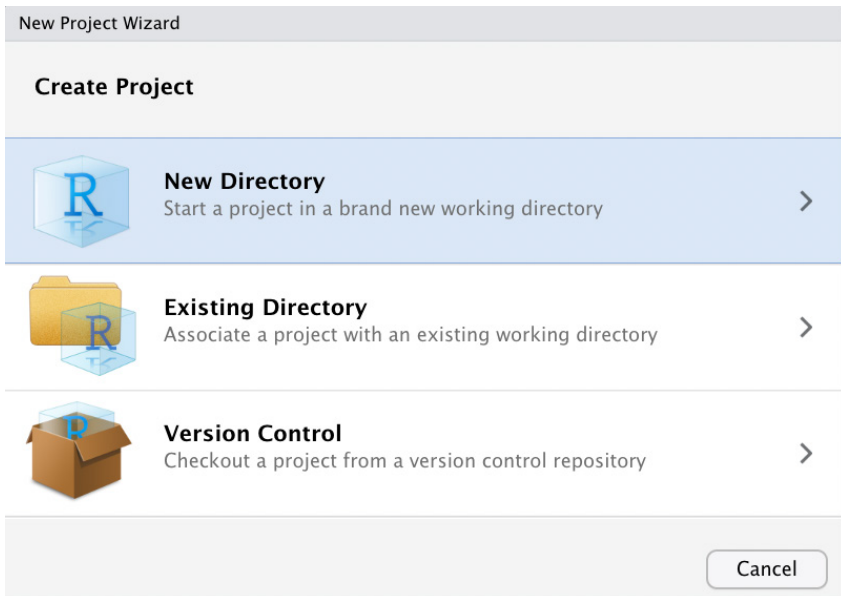
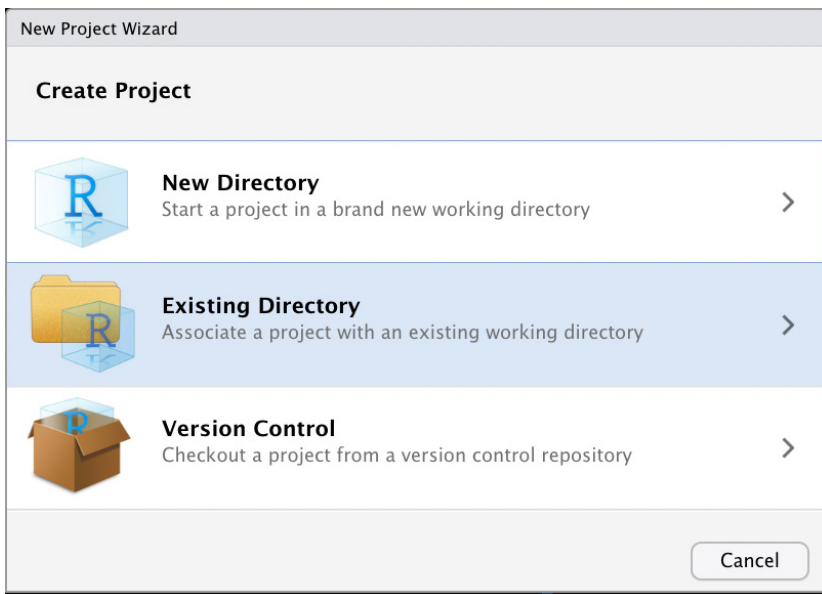


FIGURA 2.7. CREACIÓN DE UN NUEVO PROYECTO EN UNA CARPETA EXISTENTE



Las ventajas de crear un proyecto es que automáticamente nuestro directorio de trabajo cambiará al del proyecto, y desde aquí podemos manipular todos los archivos con los que trabaje. Se recomienda crear un proyecto para seguir los ejemplos de este libro, en el que incluya una carpeta de bases de datos, otra de gráficos y otra de *scripts*.

Imagine que es una persona que viaja constantemente a lugares con diferentes características climáticas. Le convendría tener una maleta con lo necesario para cuando viaja a la nieve y otra para cuando viaja a la playa. En esas maletas tendrá todo lo necesario para su viaje y en esas maletas guardará todo lo que compre en su viaje. Además, ayudaría que, en esas maletas, siempre en el mismo lugar tenga sus productos de aseo, de tal manera que siempre sepa dónde ir a buscar. Un proyecto es como a una de esas maletas. Contiene todo lo necesario para realizar nuestros análisis, es la carpeta donde se guarda nuestro trabajo y a dónde R irá a buscar los archivos necesarios; por ejemplo, un *Dataframe*. Es por ello que creamos una carpeta llamada Bases, ahí es donde iremos a buscar los *Dataframe* necesarios.

Para resolver todos los ejercicios de este libro con mayor facilidad, le recomendamos crear dentro de su proyecto la carpeta llamada Bases, con ello fácilmente podrá utilizar los códigos de ejemplo. Para crear esta carpeta siga los siguientes pasos: en el cuadrante inferior derecho que se ejemplifica en la [figura 1.5](#) apartado D, vaya a la pestaña *Files* y ahí solo presione el botón de *New Folder* para crear una nueva carpeta. Nombre a esta nueva carpeta como Bases.

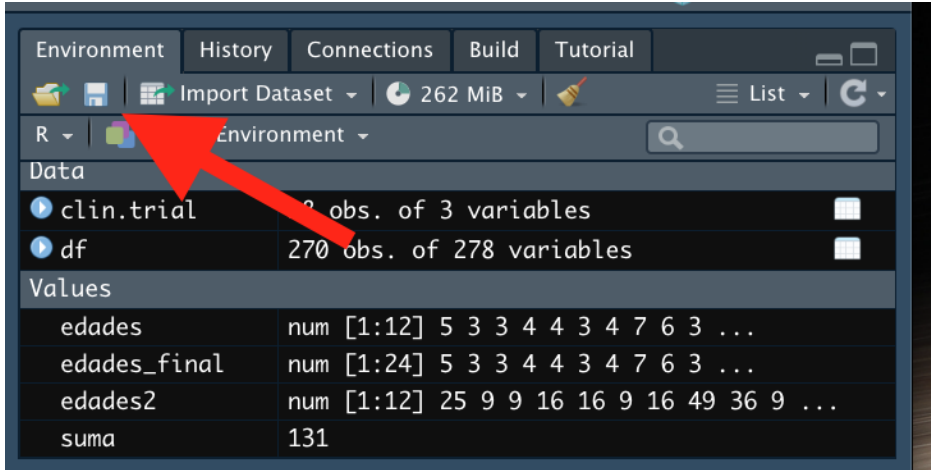
Sesión

¿Que pasaría si quiero guardar mis objetos? Recuerde que los objetos pueden ser todo R. En este caso la opción que tenemos es guardar nuestra sesión.

Los objetos y funciones de R son almacenados en la memoria RAM de nuestra computadora y no son intercambiables entre distintas sesiones. Por lo tanto, al guardar nuestra sesión podemos acceder una y otra vez a los objetos creados en determinado día de trabajo. Cuando cerramos R, cerramos nuestra sesión, aunque es posible tener más de una sesión de R activa en la misma computadora.

La forma más fácil de guardar nuestra sesión es al cerrar R. En este punto se nos preguntará si deseamos guardar el contenido de nuestra sesión. Si decidimos guárdala, crearemos un archivo con extensión *.Rdata* en tu directorio de trabajo. La otra opción es dar clic en el símbolo de guardado en nuestro ambiente de trabajo (cuadrante superior derecho). Ver [figura 2.8](#).

FIGURA 2.8. GUARDAR SESIÓN



Librerías

Para poder leer nuestro código e interpretarlo, R necesita funciones que le permitan hacer operaciones para generar resultados. Estas funciones están cargadas en librerías. En R, una librería (también llamada paquete) es una colección de funciones, datos y código básicas de R. Aunque el paquete base de R (librería que se carga cada vez que abrimos R) cuenta con las funciones básicas para poder trabajar instalando otras librerías proporcionamos a R la capacidad de realizar y utilizar funciones y operaciones más complejas, además, facilitamos mucho nuestro flujo de trabajo. Las librerías, entre otras cosas, nos sirven para:

- ♦ **Ampliar funcionalidades:** R cuenta con un conjunto básico de funciones, pero las librerías permiten añadir más, de tal manera que ahora podemos realizar análisis estadísticos avanzados, crear visualizaciones de datos, trabajar con datos complejos, manipular texto, desarrollar modelos de aprendizaje automático, crear presentaciones, reportes, libros, blogs, páginas web, entre otros.
- ♦ **Facilitar el trabajo:** las librerías contienen funciones ya escritas y probadas que le pueden ahorrar tiempo y esfuerzo. En lugar de escribir código desde cero o de escribir 500 líneas de código, podemos utilizar las funciones de las librerías para realizar tareas complejas de manera más eficiente y en una o dos líneas de código.
- ♦ **Acceder a datos:** algunas librerías incluyen conjuntos de datos de ejemplo o funciones para importar datos desde diferentes fuentes. En este libro utilizaremos muchas bases de datos (*Dataframes*) que se encuentran accesibles en librerías.

Las librerías o paquetes se encuentran alojados en el CRAN. Es importante mencionar que todos los paquetes aquí publicados se someten a un control de calidad estricto y a una validación extenuante. Aunque cualquiera puede compilar una librería en R, no todos podemos publicarla en el CRAN. Tenga la certeza que los paquetes alojados en el CRAN están perfectamente validados y no arrojarán datos erróneos.

La función que permite la instalación de librerías directamente desde el CRAN es `install.packages()`. Esta función la alimentamos con el nombre del paquete que deseamos instalar, entre comillas.

Por ejemplo:

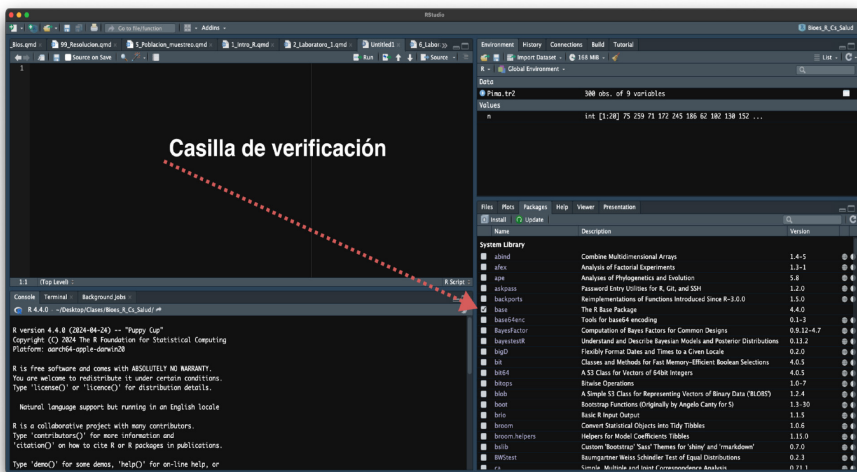
```
install.packages("Nombre-Del-Paquete")
```

Cuando el paquete se comienza a instalar aparecerán algunos mensajes en la consola mostrando el avance de la instalación. Cada vez que iniciemos una nueva sesión de R deberemos de cargar el paquete que estemos utilizando. Esto lo hacemos con la función `library()`.


```
library(Nombre-Del-Paquete) # Sin comillas
```

También podemos marcar la casilla de verificación del listado de paquetes mostrados en el cuadrante inferior derecho de Rstudio, como se muestra en la [figura 2.9](#). Cuando abre RStudio (ver [figura 1.3](#), apartado D) se muestra automáticamente la pestaña de *Files*, si quiere acceder a la pestaña de paquetes, solo debe dar clic en la pestaña de *Packages*.

FIGURA 2.9. CARGAR PAQUETES UTILIZANDO LA CASILLA DE VERIFICACIÓN



Información complementaria

- ♦ Creación de proyectos en RStudio
- ♦ Conceptos básicos de R
- ♦ ¿Cómo crear un nuevo proyecto y organizar la carpeta de trabajo?
- ♦ Instalación de paquetes en R
- ♦ Paquetes útiles en R
- ♦ R Basics

Ejercicios Laboratorio de R 1

El objetivo de estos ejercicios es poner en práctica conceptos básicos en el manejo de R como:

1. R como calculadora
2. Crear objetos
3. Realizar operaciones con objetos
4. Repasar los conceptos básicos sobre bioestadística

- ♦ **Ejercicio 2.1:** ¿qué significan los siguientes símbolos: =, -> y <- ?

Resolución ejercicio 2.1

Son símbolos de asignación. El símbolo <- se utiliza para asignar un valor a un objeto a la izquierda. Es decir, asigna las operaciones de la derecha al nombre del objeto de la izquierda, puede verlo como una especie de “flecha”. El símbolo = se utiliza para asignar un valor a un objeto. El símbolo -> es igual que el símbolo <- pero en sentido contrario.

- ♦ **Ejercicio 2.2:** ¿cuál es la diferencia entre -> y <-?

Resolución ejercicio 2.2

Ver respuesta del ejercicio anterior.

- ♦ **Ejercicio 2.3:** ¿cuál sería el código para calcular el resultado de la siguiente operación? Eleve 1.75 al cuadrado, después súmele dos. Este resultado multiplíquelo por 4 y luego divídalo en 2. Finalmente reste 4 y luego sume 8.

Resolución ejercicio 2.3

```
(((1.75^2+2)*4)/2)-4)+8
[1] 14.125
```

- ♦ **Ejercicio 2.4:** cree un objeto que contenga una secuencia de números del 1 al 256. Nombre este objeto como x.

Resolución ejercicio 2.4

```
x ← 1:256
```

- ♦ **Ejercicio 2.5:** utilizando la función `mean()` estime el promedio de este objeto.

Resolución [ejercicio 2.5](#)

```
mean(x)
[1] 128.5
```

- ♦ **Ejercicio 2.6:** a su objeto `x`, agregue otras secuencias de número del 5234 al 5240.

Resolución [ejercicio 2.6](#)

```
x ← c(x, 5234:5240)
x ← c(1:256, 5234:5240)
```

- ♦ **Ejercicio 2.7:** ¿es necesario crear otro objeto?

Resolución [ejercicio 2.7](#)

No

- ♦ **Ejercicio 2.8:** estime el promedio de su nuevo objeto `x`.

Resolución [ejercicio 2.8](#)

```
mean(x)
```

Ejercicio 2.9: a cada uno de los datos de `x` súmele 3.6 ¿Cuál es el promedio de su nuevo objeto?

Resolución [ejercicio 2.9](#)

```
x+3.6
```

- ♦ **Ejercicio 2.10:** cree un objeto que contenga al menos 6 nombres de personas y llámelo `y`.

Resolución [ejercicio 2.10](#)

```
y ← c("Nombre 1", "Nombre 2", "Nombre 3", "Nombre 4", "Nombre 5",
"Nombre 6")
```

- ♦ **Ejercicio 2.11:** consulte la ayuda para la función `rep`. ¿Cuál sería el código para realizar un objeto que contenga las palabras: “doctorado”, “biología” y “molecular”? Estas palabras deberán repetirse 6 veces antes de que aparezca la siguiente. Utilice la función: `rep`. El objeto que se genere deberá tener exactamente 90 datos. Nota: la instrucción `rep(x, r)` genera una lista de `r` valores todos iguales a `x`.

Resolución ejercicio 2.11

Hay dos formas de hacerlo:

```
help("rep")
?help
x <- rep(c("doctorado", "biología", "molecular"), each=6, times=5)
x
length(x)
```

Referencias

- Analytics Vidhya (2022). *Create packages in R and GitHub*. https://www.analyticsvidhya.com/blog/2017/03/create-packages-r-cran-github/#h2_8
- Chang, W. (2020). *R Graphics Cookbook: Practical Recipes for Visualizing Data*. O'Reilly.
- R Core Team (2024). *R: A language and environment for statistical computing*. <https://www.R-project.org/>
- Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A. y Firth, D. (2024). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. <https://cran.r-project.org/web/packages/MASS/index.html>

Introducción a la Bioestadística

Miryam Rosario Guillén Medina
Edsaúl Emilio Pérez Guerrero

En este capítulo se abordarán los conceptos básicos utilizados en bioestadística, entre otras cosas se definirán: estadística, bioestadística, estadística descriptiva, inferencial, población, muestra, universo, parámetros y estadísticos. Además, la segunda parte del capítulo se centrará en que el lector comprenda las diferencias entre los distintos tipos de variables. Finalmente, describiremos de forma muy puntual las etapas del análisis estadístico.

Conceptos básicos,

Imagine que pertenece a un grupo de investigación que tiene la tarea de probar y concluir si un nuevo tratamiento para la diabetes mellitus tipo 2 es eficaz o no. Para ello, usted deberá de diseñar el estudio, recolectar datos, analizarlos y tomar decisiones.

Para el diseño del estudio usted emplea sus conocimientos de epidemiología, metodología de la investigación, clínica y medicina basada en evidencia para proponer la realización de un ensayo clínico doble ciego con asignación al azar. Sin embargo, la recolección de los datos, su análisis y las decisiones emanadas de ellos dependerán de otras disciplinas que guardan estrecha relación con la Epidemiología y con la Metodología de la investigación: la Estadística y la Bioestadística.

La **Estadística** consiste en la recolección, cuantificación, síntesis, análisis e interpretación de la información contenida en datos para la toma de decisiones. Existen diversas ramas dentro de ella, como lo es la Estadística matemática, Estadística aplicada, Estadística descriptiva y Estadística inferencial. Para fines de su tarea, la Estadística le daría las herramientas para poder recolectar los datos de manera eficiente, almacenar esos datos, tratarlos y finalmente tomar decisiones. Sin embargo, al tratarse de datos que provienen de pacientes deben de ser entendidos utilizando otro enfoque mucho más biológico y que guarde una relación directa con la Epidemiología, Metodología de la investigación y el diseño del estudio. Por lo tanto, sería más adecuado, el empleo de la **Bioestadística**. Esta es una subdisciplina que se ocupa de los mismos aspectos descritos para estadística, pero que se encuentran relacionados a las ciencias biológicas y ciencias de la salud. Es decir, tiene las mismas funciones y aplicaciones que la Estadística, pero enfocada a disciplinas como la Biología y Medicina. Hasta este punto quizá no suene importante hacer diferencias entre Estadística y Bioestadística, sin embargo, está última toma en cuenta otros factores que de entrada no son contempladas por la Estadística, como lo pueden ser el diseño del estudio, el tipo de variables, la plausibilidad biológica, etc. La Bioestadística se enfoca en métodos estadísticos que son especialmente adecuados para abordar las características particulares de los datos biomédicos y de salud. Un ejemplo clásico de ello es el manejo de datos censurados en estudios de supervivencia. Además, la Bioestadística busca mejorar la precisión y la eficacia de las técnicas utilizadas para investigar y tratar enfermedades, evaluando la efectividad de los tratamientos y las intervenciones médicas. No es lo mismo trabajar con datos obtenidos de un censo económico a trabajar con datos obtenidos del seguimiento clínico de un paciente.

El siguiente paso para el desarrollo de su estudio sería definir su población de estudio y obtener su muestra. En Estadística, una **población** se define como el conjunto de valores por los cuales existe algún interés y que comparten características bien establecidas. Por ejemplo, nuestra población podría ser todos los pacientes que cumplan con el diagnóstico de diabetes mellitus tipo 2, con una edad de 40 a 60 años, que acudan a un hospital de segundo nivel y que hayan sido previamente tratados con insulina glargina. Estos pacientes tienen características bien delimitadas, por lo tanto, podría hablar de una población finita y podría incluir la totalidad de esa población en su estudio. En cambio, si nuestra población fuera de pacientes con el diagnóstico de pacientes con diabetes mellitus

tipo 2 sin otra característica en particular, hablaríamos de una población infinita, ya que es casi imposible saber cuántos pacientes con este padecimiento existen. Las características de nuestra población pueden ser diversas y muy variadas, por ejemplo, límites geográficos, grupos de edad, límite de tiempo, sexo etc. Sin embargo, algunos autores sostienen que la población debería de incluir tanto a los individuos pasados, como los actuales y futuros. En este escenario sería imposible acceder a una población. En esta obra consideramos que, en efecto, una población en su totalidad es difícilmente accesible, pero sí se puede delimitar.

Ahora bien, es complicado que a la hora de realizar su estudio pueda incluir a todos los pacientes con diabetes con las características de edad, procedencia y tratamientos antes descritas. En realidad, lo que usted necesita para poder realizar su estudio es una muestra. Algo que es mucho más factible económicamente y logísticamente hablando. Una **muestra** debe cumplir con ciertas características, como ser representativa, homogénea, independiente, de suficiente tamaño y aleatoria. En el [capítulo 5](#) abordaremos con más detalle las características que debe tener una muestra y cómo obtenerla.

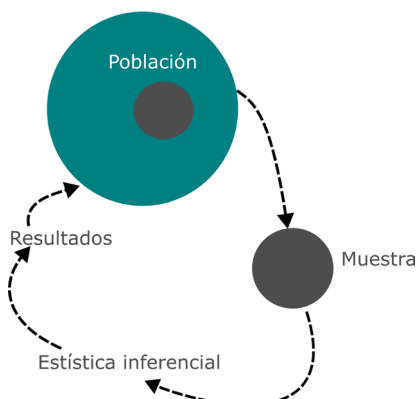
Los datos que podemos obtener de una muestra y una población son distintos. Los datos de nuestro estudio provienen de una muestra, por ejemplo, si calculáramos la media de la concentración de glucosa en ayuno, estaríamos calculando un **estadístico**. En cambio, si esa misma media es estimada para toda una población (en el supuesto de que si pudiera estimar), estaríamos hablando de un parámetro. El estadístico será solamente una aproximación de un parámetro. En cuanto más representativa sea la muestra el estadístico más se aproximará al parámetro. Las diferencias entre parámetro y estadístico las podemos resumir de la siguiente forma:

- ♦ Un estadístico es una característica de una pequeña parte de la población, es decir, una muestra. El parámetro es una medida fija que describe la población objetivo.
- ♦ El estadístico es variable, depende de la muestra de la población; en cambio, el parámetro es un valor numérico fijo y desconocido.
- ♦ Las notaciones para parámetro y estadístico son diferentes. Los parámetros son representados utilizando letras griegas, por ejemplo, la letra μ para la media, σ para la desviación estándar es σ , etc. Por su parte, los estadísticos se representan de forma distinta. $\bar{x}$ representa la media como estadístico, la letra s representa la desviación estándar muestra, etcétera.

El siguiente paso en nuestro estudio para evaluar la eficacia de un nuevo tratamiento para la diabetes es la incorporación de los que cumplen con los criterios de inclusión. Posteriormente daremos seguimiento de la evolución de los pacientes y finalmente mediremos si hubo una reducción o no de la glucosa en ayuno para evaluar eficacia. La recolección de datos y el diseño de instrumentos de recolección se escapa de los objetivos de este libro y, por tanto, no se explicará aquí. Sin embargo, el primer paso después de concluido el estudio es presentar y dar a conocer los datos obtenidos para todos los pacientes. Es imposible hacerlo de uno por uno, por lo tanto, debemos de emplear técnicas que nos permitan resumir y presentar de forma adecuada nuestros datos. Para ello, recurrimos a la **estadística descriptiva**, la cual tiene como objetivo presentar y organizar los datos; es decir, la estadística descriptiva nos permite simplificar y comprender una gran cantidad de datos mediante la generación de tablas, gráficas y la obtención de algunos parámetros que describen a estos datos.

Dado que nosotros trabajamos con una muestra, es de nuestro interés trasladar los resultados obtenidos a la población. En este sentido la **Estadística inferencial** nos permite tomar decisiones a partir de una cierta cantidad de datos (muestra) y trasladarlo a una población (totalidad de los datos). Se podría decir que el primer paso es presentar, organizar y entender los datos (estadística descriptiva) y después tomar decisiones (hacer inferencias) utilizando la estadística inferencial. El diagrama de la [figura 3.1](#) ejemplifica el significado de la Estadística inferencial y de cómo los resultados obtenidos en una muestra pueden ser trasladados a una población.

FIGURA 3.1. ESTADÍSTICA INFERENCIAL



Etapas en el análisis estadístico

Aunque las etapas descritas en el apartado anterior se podría inferir cuáles son los pasos para la realización del análisis estadístico, creemos conveniente profundizar un poco más en estos puntos. Para los autores de este libro, lo que más ha funcionado para la realización de un análisis estadístico son los siguientes pasos:

1. **Definición de los objetivos:** cualquier investigación y, por tanto, cualquier análisis estadístico requiere de una definición clara sobre lo que quiere investigar. Se tiene que tener bien claro qué es lo que se busca. ¿Estamos interesados en conocer una relación? ¿Evaluar un riesgo? ¿Predecir un resultado? ¿Comparar una variable? Si no tiene claro qué es lo que quiere realizar, difícilmente podrá realizar un análisis adecuado. En el ejemplo que describimos en la sección anterior, partimos de un objetivo. Nuestro interés era evaluar la eficacia de un nuevo tratamiento para tratar la diabetes.
2. **Diseño del estudio:** en bioestadística no se puede separar el diseño del estudio del análisis estadístico y de la conclusión. Es un aspecto fundamental que además va de la mano con los objetivos. No puede concluir ni proponer pruebas estadísticas de la misma manera para un estudio transversal que para un estudio de cohorte o para un ensayo clínico. Además, del diseño del estudio dependerá la selección y definición de variables. En su estudio sobre el tratamiento de pacientes con diabetes, se eligió un ensayo clínico, dado que, según las circunstancias, el objetivo y la epidemiología es lo más adecuado. El diseño del estudio, ventajas y desventajas no se abordarán en libro, pero sí estaremos constantemente refiriéndonos al diseño del estudio.
3. **Recolección de datos:** la recolección de datos depende de tener instrumentos adecuados que aseguren la precisión y la consistencia de los datos recogidos. En este texto obviaremos este punto y nos centraremos en los siguientes. Sin embargo, sí abordaremos las diferentes técnicas de muestreo que permiten en cierta medida asegurar una correcta recolección de los datos.
4. **Exploración de datos:** aunque las técnicas de exploración de datos son más utilizados para el contexto de *machine learning* y ciencias de datos,

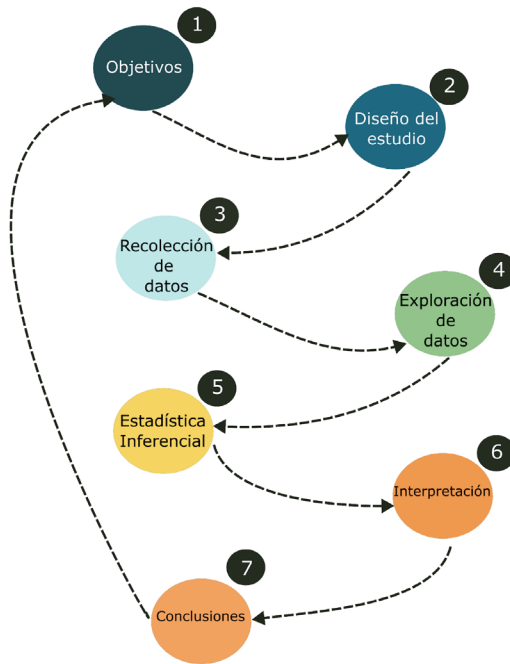
en este texto adoptaremos una versión simplificada de estas. El análisis exploratorio de datos es un paso inicial que viene después de la recolección de los datos y que va antes de la realización de pruebas de hipótesis. Nosotros nos centraremos en los siguientes pasos:

- » Visualización de datos: nos será útil para obtener información gráficamente de los datos, identificar patrones, errores, etc. Es mucho más fácil visualizar los datos que leerlos en tablas.
 - » Estadística descriptiva: nos ayudará para resumir y presentar la información contenida en los datos.
 - » Identificar datos atípicos y errores: en análisis exploratorio de datos puede ayudarnos a identificar errores de nuestro *Dataframe* y valores atípicos. Es común que los *Dataframe* contengan datos mal capturados. Imagine que identifica que la edad de uno de los pacientes con diabetes mellitus tiene 150 años.
 - » Formulación de hipótesis: aunque ya haya establecido un objetivo al inicio de su análisis estadística, el entendimiento de los datos que le puede dar la exploración le permitirá formular nuevas hipótesis que originalmente no fueron formuladas. El objetivo principal del análisis exploratorio de datos es conocer mis datos. **No puede hacer un análisis estadístico sin conocer sus datos.**
5. **Estadística inferencial:** para poder trasladar los resultados obtenidos en nuestra muestra pacientes evaluados en el ensayo clínico debemos de realizar inferencias. Principalmente se realizan pruebas de hipótesis, se utilizan intervalos de confianza o nos apoyamos de los enfoques bayesianos para este fin. La mayoría del contenido de este libro se enfoca en la estadística inferencial.
 6. **Interpretación de resultados:** finalmente debemos de ser capaces de interpretar los resultados obtenidos en las pruebas de hipótesis u otro enfoque para tomar decisiones. Por ejemplo, debemos de interpretar los resultados para concluir si el tratamiento es eficaz o no. Los autores de este libro creemos firmemente que la interpretación correcta de los resultados es un aspecto fundamental e indispensable. La intención de este libro no es formar técnicos de datos, si no que el lector sea capaz de interpretar adecuadamente cada número.

7. **Conclusiones:** en el mejor de los casos, una vez terminado con nuestro análisis estadístico, llegaremos a conclusiones. Aunque siempre existe la posibilidad de volver a regresar al principio.

La [figura 3.2](#) muestra una versión simplificada de los pasos del análisis estadístico. Estaremos haciendo referencia constantemente a ella. Sin embargo, antes explicaremos los diferentes tipos de variables.

FIGURA 3.2. ETAPAS DEL ANÁLISIS ESTADÍSTICO



Tipos de variables

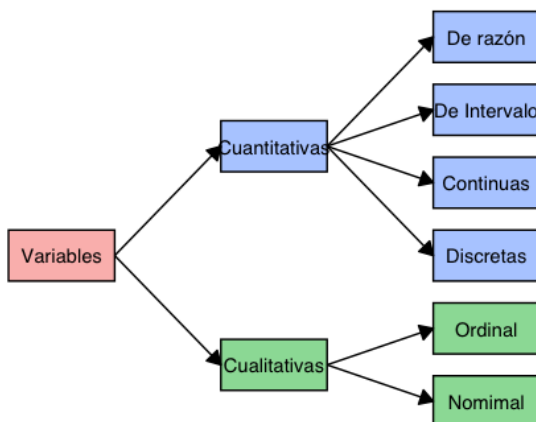
La materia prima de la Estadística y de la Bioestadística son los datos, precisamente el principal objetivo de estas disciplinas es procesar los datos para obtener información. En este libro vamos a trabajar con datos, los vamos a procesar, entender,

presentar, y con ellos vamos a llegar a conclusiones. Los datos representan características, cuando estas características pueden ser medidas e identificadas, y, además, no son constantes; en los pacientes podemos hablar de **variables**.

Una variable es cualquier característica, número o cantidad que puede ser medida o identificada. Las variables son fundamentales en el proceso de recopilación de datos y análisis estadístico, ya que representan los diferentes aspectos y dimensiones de los datos que se están investigando. Pueden tomar diferentes valores de un individuo a otro, o en diferentes momentos o lugares para el mismo individuo. Por ejemplo, en el estudio de la eficacia del nuevo tratamiento, usted probablemente recolectó datos (variables) que se refieren a números, como la edad, el tiempo de duración de la enfermedad o la concentración de glucosa en ayunas. Pero hay otras características que ofrecen una cualidad de los pacientes, por ejemplo, si el paciente tiene retinopatía o no. Por tanto, las variables se pueden clasificar. De esta clasificación dependerá la forma en la que procesemos, analicemos y concluyamos.

La manera más amplia en la que podemos clasificar a las variables es cualitativa y cuantitativa. Una variable cualitativa representa una cualidad o característica del individuo, por ejemplo, la variable que indica si el paciente solamente recibió un placebo o alguno de los tratamientos. En cambio, una variable cuantitativa es un poco más fácil de definir. Las variables cuantitativas son aquellas que se expresan como valores numéricos. Sin embargo, la clasificación es más amplia que lo descrito anteriormente. La [figura 3.3](#) muestra una visión general de los tipos de variables.

FIGURA 3.3. TIPOS DE VARIABLES



La siguiente lista muestra los principales tipos de variables, al final de cada descripción se indica si esta es cualitativa o cuantitativa.

- ♦ **Binaria:** este es el tipo de variable más simple, con solo dos respuestas posibles. Por ejemplo, fumador y no fumador; paciente y control (*cualitativa*).
- ♦ **Nominal:** este tipo de datos de cadena de caracteres es una versión más general de una variable binaria y tiene un número fijo de respuestas posibles que **no pueden ordenarse de manera útil**. Suelen estar codificadas alfanuméricamente establecidas por el investigador, por ejemplo: casado, soltero, divorciado, viudo (*cualitativa*).
- ♦ **Ordinal:** las posibles respuestas para este tipo de datos de cadena de caracteres están ordenadas linealmente. Podemos saber que un dato es mayor que otro, pero no podemos decir exactamente cuánto. Estas variables se pueden ordenar, por ejemplo: normopeso, sobrepeso y obesidad. Sabemos que el paciente con obesidad pesa más que el paciente con sobrepeso y este a su vez más que el paciente con normopeso. A pesar de que existen rangos para definir esta variable, no podemos decir con exactitud cuanto más o cuanto menos es una de las categorías (*cualitativa*).
- ♦ **Discreta:** variables numéricas que no pueden dividirse, por ejemplo, la edad medida en años o el número de pacientes (*cuantitativa*).
- ♦ **Continuas:** datos numéricos cuyo número entero está seguido por decimales, por ejemplo, el peso medido en kg (*cuantitativa*).

Las variables cuantitativas también pueden clasificarse según tengan o no en su escala un valor de cero absoluto:

- ♦ **Intervalo:** no incluye el cero absoluto, por ejemplo la medición de la temperatura en grados Celsius (*cuantitativa*).
- ♦ **De razón:** incluyen el cero absoluto, por ejemplo, la medición de la temperatura en grados Kelvin (*cuantitativa*).
- ♦ **Indexación:** por lo general, son nombres, etiquetas, números de caso o números de serie que identifican a un encuestado o grupo de encuestados. Esta variable, aunque podría clasificarse como cualitativa, solamente se utiliza para identificar los datos en un *Dataframe*.

Para R el tipo de variables es un tanto distinta al de la lista, pero este tema lo abordaremos en el [capítulo 4](#).

Ejercicios

- ♦ **Ejercicio 3.1:** ¿cuál es el objetivo de la Bioestadística?
- ♦ **Ejercicio 3.2:** diga al menos tres diferencias entre estadística descriptiva y estadística inferencial.

Los siguientes ejercicios tienen como objetivo reforzar la identificación de variables.

- ♦ **Ejercicio 3.3:** utilizando la siguiente lista de variables realice una clasificación de las mismas en cualitativa y cuantitativa, indicando, además, a qué tipo de variable corresponde dentro de esta clasificación: variables cualitativas (nominal, dicotómica u ordinal) y cuantitativas (discretas, continuas, de intervalo o de razón)
 - » Edad de un grupo de infantes expresada en meses
 - » Índice tabáquico. Es una variable que indica el riesgo de padecer complicaciones derivadas de fumar, como se muestra a continuación:

ÍNDICE	CLASIFICACIÓN
< 10	Sin riesgo
10-20	Riesgo moderado
21-41	Riesgo intenso
41>	Alto riesgo

- » Escala del dolor medida como mucho, poco o nada
- » Estado civil de los pacientes
- » Presencia o ausencia de comorbilidades
- » Índice de masa corporal
- » Medición de la actividad de la enfermedad con un índice que tiene puntajes que van desde 0 hasta 105. Solamente se pueden tener números enteros
- » Número de embarazos

- » Glucosa en ayunas expresada en mg/dL
- » Las reacciones adversas a los medicamentos se clasifican en reacciones de Tipo A, Tipo B, Tipo C, Tipo D, Tipo E y Tipo F. Una definición operativa de éstas se muestra enseguida:
 - Tipo A: son las más frecuentes y son dosis dependientes.
 - Tipo B: son menos frecuentes que las anteriores, no son dosis dependientes y se deben a otros mecanismos, como los inmunológicos o farmacogenéticos. Además, tienen alta mortalidad.
 - Tipo C: aparecen después de la administración prolongada de un fármaco. Son predecibles.
 - Tipo D: son poco frecuentes y aparecen un tiempo después de suspender o terminar la administración del fármaco.
 - Tipo E: aparecen después de suspender de forma brusca el fármaco.
 - Tipo F: no se deben al fármaco sino a otros componentes del medicamento.
- ♦ **Ejercicio 3.4:** el siguiente ejercicio muestra la descripción de la base de datos `birthwt`. De acuerdo con esta descripción, clasifique a cada una de las variables de la lista.

La base de datos `birthwt` tiene 189 filas y 10 columnas. Los datos se recopilaron en el Baystate Medical Center, Springfield, Mass durante 1986. Las variables:

 - » `low`: indicator of birth weight less than 2.5 kg.
 - » `age`: mothers age expressed years.
 - » `lwt`: mothers weight expressed pounds at last menstrual period.
 - » `race`: mothers race (1 = white, 2 = black, 3 = other).
 - » `smoke`: smoking status during pregnancy.
 - » `ptl`: number of previous premature labours.
 - » `ht`: history of hypertension.
 - » `ui`: presence of uterine irritability.
 - » `ftv`: number of physician visits during the first trimester.
 - » `bwt`: birth weight expressed grams.
- ♦ **Ejercicio 3.5:** el siguiente ejercicio muestra una descripción de la base de datos `Pima.tr`. Clasifique las variables de la lista. La base de datos `Pima.tr` contiene los datos de una población de mujeres que tenían al menos 21 años de edad, descendientes de indígenas pima y que vivían cerca de

Phoenix, Arizona, se sometieron a pruebas de diabetes de acuerdo con los criterios de la Organización Mundial de la Salud. Los datos fueron recopilados por el Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales de Estados Unidos. Utilizamos los 532 registros completos después de descartar los datos (principalmente faltantes) sobre la insulina sérica. Las variables de este *Dataframe* son:

- » npreg: number of pregnancies.
- » glu: plasma glucose concentration at an oral glucose tolerance test.
- » bp: diastolic blood pressure (mm Hg).
- » skin: triceps skin fold thickness (mm).
- » bmi: body mass index (kg/m^2).
- » ped: diabetes pedigree function.
- » age: age expressed years.
- » type: Yes or No, diagnostic diabetic according to WHO criteria.

- ♦ **Ejercicio 3.6:** realice una tabla comparativa entre Estadística y Bioestadística.
- ♦ **Ejercicio 3.7:** realice una tabla comparativa entre parámetro y estadístico.

Referencias

- Celis de la Rosa, A. y Labrada, V. (2014). *Bioestadística*. Manual Moderno.
- Cataldo, R., Arancibia, M., Stojanova, J. y Papuzinski, C. (2019). General concepts in biostatistics and clinical epidemiology: Observational studies with cross-sectional and ecological designs. Conceptos generales en bioestadística y epidemiología clínica: estudios observacionales con diseños transversal y ecológico. *Medwave*, 19(8), e7698. <https://doi.org/10.5867/medwave.2019.08.7698>
- Cataldo, R., Arancibia, M., Stojanova, J. y Papuzinski, C. (2019). General concepts in biostatistics and clinical epidemiology: Observational studies with cross-sectional and ecological designs. Conceptos generales en bioestadística y epidemiología clínica: estudios observacionales con diseños transversal y ecológico. *Medwave*, 19(8), e7698. <https://doi.org/10.5867/medwave.2019.08.7698>
- Dalgaard, P. (2008). *Introductory Statistics with r*. 2nd ed. *Statistics and Computing*. Springer.

- Estrada, S., Arancibia, M., Stojanova, J. y Papuzinski, C. (2020). General concepts in biostatistics and clinical epidemiology: Experimental studies with randomized clinical trial design. Conceptos generales en bioestadística y epidemiología clínica: estudios experimentales con diseño de ensayo clínico aleatorizado. *Medwave*, 20(3), e7869. <https://doi.org/10.5867/medwave.2020.02.7869>
- Komorowski, M., Marshall, D. C., Saliccioli, J. D. y Crutain, Y. (2016). Exploratory Data Analysis. En *MIT Critical Data*. Springer. <http://www.ncbi.nlm.nih.gov/books/NBK543641/>
- MacFarland, T. W. (2024). Role of statistics for decision-making in biostatistics. En *Introduction to Data Science in Biostatistics: Using R, the Tidyverse Ecosystem, and APIs* (pp. 147-173). Springer. <https://doi.org/10.1007/978-3-031-46383-9>
- Martínez, M. Á., Sánchez-Villegas, A., Toledo, E. y Faulín, J. (2020). Introducción a los métodos de la epidemiología y la bioestadística. En M. Á. Martínez, A. Sánchez-Villegas, E. Toledo y J. Faulín, *Bioestadística Amigable* (pp. 1-13). Elsevier.
- Mukhiya, S. K. y Ahmed, U. (2020). *Hands-On Exploratory Data Analysis with Python*.
- Reiczigel, J., Marozzi, M., Fábíán, I. y Rózsa, L. (2019). Biostatistics for Parasitologists - A Primer to Quantitative Parasitology. *Trends in parasitology*, 35(4), 277-281. <https://doi.org/10.1016/j.pt.2019.01.003>
- Shahbaba, B. (2011). Data exploration. En *Biostatistics with R: An Introduction to Statistics Through Biological Data* (pp. 17-59). Springer. <https://doi.org/10.1007/978-1-4614-1302-8>
- Shahbaba, B. (2011). Introduction. En *Biostatistics with R: An Introduction to Statistics Through Biological Data* (pp. 1-16). Springer. <https://doi.org/10.1007/978-1-4614-1302-8>
- Stallings, E. (2023). Biostatistics for gastroenterologists. Part I. *Revista española de enfermedades digestivas*, 115(4), 157-159. <https://doi.org/10.17235/reed.2023.9494/2023>
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics with S. Fourth Edition*. Springer.

Laboratorio de R 2

Edsaúl Emilio Pérez Guerrero

Para R cada uno de los tipos de datos tiene propiedades muy distintas que lo distinguen de los otros; el tipo de dato entre otras cosas condicionará el tipo de operaciones que podemos hacer con ellos. Por ejemplo, no podemos estimar la media de una variable dicotómica que represente la presencia o ausencia de una enfermedad. A pesar de estas restricciones, R intenta coercionar (transformar) los datos hacia uno más complejo o más adecuado a la operación que estamos solicitando. Sin embargo, no todos los tipos de datos pueden ser coercionados a otros tipos de datos.

Este laboratorio de R tiene como objetivo:

- ♦ Describir las características y propiedades de los tipos de variables y datos para R
- ♦ Utilizar funciones que permiten conocer el tipo de variable
- ♦ Transformar los tipos de variables en otras
- ♦ Conocer los fundamentos sobre la coerción de datos en R

Tipos de datos en R

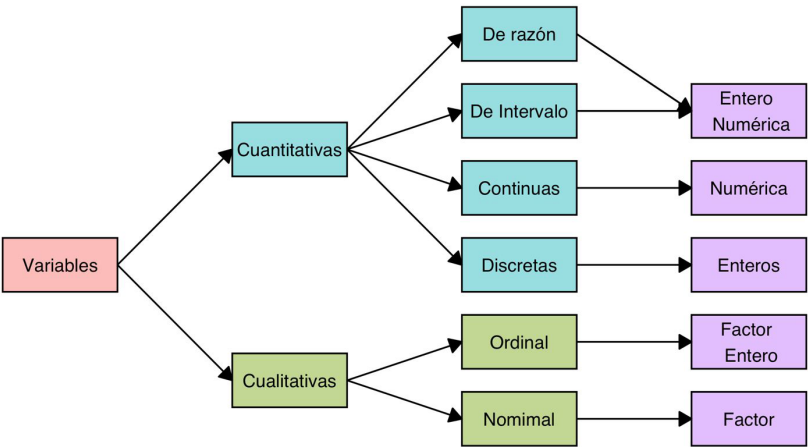
La [tabla 4.1](#) muestra los principales tipos de datos en R.

TABLA 4.1. TIPO DE DATOS EN R

TIPO DE DATO	NOMBRE	EJEMPLO	COMPROBAR	TRANSFORMAR
Entero	integer	4	is.integer()	as.integer()
Numérico	numeric	4.4000000000000004	is.numeric()	as.numeric()
Cadena	character	"4.4"	is.character()	as.character()
Factor	factor	cuatro	is.factor()	as.factor()
Lógico	logical	TRUE	is.logical()	as.logical()
Vacío	null	NULL	is.null()	NA
Perdido	NA	NA	is.na()	NA

En la [figura 4.1](#) se muestra los tipos de variables y su equivalente para R.

FIGURA 4.1. TIPOS DE VARIABLES Y DATOS EN R



A continuación, describiremos más a detalle cada uno de los tipos de datos en R.

Datos numéricos (numeric) y enteros (integer)

Ambos tipos de datos se refieren a datos que contienen números. La principal diferencia radica en que los datos enteros (*integer*) no incluyen decimales ni punto decimal. En cambio, el dato numérico puede tener una fracción decimal. Un dato numérico es el equivalente a una variable cuantitativa de tipo continua y un dato entero sería el equivalente que una variable cuantitativa discreta.

Para R ambos datos son similares y para fines prácticos para nosotros también lo serán. Podemos hacer el mismo tipo de operaciones con datos numéricos y datos enteros.

Podemos comprobar si cierto tipo de dato es numérico con la función:

```
is.numeric()
```

Esta función nos dará como resultado TRUE si el dato u objeto es un dato de tipo numérico y FALSE si no lo es.

Algunos datos pueden ser convertidos en otro tipo de dato, siempre y cuando la coerción sea correcta (ver [Coerción de datos](#)). Para convertir cierto tipo de dato en numérico utilice la función:

```
as.numeric()
```

Para evaluar un dato entero (*integer*), utilice las siguientes funciones: `is.numeric()` y para convertir a un dato entero: `as.numeric()`.

Cadenas de texto (character)

La cadena de texto en R es un dato que podemos identificar por estar entre comillas. Este tipo de dato admite números, texto, espacios y símbolos, siempre y cuando estén entre comillas. El uso de cadena de texto debería limitarse a variables de identificación, ya que la cantidad de operaciones que podemos hacer con ellos es limitada. Si, por ejemplo, tenemos una variable que indica si el paciente padece o no hipertensión, cuyos valores son “sí” y “no”, nos convendría más tener un factor (ver más adelante) que una cadena de texto. Ya

que en el factor podemos tener niveles y una jerarquía que indique cual valor representa la ausencia o presencia de una enfermedad. En una cadena de texto son indistintos.

Utilice la función `is.character()` para comprobar si su variable es un carácter. Utilice la función `as.character()` para convertir una variable a cadena de texto. La ventaja más grande de las cadenas de texto es que podemos convertir prácticamente cualquier dato en una secuencia de caracteres (cadena de texto).

Factores

Los factores en R son un tipo de dato utilizado para representar una variable categórica en la que existen niveles. Es decir, en los factores, hay un nivel 1, un nivel 2, etc. Por ejemplo, para la variable hipertensión cuyas opciones son “no” y “sí”, el nivel más bajo será “no” y el nivel más alto “sí”. Ya que, a menos que se indique lo contrario, los niveles se ordenarán siguiendo una jerarquía alfabética. Para la Bioestadística este orden es especialmente importante, ya que muchas veces nos interesa comparar un grupo de pacientes con otro grupo de individuos con la ausencia de la enfermedad, como el caso de los riesgos.

Los factores son especialmente útiles para manejar datos categóricos en análisis estadísticos y gráficos, porque aseguran que las operaciones estadísticas y las visualizaciones respeten la naturaleza categórica de los datos. Los factores son más complejos que los caracteres por que existe un orden. En la medida de lo posible todas las variables categóricas deberían de ser factores. Al menos así será en este libro.

Utilice las funciones `is.factor()` y `as.factor()` para este tipo de dato.

Datos lógicos

Otro tipo de dato muy útil en R es el dato lógico. Este solo puede tener dos valores: `TRUE` y `FALSE`. Representan si una condición o estado se cumple (falso o verdadero).

Un dato lógico es la respuesta a diversas operaciones que representan condiciones. Por ejemplo, ¿es mi dato una cadena de texto? ¿La edad de los pacientes es mayor a 50 años? Además, los datos lógicos se utilizan bastante

como argumentos de las funciones de R, en donde indicaremos, por ejemplo, si queremos que la función para estimar la media `mean()` omita los valores perdidos `mean(x=edad, na.omit=TRUE)`.

Utilice las funciones `is.logical()` y `as.logical()` para este tipo de dato.

Datos vacíos y nulos

Supongamos que, al ingreso de un paciente a nuestra investigación, no contaba con la medición de glucosa basal. Este dato no puede ser capturado en una base de datos. El dato no está disponible. Este dato sería un NA; un dato dejado en blanco intencionalmente. Recuerde que no le fue medida la concentración de glucosa al inicio del estudio, por lo que no podemos tener ese dato.

En cambio, un dato de tipo NULL es un dato vacío, que representa la ausencia de datos. Aunque son muy similares, existen diferencias sustanciales. Un dato NULL aparece solo cuando R intenta recuperar un dato y no encuentra nada, mientras que NA es usado para representar explícitamente datos perdidos, omitidos o que por alguna razón son faltantes.

Si pedimos a R el valor de la concentración de glucosa en ayunas de una persona que no existe en la base de datos, obtendríamos un dato de tipo NULL. En cambio, si pedimos la concentración de glucosa del paciente que tuvo la medición tendremos un NA. En el primer caso el paciente no existe y en el segundo sí existe, pero el dato no fue dejado explícitamente como vacío.

Trabajar con NA puede ser un dolor de cabeza, debemos evitarlos en la medida de lo posible. Nunca olvide esta afirmación: **un NA siempre produce un NA.**

Coerción de datos

Los datos en R se pueden transformar o coercionar de un tipo de dato a otro. Cuando hacemos operaciones en R se intentará convertir el dato a otro que sea correcto para poder realizar la operación. Por ejemplo, si intentamos calcular el promedio de un conjunto de datos, R intentará convertir los datos en numéricos, para poder realizar esta operación. Sin embargo, muchas veces obtenemos un error, ya que el dato no pudo ser convertido a uno numérico. Este error se presentam ya que la coerción tiene ciertas restricciones.

La coerción se da en el siguiente orden, pero no al revés:

```
lógico → entero → numérico → cadena de texto
(logical → integer → numeric → character)
```

Los datos pueden ser coercionados del dato más restrictivo a los más flexibles. Puede encontrar un buen artículo sobre coerción en el siguiente [blog](#).

Es importante que se dé tiempo para conocer la coerción de los datos utilizados en R. Le podrá ahorrar muchos dolores de cabeza. Los errores más problemáticos muchas veces se deben a una mala coerción de los datos.

Importar un Dataframe a R

Definición de un Dataframe

En este libro trabajaremos con estructuras de datos ordenados. Estas estructuras reciben el nombre base de datos o *Dataframe* (del inglés marco de datos). Aunque el término correcto en español es “base de datos”, en este libro utilizaremos constantemente *Dataframe*, ya que esto permitirá al lector familiarizarse con los términos utilizados en R y otros lenguajes de programación.

Un *Dataframe* organiza la información en columnas, que representan variables, y en filas, que corresponden a los casos individuales.

Por ejemplo, las [figura 4.2](#) y la [tabla 4.2](#) ilustran la estructura de un *Dataframe*. En particular, la [figura 4.2](#) muestra un segmento de un *Dataframe* que incluye datos clínicos de un grupo de pacientes. Cada columna representa una variable específica, como la edad, el peso y la altura de los pacientes. Las filas, por otro lado, corresponden a distintos pacientes o casos, mostrando en esta imagen los datos de seis pacientes, destacando los casos 1 y 4. La primera columna, *No_Caso*, sirve como una variable de indexación, que facilita la identificación y el acceso ordenado a la información de cada paciente. Las variables de indexación no se utilizan para realizar cálculos, sino que ayudan a identificar casos o pacientes específicos. En esta figura,

también se pueden observar valores perdidos NA en la columna de edad y en la de diabetes.

FIGURA 4.2. EJEMPLO DE UN DATAFRAME

Columnas (variables)

Datos

Variable de indexación

Valores perdidos

No. Caso	Nombre_Paciente	Expdiente	Edad	Peso	Talla	Diabetes
1	Moises Quintero	A12345	50	80	172	Si
2	Mia Castellano	A12346	52	102	163	No
3	Irati Oliveira	A12347	51	97	170	NA
4	Mar Barrera	A12348	44	112	152	Si
5	Antonio Vilches	A12349	48	76	168	Si
6	Alex Villena	A12350	NA	75	176	No

TABLA 4.2. OTRO EJEMPLO DE UN DATAFRAME

NPREG	GLU	BP	SKIN	BMI	PED	AGE	TYPE
5	86	68	28	30.2	0.364	24	No
7	195	70	33	25.1	0.163	55	Yes
5	77	82	41	35.8	0.156	35	No
0	165	76	43	47.9	0.259	26	No
0	107	60	25	26.4	0.133	23	No
5	97	76	27	35.6	0.378	52	Yes

De momento exploraremos dos opciones para la importación de *Dataframes* en R, la primera es cargar base de datos contenidas en paquetes y la segunda

es utilizar el menú de RStudio para importar bases de datos que se encuentran guardados en extensiones como *.txt*, *.xlsx*, *.csv* y *.sav*.

Importar base de datos de paquetes

Muchas de las librerías de R tienen base de datos que podemos utilizar para resolver los ejemplos de ayuda de las funciones de esa librería. En este compendio estaremos utilizando algunas de estas bases de datos.

Vamos a cargar el *Dataframe WeightLoss* de la librería *car*. Para ello podemos utilizar el siguiente código:

```
# Primero cargamos la librería que contiene el Dataframe
library(car)
Loading required package: carData

Attaching package: 'car'
The following object is masked from 'package:dplyr':
  recode
The following object is masked from 'package:purrr':
  some
# Después llamamos al Dataframe con la función data()
data("WeightLoss")# Tiene que ir entre comillas
```

Si prestamos atención, ahora en nuestro ambiente de trabajo tendremos un objeto llamado *WeightLoss*.

Si queremos visualizar esta base de datos, podemos utilizar la función *view()* y la base de datos se mostrará en el apartado superior izquierdo como lo muestra la [figura 4.3](#).

FIGURA 4.3. VISUALIZACIÓN DE UN DATAFRAME CON LA FUNCIÓN VIEW

	group	wl1	wl2	wl3	se1	se2	se3
1	Control	4	3	3	14	13	15
2	Control	4	4	3	13	14	17
3	Control	4	3	1	17	12	16
4	Control	3	2	1	11	11	12
5	Control	5	3	2	16	15	14
6	Control	6	5	4	17	18	18
7	Control	6	5	4	17	16	19
8	Control	5	4	1	13	15	15
9	Control	5	4	1	14	14	15
10	Control	3	3	2	14	15	13
11	Control	4	2	2	16	16	11
12	Control	5	2	1	15	13	16

Showing 1 to 12 of 34 entries, 7 total columns

Un *Dataframe* es un objeto que a su vez tiene variables que pueden ser consideradas como objetos independientes. Si queremos acceder a los objetos (variables) de nuestra base de datos debemos utilizar la función `attach()`, de esta forma podemos acceder a las variables de nuestra base de datos y hacer operaciones con ellas.

Por ejemplo:

```
attach(WeightLoss)
```

Ahora utilizando la variable `wl1` puedo estimar la media del peso al primer mes de un grupo de ratones.

```
mean(wl1)
[1] 5.294118
```

Si utilizamos la función `mean(wl1)` sin haber adjuntado nuestra base de datos, R nos dará el siguiente error:

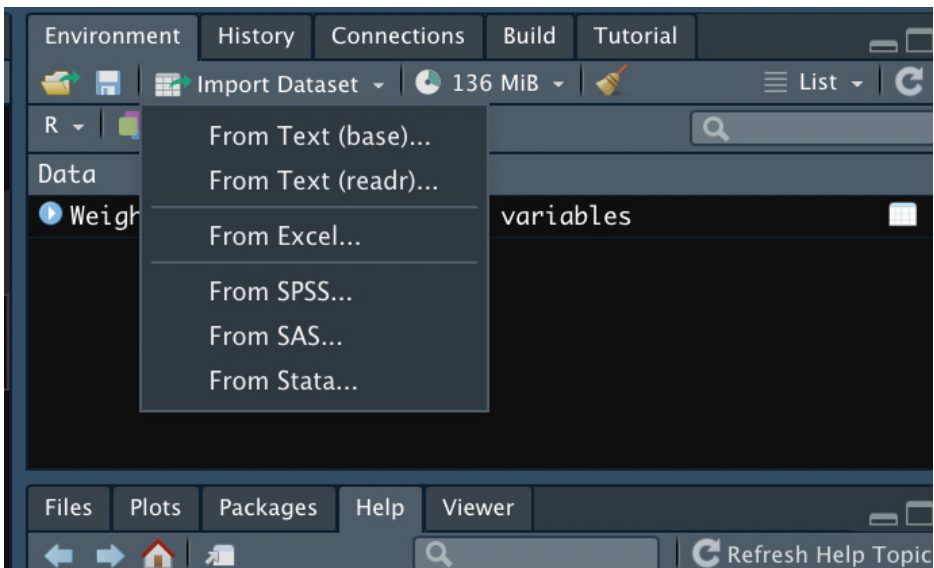
```
Error in mean(wl1) : object 'wl1' not found
```

Importar base de datos desde el menú de RStudio

La [figura 4.4](#) muestra el menú de RStudio con el que podemos importar bases de datos con los siguientes formatos:

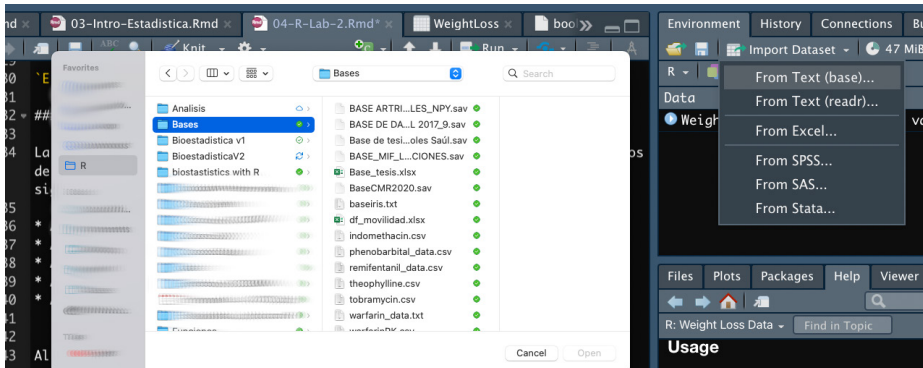
- ♦ Archivos de texto, formato .txt y .csv
- ♦ Archivos desde excel, formato .xlsx
- ♦ Archivos de SPSS, formato .sav
- ♦ Archivos de SAS, formato .sas
- ♦ Archivos de STATA

FIGURA 4.4. IMPORTAR BASES DE DATOS DESDE EL MENÚ DE RSTUDIO



Al hacer clic en cualquiera de las opciones, lo primero que debemos de hacer es seleccionar el archivo que queremos importar ([figura 4.5](#)).

FIGURA 4.5. BUSCAR ARCHIVOS PARA IMPORTAR UTILIZANDO EL MENÚ DE RSTUDIO



Archivos .txt y .csvs

Las siguientes líneas explican a grandes rasgos como importar archivos de texto y separados por comas. Ambos se pueden exportar desde el menú: Import Dataset → From Text (base). En las últimas versiones de RStudio importar cualquiera de los tipos de archivos no debería llevar mayor complicación y RStudio debería reconocer el tipo de separador correcto de los archivos. Sin embargo, el separador común para los archivos .txt es *white space* (ver [figura 4.6](#)) y para los archivos de .csvs es *comma* (ver [figura 4.7](#)). Si tenemos el separador correcto, nuestra base de datos se previsualizará en un rectángulo o cuadrado en donde las variables están perfectamente alineadas como lo muestran las figuras. Si tiene el separador incorrecto, su base se previsualizará como lo muestra la [figura 4.8](#).

FIGURA 4.6. IMPORTAR ARCHIVOS .TXT CON EL SEPARADOR CORRECTO

Import Dataset

Name: indomethacin

Input File: id,time,amount,y

Encoding: Automatic

Heading: ☒ Yes ☐ No

Row names: Automatic

Separator: Comma

Decimal: Period

Quote: Double quote (")

Comment: None

na.strings: NA

☐ Strings as factors

Data Frame

id	time	amount	y
1	0.00	1	.
1	0.25	.	1.5
1	0.50	.	0.94
1	0.75	.	0.78
1	1.00	.	0.48
1	1.25	.	0.37
1	2.00	.	0.19
1	3.00	.	0.12
1	4.00	.	0.11
1	5.00	.	0.08
1	6.00	.	0.07
1	8.00	.	0.05
2	0.00	1	.

Import Cancel

FIGURA 4.7. IMPORTAR ARCHIVOS .CVS CON EL SEPARADOR CORRECTO

Import Dataset

Name: baseiris

Input File: "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"

Encoding: Automatic

Heading: ☒ Yes ☐ No

Row names: Automatic

Separator: Whitespace

Decimal: Period

Quote: Double quote (")

Comment: None

na.strings: NA

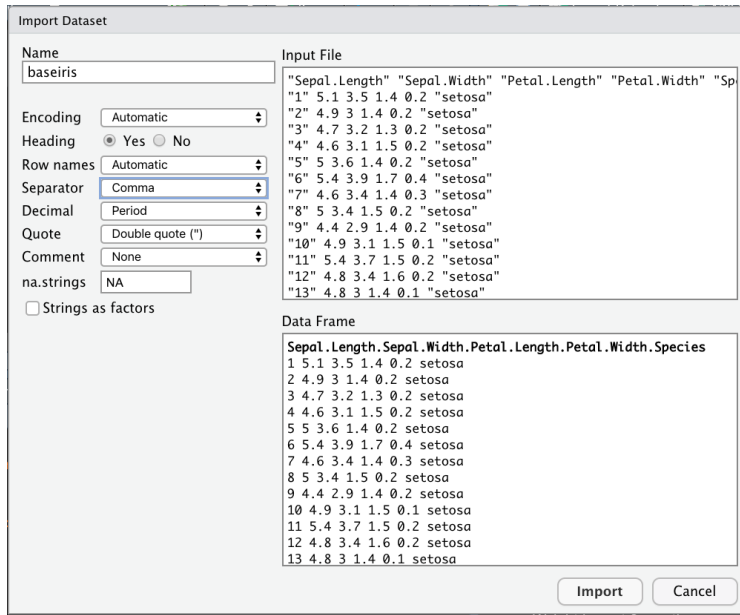
☐ Strings as factors

Data Frame

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa

Import Cancel

FIGURA 4.8. IMPORTAR ARCHIVOS CON UN SEPARADOR INCORRECTO



Puede visualizar algunos ejemplos en:

- ❖ [Cómo IMPORTAR archivos CSV a R](#)
- ❖ [Introducción R studio: importar datos de excel y texto](#)

Importar archivos desde Excel, SPSS o Sas, no requiere mayores complicaciones por lo que no se explicarán a fondo en este compendio. Solo es necesario seguir el menú de RStudio, es muy intuitivo. Sin embargo, puede consultar algunos ejemplos en:

- ❖ [Cómo cargar bases de datos de SPSS y Excel en R y RStudio](#)
- ❖ [Importar registros de Excel a RStudio | Paso a Paso](#)
- ❖ [Cómo IMPORTAR datos de EXCEL a R](#)
- ❖ [Importar datos SAS, STATA, SPSS a R](#)

Ejercicios

El objetivo de estos ejercicios es que, una vez que importamos una base de datos, podamos identificar el tipo de que está siendo utilizada por R.

Se muestra un ejemplo de cómo deberán ser resueltos los ejercicios de este capítulo.

Importe la base de datos `birthwt` de la librería `MASS` a su ambiente de trabajo. Esta base contiene información de variables asociadas al bajo peso en el nacimiento de algunos infantes. Para importar la base de datos a su ambiente de trabajo utilice el siguiente código (puede copiar y pegar):

```
library(MASS)
data("birthwt")
attach(birthwt)## Paso importante
```

Utilice la función de ayuda para obtener la información de las variables de la base `birthwt`.

```
help("birthwt")
```

La información que muestra la ayuda es aproximadamente la siguiente:

- ♦ `low`: indicator of birth weight less than 2.5 kg.
- ♦ `age`: mother's age in years.
- ♦ `lwt`: mother's weight in pounds at last menstrual period.
- ♦ `race`: mother's race (1 = white, 2 = black, 3 = other).
- ♦ `smoke`: smoking status during pregnancy.
- ♦ `ptl`: number of previous premature labours.
- ♦ `ht`: history of hypertension.
- ♦ `ui`: presence of uterine irritability.
- ♦ `ftv`: number of physician visits during the first trimester.
- ♦ `bwt`: birth weight in grams

Para comprobar la estructura de una base de datos en general puede utilizar la función `str` como sigue:

```
str(birthwt)
'data.frame':  189 obs. of  10 variables:
 $ low  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
$ age : int 19 33 20 21 18 21 22 17 29 26 ...
$ lwt : int 182 155 105 108 107 124 118 103 123 113 ...
$ race : int 2 3 1 1 1 3 1 3 1 1 ...
$ smoke: int 0 0 1 1 1 0 0 0 1 1 ...
$ ptl : int 0 0 0 0 0 0 0 0 0 0 ...
$ ht : int 0 0 0 0 0 0 0 0 0 0 ...
$ ui : int 1 0 0 1 1 0 0 0 0 0 ...
$ ftv : int 0 3 1 2 0 0 1 1 1 0 ...
$ bwt : int 2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

Podrá notar que existen algunas variables que no se encuentran correctamente clasificadas, por ejemplo: la variable `low` es reconocida por R como una variable numérica cuando de acuerdo con la descripción de la base debería ser un factor. Recuerde que podemos coercionar o transformar los datos para poder trabajar adecuadamente con ellos. Las siguientes líneas de código muestran cómo hacer esta transformación.

Primero verificamos de nueva cuenta con la función `class` el tipo de variable:

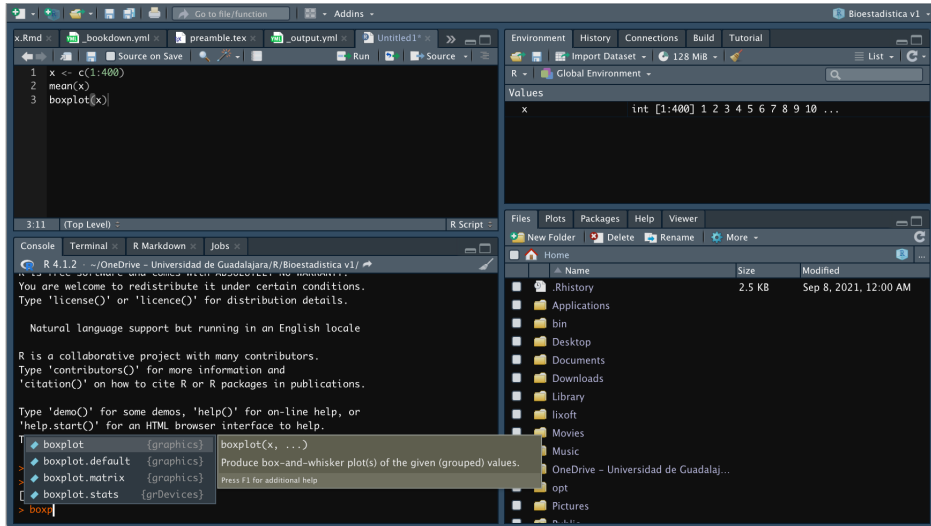
```
class(low)
[1] "integer"
```

Efectivamente es una variable numérica y debería de ser un factor. Podemos transformarla a factor con el siguiente código:

```
low <- as.factor(low)
```

Del código anterior podemos destacar algunos puntos. El primero es que sobrescribimos al objeto `low`. El segundo es que no sobre escribimos en la base `birthwt` sino en el objeto `low` que se creó después de adjuntar la base de datos, utilizando la función `attach`. Para entender este concepto podemos imaginar al objeto `birthwt` como un gran objeto que contiene a su vez muchos objetos dentro de él (cada una de las variables) (ver [figura 4.9](#)). Para acceder a estas variables una de las estrategias es utilizar la función `attach`.

FIGURA 4.9. ACCEDER A LAS VARIABLES DE UNA BASE DE DATOS



Ahora podemos comprobar el tipo de objeto que tenemos:

```
class(low)
[1] "factor"
```

- ♦ **Ejercicio 4.1:** repita estos pasos para cada uno de las variables que no se encuentran bien clasificadas en el *Dataframe* birtwht.
- ♦ **Ejercicio 4.2:** importe la base Pima.tr de la librería MASS y utilizando los pasos descritos en el ejercicio anterior identifique el tipo de variables para cada una de las variables contenidas en esta base de datos. De ser necesario cambie el tipo de dato para que R las lea de adecuadamente.

Resolución ejercicio 4.2

```
library(MASS)
data("Pima.tr")
str(Pima.tr)
```

No es necesario realizar modificaciones.

- ♦ **Ejercicio 4.3:** importe la base Melanoma de la librería MASS y utilizando los pasos descritos en el ejercicio anterior identifique el tipo de variables

para cada una de las variables contenidas en esta base de datos. De ser necesario cambie el tipo de dato para que R las lea de adecuadamente.

Resolución ejercicio 4.3

```
library(MASS)
data("Melanoma")
str(Melanoma)
Melano$status ← as.factor(Melanoma$status)
Melanoma$sex ← as.factor(Melanoma$sex)
Melanoma$ulcer ← as.factor(Melanoma$ulcer)
```

- ♦ **Ejercicio 4.4:** Importe la base Tipos de variables.rds que se encuentra en el [siguiente enlace](#). Esta base de datos se encuentra en formato .rds, para importar su base utilice la función readRDS() y después clasifique cada una de las variables que se encuentran ahí. De ser necesario cambie el tipo de dato para que R las lea de adecuadamente.

Resolución ejercicio 4.4

Si guardó la base de datos en su carpeta de bases de su ambiente de trabajo, puede utilizar el siguiente código:

```
df ← readRDS("Bases/Tipos de variables.rds")
str(df)
df$Genero ← as.factor(df$Genero)
df$Escolaridad ← as.factor(df$Escolaridad)
df$ocupacion ← as.factor(df$ocupacion)
df$Alcohol ← as.factor(df$Alcohol)
df$Tabaquismo ← as.factor(df$Tabaquismo)
df$toxicomania ← as.factor(df$toxicomania)
df$sedentarismo ← as.factor(df$sedentarismo)
df$HAS ← as.factor(df$HAS)
df$DM ← as.factor(df$DM)
df$Gdo_Escolaridad ← as.factor(df$Gdo_Escolaridad)
df$Trabajador ← as.factor(df$Trabajador)
```

Referencias

- Gandrud, C. (2013). *Learning R: A Step-by-Step Function Guide to Data Analysis*. O'Reilly Media, Inc.
- Long, J. y Teetor, P. (2019). *R Cookbook* (2nd ed.). <https://rc2e.com>
- Mailund, T. (2017). Introduction to R programming. En *Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist* (pp. 1-28).
- Moscarelli, M. (2023). Data types in 'R'. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 41-46). Springer Cham. <https://doi.org/10.1007/978-3-031-33073-5>
- R Core Team (2024). *R: A language and environment for statistical computing*.
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics*.
- Wickham, H. (2019). *Advanced R* (2nd ed.). Chapman & Hall/CRC The R Series.

Población y muestreo

Edsaúl Emilio Pérez Guerrero

En este capítulo se abordarán los conceptos de población y muestra. Además, revisaremos algunas de las técnicas de muestreo más utilizadas en Ciencias de la Salud. Aunque las definiciones de población y muestra fueron revisadas en el [capítulo 3](#), aquí profundizaremos en estos conceptos. Finalmente, realizaremos algunos ejercicios en R relacionados al muestreo.

Población

En estadística una población puede entenderse como el grupo objetivo del cual se eligen los sujetos de la muestra y para quienes se generalizarían los resultados. El grupo objetivo debe tener características previamente definidas y muy bien establecidas. Algunos ejemplos de población son:

- ♦ En un estudio sobre la prevalencia e incidencia de virus del papiloma humano en mujeres, la población podría ser todas las mujeres mayores a 35 años de la región sur de México.
- ♦ Para un programa de control del cáncer de próstata, la población objetivo podría ser todos los hombres de más de 45 años del occidente de México.
- ♦ Para estudiar los factores asociados al cáncer de mama, la población objetivo podría ser todas las mujeres mayores a 35 años que acudieron a revisión en un hospital de segundo nivel.

En los ejemplos anteriores, podemos notar que las poblaciones objetivo tienen características muy bien definidas que nos permiten delimitar a la población. Una población, por lo tanto, está formada por todas las unidades sobre las que se pueden aplicar los resultados de la investigación. En otras palabras, la población es un conjunto de todas las unidades que poseen características variables bajo estudio y para las cuales los resultados de la investigación pueden generalizarse.

El concepto de población no es perfecto y presenta algunos desafíos, por ejemplo:

- ♦ No hay garantía que los resultados del estudio puedan ser aplicados a casos futuros.
- ♦ “En medicina el concepto de población es más hipotético que real” (Indrayan y Malhotra, 2017). Sería imposible en la mayoría de los casos identificar a todos los pacientes con una característica en común.
- ♦ En el hipotético caso que se logre incluir todos los casos existentes de una población, sigue siendo una muestra. Ya que no se incluyen casos futuros.

En algunas ocasiones nos podemos referir a una **población finita** o **población infinita**. El primer concepto se refiere a una población en el que las unidades se encuentran perfectamente definidas y podemos estimar cuantas unidades conforman las misma, por ejemplo: todas las mujeres casadas mayores de 40 años del barrio del Santuario en Guadalajara Jalisco. Otro ejemplo de una población infinita son los pacientes que cumplan con el diagnóstico de diabetes mellitus tipo 2, con una edad de 40 a 60 años, que acudan a un hospital de segundo nivel y que hayan sido previamente tratados con insulina glargina. En cambio, una población infinita no puede ser contabilizada, por ejemplo: la cantidad de pacientes con diagnóstico de diabetes de México. Aunque existen estimaciones sobre la cantidad de pacientes con esta enfermedad en nuestro país, no es posible conocer con exactitud el número de personas afectadas con diabetes.

En Bioestadística, saber si se está trabajando con una población finita o infinita es crucial para varias razones, especialmente en el diseño del estudio y en la selección de las técnicas de análisis estadístico adecuadas. Además, conocer el tipo de población (finita o infinita) es especialmente útil para la determinación del tamaño de muestra, ya que se deberá hacer cambios dependiendo si se trata de una población finita o infinita.

Muestra y muestreo

En la [sección “Conceptos básicos”](#) ya se describió el concepto de muestra. Sin embargo, aquí nos enfocaremos en sus ventajas y desventajas.

♦ Ventajas:

- » Única opción viable de obtener datos en ciertos contextos: en algunos escenarios, el muestreo puede ser el único método viable para recolectar datos. Por ejemplo, en estudios epidemiológicos a gran escala donde no es posible examinar a toda la población. Además, económica y logísticamente sería imposible incluir todos los casos posibles.
- » Reducción de costos y uso de recursos: el muestreo reduce significativamente los costos y la demanda de recursos como personal y uso de laboratorio. Esto es crucial en investigaciones donde los fondos son limitados. Imagine el costo y la logística para determinar las concentraciones de glucosa en ayuno de todos los pacientes con el diagnóstico de diabetes del occidente de México.
- » Confiabilidad de la información: con métodos de muestreo apropiados, es posible obtener datos confiables y acercarnos a la realidad de la información.

♦ Desventajas:

- » Variabilidad entre muestras: cada muestra tomada de la misma población puede variar significativamente, lo que puede afectar la consistencia de los resultados. Toda muestra es diferente a otra. Esto es particularmente relevante en estudios que dependen de condiciones homogéneas entre los sujetos de prueba. Este problema se puede reducir si emplean método de muestreo adecuados
- » Limitaciones en la representatividad: no todas las muestras pueden llegar a ser representativas. Además, en muchas de las ocasiones algunos grupos pequeños no son representados de forma adecuada.

Muestreo

Conceptos básicos

Cuando se realiza muestreo debemos de tener claro a qué nos referimos con: unidad de muestreo, unidad de investigación, marco de muestreo y tamaño de muestra. Para poder comprender mejor estos conceptos, revisaremos un ejemplo: en un censo realizado por el Inegi (Instituto Nacional de Estadística y Geografía) sobre la desnutrición infantil, la unidad de muestreo podría ser una familia, pero la unidad de investigación podría ser un niño menor de 5 años. En otras palabras, la **unidad de muestreo** es de donde serán tomadas las **unidades de investigación**, es decir, cada uno de los casos seleccionados durante el muestreo. Una unidad de muestreo puede tener múltiples unidades o ninguna unidad sobre la cual investigar.

Los encuestadores del Inegi para realizar esta encuesta tuvieron acceso a una lista de todas las unidades de muestreo que son mutuamente excluyentes una de la otra, con ello nos referimos al **marco de muestreo**. La preparación del marco requiere una definición precisa de la unidad, así como de la población por lo que en el marco de muestreo se toma en cuenta los **criterios de inclusión y de exclusión**. Recuerde que la Bioestadística, la Metodología de la investigación y la Epidemiología siempre van de la mano.

Ahora bien, a los encuestadores les surge la siguiente pregunta: ¿qué cantidad de muestra es la adecuada para ser representativa de una población? La respuesta a esta pregunta es el **tamaño de muestra**. Este concepto se refiere al número de sujetos o unidades de muestreo mínimos para identificar diferencias. Debe ser lo suficientemente grande para responder nuestra pregunta de investigación y para permitirnos encontrar diferencias y/o asociaciones. Se utiliza la letra n para referirnos a ella.

Muestreo aleatorio y no aleatorio

Una muestra se denomina aleatoria cuando la inclusión o exclusión de un sujeto elegible en particular depende del azar y no se puede predecir de antemano.

♦ Para hacer un muestreo aleatorio necesitamos:

- » Un gran número de unidades de muestreo.
- » O conocer todos los individuos que reúnen con mis criterios de inclusión y exclusión

Por ejemplo, si los encuestadores del Inegi recurren a un muestro aleatorio, deberán de contar con un listado exacto de todos los habitantes del país y de ese listado seleccionar aleatoriamente a las unidades de muestreo o unidades de investigación a encuestar. Lo cual es poco factible.

Tener o no un muestreo aleatorio no debería ser el fin del mundo ni terminar con la validez de nuestro análisis estadístico. La selección aleatoria es solo una técnica para asegurar que una muestra sea representativa de una población. A medida que el tamaño de la muestra se incrementa en relación al tamaño total de la población, también lo hace la probabilidad de que la muestra refleje adecuadamente a la población, ya sea que la muestra sea aleatoria o no. Sin embargo, el uso de muestreo aleatorio proporciona probabilidades específicas que facilitan la realización de inferencias estadísticas más robustas y confiables, de tal manera que los “errores” se distribuyen aleatoriamente en las unidades de investigación. En otras palabras, mientras más grande sea la muestra, mejor será su capacidad para reflejar a la población; sin embargo, el muestreo aleatorio añade la ventaja de proporcionar una base sólida para análisis estadísticos debido a su naturaleza probabilística.

Técnicas de muestreo

En esta sección se revisarán los siguientes métodos de muestreo:

- ♦ Muestreo aleatorio simple
- ♦ Muestreo aleatorio estratificado
- ♦ Muestreo aleatorio multietapa
- ♦ Muestreo aleatorio por conglomerados (clústeres)
- ♦ Muestreo aleatorio sistemático
- ♦ Muestreo consecutivo
- ♦ Muestreo secuencial

Algunos de los tipos de muestreo no serán definidos en primera instancia, pero sí serán puestos en práctica.

Muestreo aleatorio simple

En el muestreo aleatorio simple es esencial que cada individuo o unidad de muestreo posea la misma probabilidad de ser seleccionado. Este método garantiza que todas las muestras, sin importar su tamaño, tengan igual probabilidad de ser elegidas. Sin embargo, una limitación significativa de este enfoque es que depende críticamente de la disponibilidad de un marco muestral completo.

Por ejemplo, suponga que quiere hacer un muestreo aleatorio en el que su población de interés son los estudiantes de posgrado de CUCS, para ello necesitará una lista de todos los estudiantes que cumplan con sus criterios (incluidos maestrías, doctorados y especialidades). Es decir, se necesita todo su marco de muestreo para seleccionar aleatoriamente a los estudiantes. La [figura 5.1](#) ejemplifica el muestreo aleatorio simple.

FIGURA 5.1. MUESTREO ALEATORIO SIMPLE



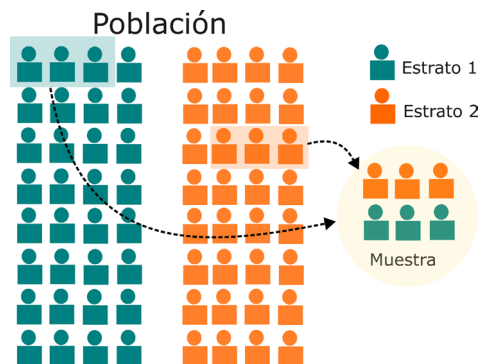
Muestreo aleatorio estratificado

Según la RAE, un estrato “es un conjunto de elementos que, con determinados caracteres comunes, se ha integrado con otros conjuntos previos”.¹ Ahora imagine que se realiza un estudio para evaluar los niveles de una hormona en mujeres. Se espera que la muestra sea representativa de todos los hospitales de segundo nivel de Guadalajara. ¿Cómo nos aseguramos de que todos los hospitales tengan una representación adecuada? Para ello, recurrimos al muestreo estratificado siguiendo los siguientes pasos:

- ♦ Primero, se realiza la identificación de los estratos (los hospitales).
- ♦ Después, se realiza un muestreo aleatorio para cada estrato.
- ♦ Se determina el tamaño de la muestra para cada estrato.
- ♦ Una muestra obtenida por muestreo aleatorio estratificado implica que la muestra de cada estrato está en la misma proporción que en la población.

En el muestreo por estratos, la población se divide en subgrupos o estratos homogéneos basados en una característica específica, como la edad, el género, la ubicación geográfica, el nivel socioeconómico o la condición de salud. Luego, se selecciona una muestra aleatoria de cada estrato. Para el ejemplo anterior los estratos podrían ser: mujeres menores a 20 años, mujeres de entre 20 y 40 años y mujeres mayores a 40 años. La [figura 5.2](#) ilustra el muestreo por estratos:

FIGURA 5.2. MUESTREO ALEATORIO ESTRATIFICADO



¹ <https://www.rae.es/desen/estrato#>

Otro ejemplo de muestreo por estratos podría ser el siguiente. Suponga que desea llevar a cabo un estudio para evaluar la prevalencia de la diabetes en una población de adultos mayores en una ciudad determinada. Sabe que esta población se compone de cuatro estratos en función de la ubicación geográfica: centro de la ciudad, suburbios, áreas rurales y otras áreas. Cada estrato tiene una población de adultos mayores con características socioeconómicas y de acceso a la atención médica ligeramente diferentes. Una vez identificado los estratos, deberá de seleccionar aleatoriamente un número determinado de pacientes de cada uno de los estratos, de tal manera que cada uno de ellos tenga representatividad.

Algunas de las ventajas para el muestreo por estratos son:

- ♦ Permite obtener estimaciones precisas para cada estrato, lo que es útil cuando se sabe que diferentes estratos tienen diferentes tasas o características de interés.
- ♦ Puede proporcionar una representación equitativa de cada estrato, lo que es útil para garantizar la inclusión de grupos minoritarios en la muestra.
- ♦ Facilita un análisis detallado de cada estrato por separado.

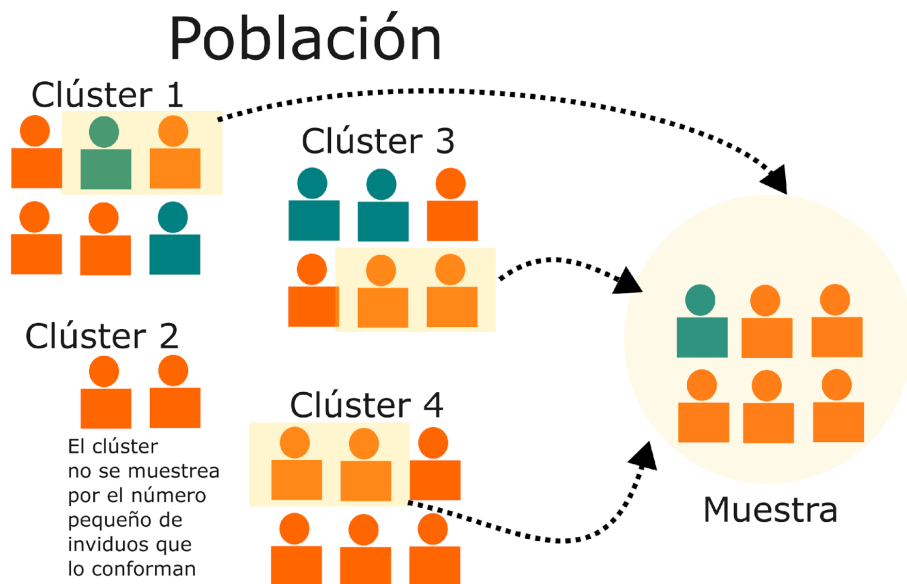
Mientras que algunas de sus desventajas son:

- ♦ Puede ser más costoso y requerir más tiempo que otras técnicas de muestreo.
- ♦ Requiere información precisa sobre la población en términos de estratos.

Muestreo aleatorio por clústeres (conglomerados)

El muestreo aleatorio por clústeres suele confundirse con el muestreo por estratos, sin embargo, a diferencia del análisis por estratos, el análisis por clústeres permite que ciertas unidades de muestreo, sobre todo cuando son pequeñas o reúnen ciertas características, no sean incluidas. Es decir, podemos no incluir algunas unidades de muestreo con tamaño pequeño. La [figura 5.3](#) ilustra el muestreo por clúster.

FIGURA 5.3. MUESTREO POR CLÚSTER



Quizá la diferencia más importante entre el muestreo por clústeres y por estratos es que el primero se aprovecha de divisiones ya hechas en la población, mientras que el segundo no. Además, el muestreo por clúster permite la no inclusión de ciertas unidades de muestreo.

Un ejemplo de muestreo por conglomerados podría ser: imagine que realiza un estudio para evaluar la prevalencia de enfermedades transmitidas por vectores, como el dengue y el zika, en una región rural de un país. En lugar de realizar un muestreo por estratos, donde dividirías la población en grupos basados en alguna característica específica, optas por el muestreo por clústeres debido a la falta de una lista completa y actualizada de todos los hogares en la región. Por lo tanto, realizas un muestreo por localidades, omitiendo a aquellas localidades pequeñas.

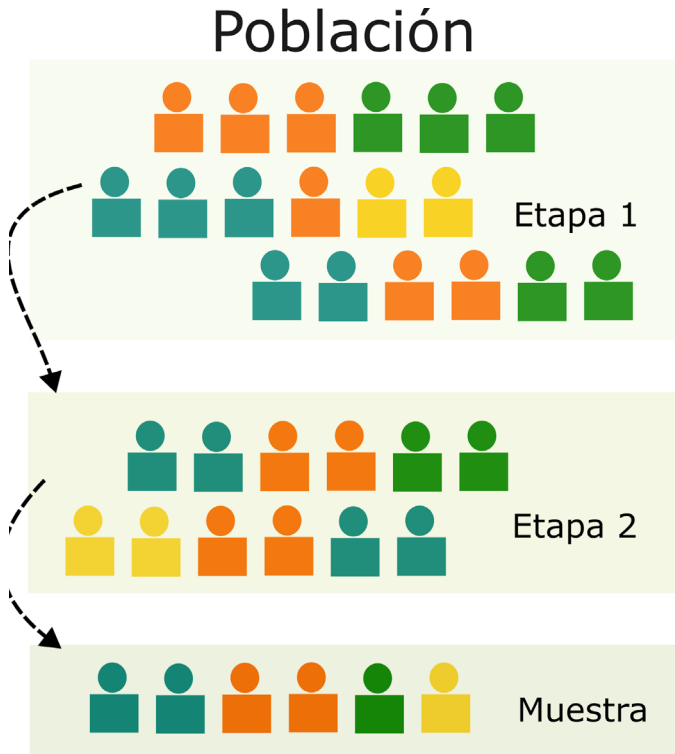
Muestreo aleatorio multietapa

Otro tipo de muestreo muy utilizado es el muestreo aleatorio multietapa, el cual, como su nombre lo dice, consiste en extraer una muestra por etapas. Por ejemplo, imaginemos que es de nuestro interés estimar la prevalencia de tabaquismo

en el estado de Jalisco. Para ello, se puede, por ejemplo, seleccionar primero 4 municipios del estado, luego 12 ciudades dentro de cada municipio seleccionado y luego 50 familias dentro de cada bloque seleccionado, todo por método aleatorio.

El muestreo multietapa es útil para poblaciones de gran tamaño que consiste en extraer las muestras por etapas y se realiza el muestreo de la unidad mayor a unidad menor.

FIGURA 5.4. MUESTREO ALEATORIO MULTIETAPA

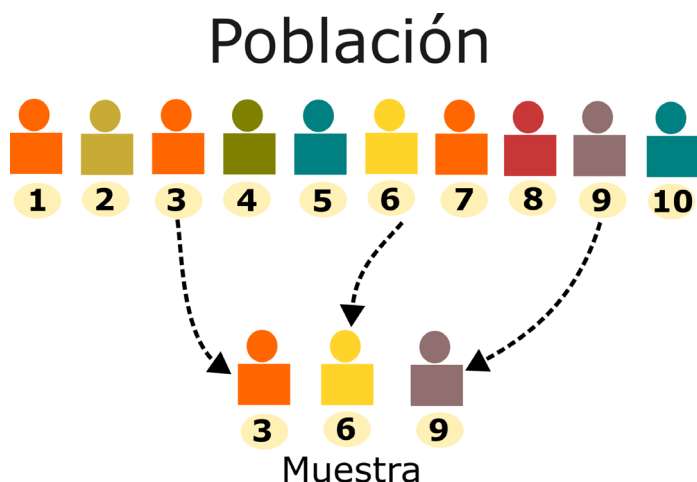


Muestro aleatorio sistemático

Imaginemos ahora que tenemos un grupo de niños de una escuela primaria en el que se encuentran ordenados por estatura. Nuestro interés es conocer el índice masa corporal promedio de los niños de esa primaria. Para poder

tener una muestra representativa de estos, se realizará un muestreo aleatorio sistemático. Este tipo de muestro consiste en seleccionar uno de los niños a azar (por ejemplo, mediante un muestreo aleatorio), posteriormente “sistemáticamente” selecciona el siguiente niño. Por ejemplo, podemos seleccionar un niño y medir su índice de masa corporal cada 20 niños. Es decir, después de 20 niños un niño será seleccionado. Con esto, nos aseguramos que por ejemplo no estemos haciendo nuestro muestreo con sesgo. Es decir, tendrías un error si solo seleccionamos a los niños más pequeños o a los más grandes (figura 5.5).

FIGURA 5.5. MUESTREO ALEATORIO MULTIETAPA



Casos consecutivos

Consiste en incluir a un individuo o paciente siempre y cuando cumplan con los criterios de inclusión en el momento que este acuda a consulta, centro de referencia, laboratorio etc. Es un tipo de muestreo no probabilístico, los individuos no tienen las mismas probabilidades de ser seleccionados. Si el paciente no va a consulta, no podrá ser seleccionado. En este tipo de muestreo se debe de evitar el sesgo de los días. Por ejemplo, en una consulta de un hospital de tercer nivel los lunes se revisan las pacientes con cáncer de mama que respondieron adecuadamente al tratamiento, mientras que los martes se revisan todos los casos nuevos. Si acudimos

a la consulta solamente los lunes, no podríamos seleccionar a las pacientes recién diagnosticadas y solo veríamos a las pacientes con respuesta al tratamiento. En las ciencias de la salud, este es uno de los muestreos más utilizados.

Muestreo secuencial

En el muestreo secuencial, los sujetos elegibles de la población objetivo se seleccionan uno a uno de manera aleatoria y se evalúan. El muestreo posterior se detiene tan pronto como se dispone de un resultado fiable en un sentido u otro. Por ejemplo, si es de nuestro interés asociar el índice de masa corporal con el riesgo de presentar infarto agudo al miocardio, detendremos el muestreo cuando logremos ver esta asociación. Este método de muestreo no es tan popular en medicina y se trata de un método no probabilístico.

Muestreo en R

El siguiente apartado de este capítulo describe algunas funciones útiles para realizar muestreo dentro de R. Veremos los siguientes puntos:

- ♦ Muestreo aleatorio utilizando la función `sample`
- ♦ Muestreo aleatorio en una base de datos
- ♦ Muestreo aleatorio sistemático
- ♦ Muestreo utilizando el paquete `dplyr`

Muestreo aleatorio simple en R

Para simular un muestreo aleatorio en R utilizaremos la función `sample()`. La función `sample()` en R es una herramienta versátil para realizar muestreos aleatorios de un conjunto de datos y en cierta manera obtener datos aleatorios. Esta función devuelve un número determinado de datos de un objeto y requiere al menos los siguientes argumentos:

- ♦ Un objeto del que vamos a extraer los datos
- ♦ La cantidad de datos que vamos extraer
- ♦ Un argumento lógico para indicar si se quiere remplazo

Un argumento es el parámetro(s) proporcionados a una función para realizar las operaciones para las que fue creada. Es decir, un argumento le dice a una función qué hacer y cómo hacerlo. En este libro utilizaremos constantemente la expresión “la función se alimenta de los siguientes argumentos”. Al decir esto, nos referimos a que la función necesita los argumentos mencionados.

Ejemplo 5.1: de una lista de 2 000 datos vamos a extraer 10 de ellos sin reemplazo

```
set.seed(4)# Función para que todos tengamos
los mismos datos. Sembramos la misma semilla
sample(x=1:2000, size=10, replace = FALSE)
[1] 1528 587 819 1795 71 684 371 757 698 307
```

El [ejemplo 5.1](#) requiere de la explicación de algunos puntos:

- ♦ La función `set.seed()` permite generar números “aleatorios” bajo ciertos parámetros. En R los números aleatorios en realidad no son aleatorios, sino pseudoaleatorios. Imagine que tiene un puñado de semillas, y no conoce su procedencia ni el árbol al que darán vida. Por lo tanto, si las siembra, el árbol que crecerá será un misterio. Ser pseudoaleatorio en lugar de aleatorio puro significa que R necesita conocer la procedencia de la semilla antes de plantar un árbol. La función `set.seed` permite modificar esta semilla e indicarle a R cual semilla seleccionar en lugar de seleccionar una al azar. Por ello, utilizaremos constantemente la siguiente analogía: “Vamos a sembrar una semilla”.
- ♦ Utilizando la función `sample` generamos de nuestro argumento `x` 10 números aleatorios (`size`) sin reemplazo. Explicado de otra forma, del argumento `x` que contiene una secuencia de números del 1 a 2 000 se seleccionaron al azar 10 número sin reemplazo. El argumento `replace` indica si la muestra será seleccionada sin o con reemplazo.

Muestreo sistemático en R

El [ejemplo 5.2](#) muestra cómo realizar muestreo sistemático utilizando R.

- ♦ **Ejemplo 5.2**


```
library(MASS) # Librería que contiene el Dataframe
data(Pima.tr2) # Llamar al Dataframe
Pima.tr2$ID ← 1:300 # Crear un objeto con una variable de indexación
set.seed(4) # Indicamos la semilla a sembrar
n ← sample(Pima.tr2$ID, size=20) # Seleccionar una muestra de 20 sin reemplazo
n+5 # Seleccionamos sistemáticamente
[1] 80 264 76 177 250 191 67 107 135 157 89 260 240 53 181 202 155
59 208[20] 85
```

Muestreo con otros paquetes

El [ejemplo 5.3](#) describe cómo hacer muestreo aleatorio utilizando la librería `dplyr`.

♦ Ejemplo 5.3

```
# Instalar el paquete dplyr
install.packages("dplyr")
# Otra alternativa de instalación
install.packages("devtools")
devtools::install_github("tidyverse/dplyr")
# Una vez instalada la librería podemos cagarla
library(dplyr) # Librería necesario
# Objeto con una muestra de 20 sin reemplazo:
muestra ← Pima.tr2 %>%
  sample_n(size=20, replace=F)
head(muestra) # Visualizar los primeros 6 datos de nuestro objeto
```

	npreg	glu	bp	skin	bmi	ped	age	type	ID
1	2	157	74	35	39.4	0.134	30	No	180
2	6	109	60	27	25.0	0.206	27	No	65
3	9	152	78	34	34.2	0.893	33	Yes	152
4	4	134	72	NA	23.8	0.277	60	Yes	246
5	0	167	NA	NA	32.3	0.839	30	Yes	288
6	2	112	86	42	38.4	0.246	28	No	126

No se preocupe si no comprende el código del ejemplo conforme avance en la lectura del libro podrá comprenderlo.

El [ejemplo 5.4](#) muestra cómo realizar un muestreo por proporción de casos.

Ejemplo 5.4

```
muestra2 <- Pima.tr2 %>%
  sample_frac(0.10)# Extraer el 10% de lo casos
head(muestra2)
```

	npreg	glu	bp	skinbmi	ped	age	type	ID	
1	8	133	72	NA	32.9	0.270	39	Yes	278
2	7	103	66	32	39.1	0.344	31	Yes	113
3	0	119	64	18	34.9	0.725	23	No	191
4	2	109	92	NA	42.7	0.845	54	No	228
5	4	123	62	NA	32.0	0.226	35	Yes	258
6	0	125	96	NA	22.5	0.262	21	No	291

Ejercicios

- ♦ **Ejercicio 5.1:** cree un objeto con todos los nombres de los alumnos de la clase. Extraiga una muestra aleatoria con reemplazo de 50.
- ♦ **Ejercicio 5.2:** función `sample()` aplicada a bases de datos. De la base de datos `pima.tr2` de la librería `mass` utilizando la variable `age`, seleccione al azar 20 datos.

Resolución [ejercicio 5.2](#)

```
library(MASS)# Cargar paquete
data("Pima.tr2") # Cargar Dataframe
sample(Pima.tr2$age, size=20)
```

[1] 24 29 58 24 33 57 60 22 23 32 25 63 27 41 21 22 40 28 46 23

Otra opción sería:

Resolución ejercicio 5.2

```
library(MASS)# Cargar paquete
data("Pima.tr2") # Cargar Dataframe
sample(Pima.tr2$age, size=20)

[1] 24 29 58 24 33 57 60 22 23 32 25 63 27 41 21 22 40 28 46 23
```

Otra opción sería:

```
# Otra forma de resolución
Pima.tr2$ID ← 1:300 # Creamos una variable de indexación
head(Pima.tr2) # Visualizar los primeros 6 datos de un data frame
```

	npreg	glu	bp	skin	bmi	ped	age	type	ID
1	5	86	68	28	30.2	0.364	24	No	1
2	7	195	70	33	25.1	0.163	55	Yes	2
3	5	77	82	41	35.8	0.156	35	No	3
4	0	165	76	43	47.9	0.259	26	No	4
5	0	107	60	25	26.4	0.133	23	No	5
6	5	97	76	27	35.6	0.378	52	Yes	6

```
sample(Pima.tr2$ID, size=20)

[1] 126 25 177 159 138 127 139 218 269 192 211 98 270 46 292 47 280 213 35
[20] 53
```

- ♦ **Ejercicio 5.3:** suponga que queremos aleatorizar a los participantes en dos grupos (A y B), y queremos aleatorizar una muestra de 50 pacientes.

Resolución ejercicio 5.3

```
sample(x=c("A","B"), size = 50, replace = T)
[1] "A" "A" "B" "B" "A" "B" "A" "A" "B" "A" "B" "B" "B" "B" "A" "B"
"B" "B" "B"
[20] "B" "A" "B" "A" "B" "B" "A" "A" "B" "B" "B" "A" "B" "B" "B" "B"
"A" "B" "B"
[39] "B" "A" "A" "A" "A" "B" "A" "A" "B" "A" "A" "A"
```

- ♦ **Ejercicio 5.4:** ¿por qué la muestra de pacientes no incluyó 25 pacientes en el grupo A y 25 pacientes en el grupo B? ¿Considera que es un muestreo aleatorio?

Resolución ejercicio 5.4

Si los datos son aleatorios, es muy difícil que se generen grupos iguales.

- ♦ **Ejercicio 5.5:** de la muestra anterior extraiga una segunda muestra que contenga 60% del total de los casos de la muestra anterior.

Resolución ejercicio 5.5

```
sample(x=c("A","B"), size = 50*.6, replace = T)
[1] "B" "A" "A" "B" "A" "B" "A" "B" "B" "B" "B" "B" "B" "B" "B" "B"
"B" "A" "A"
[20] "A" "A" "B" "B" "A" "B" "A" "A" "B" "A" "B"
muestra ← sample(x=c("A","B"), size = 50, replace = T)
ID ← 1:50
datos ← data.frame(ID, muestra)
library(dplyr)
muestra2 ← datos |>
  sample_frac(0.60)# Extraer el 60% de los casos
print(muestra2)
```

	ID	muestra
1	33	A
2	46	B
3	30	B
4	31	B
5	27	A

6	15	A
7	50	B
8	8	B
9	39	B
10	20	A
11	16	A
12	17	A
13	18	B
14	19	B
15	44	B
16	14	B
17	11	B
18	49	A
19	1	B
20	29	B
21	38	B
22	6	A
23	22	A
24	47	B
25	32	B
26	4	B
27	45	B
28	26	B
29	7	B
30	35	A

- ♦ **Ejercicio 5.6:** si su interés es tener una muestra con al menos 50 individuos en cada uno de los grupos, ¿qué estrategias podría seguir? ¿Su estrategia genera una muestra aleatoria?
- ♦ **Ejercicio 5.7:** un grupo de investigadores realiza una aleatorización de 30 pacientes con el objetivo de tener 15 individuos en el grupo A y 15 individuos en el grupo B. ¿Se puede considerar esta como una muestra aleatoria?

Resolución ejercicio 5.7

No

- ♦ **Ejercicio 5.8:** utilizando la base de datos `Pima.tr2` obtenga una muestra de pacientes con diabetes de 20.

Resolución ejercicio 5.8

```
library(MASS)
data(Pima.tr2)
Pima.tr2$ID <- 1:300
# Filtrar pacientes con diabetes
Pima.tr2.DM <-
  subset(Pima.tr2, Pima.tr2$type=="Yes")
sample(x=Pima.tr2.DM$ID, size = 20, replace = F)
[1] 120 298 218 73 83 79 203 87 184 242 167 269 18 161 251 84 76 243 102
[20] 96
```

#Objeto con muestra:

```
muestraDM <- Pima.tr2.DM |>
  sample_n(size=20, replace=F)
head(muestraDM)
```

	npreg	glu	bp	skin	bmi	ped	age	type	ID
1	6	102	82	NA	30.8	0.180	36	Yes	266
2	0	180	90	26	36.5	0.314	35	Yes	142
3	5	112	66	NA	37.8	0.261	41	Yes	225
4	1	128	48	45	40.5	0.613	24	Yes	193
5	2	134	70	NA	28.9	0.542	23	Yes	201
6	1	189	60	23	30.1	0.398	59	Yes	13

Referencias

Berndt, A. E. (2020). Sampling Methods. *Journal of human lactation: official journal of International Lactation Consultant Association*, 36(2), 224-226. <https://doi.org/10.1177/0890334420906850>

- Elfil, M. y Negida, A. (2017). Sampling methods in Clinical Research; an Educational Review. *Emergency (Tehran, Iran)*, 5(1), e52.
- Indrayan, A. y Malhotra, R. K. (2017). Sampling Methods. En *Medical Biostatistics* (4ª ed., pp. 37-49). Chapman & Hall/CRC Biostatistics Series.
- Kamper, S. J. (2020). Sampling: Linking Evidence to Practice. *The Journal of orthopaedic and sports physical therapy*, 50(12), 725-726. <https://doi.org/10.2519/jospt.2020.0704>
- Martínez-López, E., Pérez-Guerrero, E. E., Torres-Carrillo, N. M., López-Quintero, A., Betancourt-Núñez, A. y Gutiérrez-Hurtado, I. A. (2022). Methodological Aspects in Randomized Clinical Trials of Nutritional Interventions. *Nutrients*, 14(12), 2365. <https://doi.org/10.3390/nu14122365>
- Martínez-Mesa, J., González-Chica, D. A., Duquia, R. P., Bonamigo, R. R. y Bastos, J. L. (2016). Sampling: how to select participants in my research study? *Anais brasileiros de dermatologia*, 91(3), 326-330. <https://doi.org/10.1590/abd1806-4841.20165254>
- Purna Singh, A., Vadakedath, S. y Kandi, V. (2023). Clinical Research: A Review of Study Designs, Hypotheses, Errors, Sampling Types, Ethics, and Informed Consent. *Cureus*, 15(1), e33374. <https://doi.org/10.7759/cureus.33374>
- Setia, M. S. (2016). Methodology Series Module 5: Sampling Strategies. *Indian journal of dermatology*, 61(5), 505-509. <https://doi.org/10.4103/0019-5154.190118>
- Spolarich, A. E. (2023). Sampling Methods: A guide for researchers. *Journal of dental hygiene: JDH*, 97(4), 73-77.
- Upton, G. y Cook, I. (2014). *A Dictionary of Statistics* (3rd ed.). Oxford University Press. <https://www.oxfordreference.com/display/10.1093/acref/9780199679188.001.0001/acref-9780199679188>

Laboratorio de R 3

Edsaúl Emilio Pérez Guerrero

En este capítulo se revisarán a los operadores, unos símbolos que nos serán de mucha ayuda para realizar operaciones, asignaciones, pruebas lógicas, filtrado de bases de datos etc. Algunos de estos operadores (los numéricos) ya fueron revisados en el capítulo 1. En esta sección, solamente revisaremos los operadores relacionales, lógicos y de asignación.

Operadores relacionales

Muchas veces necesitamos hacer comparaciones de dos variables o de dos números. Los operadores relacionales en R, como en muchos otros lenguajes de programación, nos ayudan a comparar valores. Estos operadores permiten evaluar condiciones y tomar decisiones basadas en estas evaluaciones, lo que es fundamental en la programación y el análisis de datos. Como resultado siempre obtendremos una de dos respuestas TRUE o FALSE, cuando se cumpla o no respectivamente la condición de relación que establecimos.

La [tabla 6.1](#) muestra los operadores relacionales que necesitaremos para nuestros ejercicios.

TABLA 6.1. OPERADORES RELACIONALES EN R

OPERADORES RELACIONALES			
Operador	Comparación	Ejemplo	Resultado
<	Menor que	7 < 4	FALSE
<=	Menor o igual que	7 <= 4	FALSE
>	Mayor que	7 > 4	TRUE
>=	Mayor o igual que	7 >= 4	TRUE
==	Exactamente igual que	7 == 4	FALSE
!=	No es igual que	7 != 4	TRUE

Note cómo el símbolo = indica igualdad, en R un doble igual se utiliza para indicar igualdad. Es muy importante que recuerde esto, ya que un solo signo de = se utiliza para asignar un dato a un objeto.

Los operadores relacionales tienen la ventaja que nos permite comparar cualquier tipo de dato sin arrojar un error. Aunque comparar rojo con blanco no tiene mucho sentido, nos serán útiles para comparar datos y objetos, hacer subconjuntos de datos (por ejemplo, incluir todos los pacientes con una edad mayor a 50 años), comparar resultados de análisis estadísticos e inclusión en de comparaciones en funciones.

Operadores lógicos

Los operadores lógicos son operaciones que se utilizan para describir relaciones lógicas y siempre darán como resultado una respuesta dicotómica. En el caso de cumplirse la relación lógica que estamos evaluando, el resultado será TRUE; si esta, por el contrario, no se cumple, el resultado será FALSE. La [tabla 6.2](#) muestra los operadores lógicos utilizados en R.

TABLA 6.2. OPERADORES LÓGICOS EN R

OPERADOR	COMPARACIÓN
<code>x y</code>	x Ó y es verdadero
<code>x & y</code>	x Y y son verdaderos
<code>!x</code>	x no es verdadero (negación)
<code>isTRUE(x)</code>	x es verdadero (afirmación)

Debemos de realizar algunas aclaraciones sobre las operaciones lógicas. El símbolo `!` invierte el resultado lógico, es decir, convierte TRUE en FALSE y viceversa. Por ello, lo utilizamos para buscar un resultado diferente a. Aunque en este libro no utilizaremos las expresiones lógicas `&&` y `||` es necesario que conozca que existen y que se utilizan de forma común para comparar el primer elemento de cada vector (conjunto lineal de datos) y no todos los elementos.

Vamos a poner en práctica y a usar los operadores relacionales y lógicos. Para ello, vamos a utilizar la muestra de 12 pacientes de la base de datos *bacteria* [tabla 6.1](#).

TABLA 6.1. MUESTRA DE 12 DATOS DEL DATAFRAME BACTERIA

ID	Y	AP	HILO	WEEK	TRT	TIME	AGE
Patient A	n	a	hi	6	drug+	24	5
Patient B	n	a	hi	2	drug+	18	3
Patient C	n	a	hi	4	drug+	7	3
Patient D	n	a	hi	6	drug+	26	4
Patient E	n	p	hi	0	placebo	26	4
Patient F	y	p	hi	6	placebo	NA	3
Patient G	y	p	hi	11	placebo	NA	4
Patient H	y	a	lo	0	drug	NA	7
Patient I	y	a	lo	2	drug	NA	6
Patient J	y	a	lo	4	drug	NA	3
Patient K	y	a	lo	6	drug	NA	3
Patient L	n	a	lo	2	drug	20	4

Lo primero será crear tres objetos, uno para las edades de los pacientes A al D (drug+), otro para las edades de los pacientes E al G (placebo) y uno más para los pacientes H al L (drug). Para fines prácticos nombraremos estos objetos de la siguiente forma:

- ♦ Pacientes_A, para los pacientes A al D (drug+)
- ♦ Pacientes_E, para los pacientes E al G (placebo)
- ♦ Pacientes_H, para los pacientes H al L (drug)

El siguiente código muestra cómo crear los objetos:

```
Pacientes_A ← c(5,3,3,4)
Pacientes_E ← c(4,3,4)
Pacientes_H ← c(7,6,3,3,4)
# Recuerde que el símbolo ← es el símbolo de asignación
```

Supongamos que necesitamos saber cuantos pacientes de cada uno de los grupos tienen una edad mayor a 5, para ello podemos utilizar el siguiente código:

```
Pacientes_A>5
```

Esto dará como resultado:

```
[1] FALSE FALSE FALSE FALSE
```

Ninguno de los pacientes tiene una edad mayor a 5. Ahora pruebe con los otros dos objetos:

```
Pacientes_E>5
[1] FALSE FALSE FALSE
Pacientes_H>5
[1] TRUE TRUE FALSE FALSE FALSE
```

Con los operadores relacionales también podemos comparar dos objetos. Por ejemplo, ¿las edades de los pacientes del grupo A serán mayor que la de las edades del grupo E?

```
Pacientes_A>Pacientes_E
Warning in Pacientes_A > Pacientes_E: longer object length is not a
multiple of shorter object length
[1] TRUE FALSE FALSE FALSE
```

Lo que hizo R fue comparar cada por pares las edades del grupo A con las del grupo E. Preste atención a la advertencia que indica que uno de los objetos tiene una menor cantidad de datos.

Ejercicios

- ♦ **Ejercicio 6.1:** identifique cuántos de los pacientes del grupo H (Pacientes_H) tienen una edad menor a 3 o una edad mayor a 5.

Resolución ejercicio 6.1

Lo primero es crear los objetos:

```
Pacientes_A ← c(5,3,3,4)
Pacientes_E ← c(4,3,4)
Pacientes_H ← c(7,6,3,3,4)
```

El código anterior es igual al creado anteriormente, solo se muestra para fines prácticos.

Identifique cuántos de los pacientes del grupo H (Pacientes_H) tienen una edad menor a 3 o una edad mayor a 5.

```
Pacientes_H>5|Pacientes_H<3
[1] TRUE TRUE FALSE FALSE FALSE
```

Hay dos pacientes con una edad menor a 3 o una edad mayor a 5.

- ♦ **Ejercicio 6.2:** en los tres grupos de pacientes, ¿cuántos tienen edades entre 4 y 6 años?

Resolución ejercicio 6.2

En los tres grupos de pacientes, ¿cuántos tienen edades entre 4 y 6 años? Para facilitar la resolución de las siguientes preguntas se creará un objeto llamado `edades` que contendrá la unión de las edades de los tres grupos. Recuerde que puede unir y hacer operaciones con los objetos.

```
edades ← c(Pacientes_A, Pacientes_E, Pacientes_H)
```

Ahora ya podemos conocer cuántos pacientes tienen edades entre 4 y 6 años.

```
edades ≥ 4 & edades ≤ 6
```

```
[1] TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE  
FALSE TRUE
```

Hay 6 pacientes con edades entre los 4 y 6 años. Note cómo se utilizó el símbolo `&` para indicar dos condiciones. En estos casos es necesario volver a poner la condición completa y utilizar de nueva cuenta `edades`. Se puede utilizar la función `table` que permite contar cuántos falsos y cuántos negativos existen.

```
table(edades ≥ 4 & edades ≤ 6)
```

```
FALSE      TRUE  
6         6
```

- ♦ **Ejercicio 6.3:** ¿cuántos pacientes del grupo A tienen una edad mayor o igual a 4?

Resolución ejercicio 6.3

```
table(Pacientes_A ≥ 4)
```

```
FALSE      TRUE  
2         2
```

La función `table()` permite contar.

- ♦ **Ejercicio 6.4:** ¿cuántos pacientes de los tres grupos tienen una edad exactamente igual a 4?

Resolución ejercicio 6.4

```
table(edades==4 )
FALSE      TRUE
      8      4
```

- ♦ **Ejercicio 6.5:** ¿cuántos de los pacientes en los tres grupos tienen una edad diferente de 5?

Resolución ejercicio 6.5

```
table(edades!=5)
FALSE      TRUE
      1      11
```

- ♦ **Ejercicio 6.6:** ¿cuántos pacientes tienen una edad exactamente igual 3 o una edad mayor o igual a 6 años?

Resolución ejercicio 6.6

```
table(edades==3|edades>=6)
FALSE      TRUE
      5      7
```

- ♦ **Ejercicio 6.7:** ¿el promedio de las edades de los pacientes del grupo A (Pacientes_A) es mayor o igual al promedio de las edades del grupo H (Pacientes_H)?

Resolución ejercicio 6.7

```
mean(Pacientes_A)>mean(Pacientes_H)
[1] FALSE
```

Para la resolución de esta pregunta se utilizó la función `mean` que devuelve la media de un conjunto de datos.

Referencias

- Gandrud, C. (2013). *Learning R: A Step-by-Step Function Guide to Data Analysis*. O'Reilly Media, Inc.
- Long, J. y Teetor, P. (2019). *R Cookbook* (2nd ed.). <https://rc2e.com>
- Mailund, T. (2017). Introduction to R programming. En *Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist* (pp. 1-28).
- Moscarelli, M. (2023). Data types in 'R'. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 41-46). Springer Cham. <https://doi.org/10.1007/978-3-031-33073-5>
- R Core Team (2024). *R: A language and environment for statistical computing*.
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics*.
- Wickham, H. (2019). *Advanced R* (2nd ed.). Chapman & Hall/CRC The R Series.
- Wickham, H., Çetinkaya-Rundel, M. y Grolemund, G. (2023). Workflow: Code style. En *R for Data Science* (2nd ed.). O'Reilly Media, Inc. <https://r4ds.hadley.nz/workflow-style.html#sec-pipes>

Laboratorio de R 4

Edsaúl Emilio Pérez Guerrero

En este laboratorio de R se abordará el manejo de estructuras de datos. En R podemos encontrar distintas estructuras de datos, es decir, en R se pueden encontrar datos ordenados de distintas formas. Además, muchas veces es de nuestro interés acceder a datos en específico, manipular las estructuras o crear *subsets* con características peculiares. Por lo que es necesario comprender cómo estas estructuras se encuentran organizadas y cómo podemos hacer operaciones con ellas.

De acuerdo con su grado de complejidad, las estructuras de datos en R pueden ser: vectores, matrices y *Dataframe*. En R hay más estructuras de datos, sin embargo, en este libro solo nos enfocaremos en las tres mencionadas anteriormente.

Vectores

Tienen una dimensión y solo tienen un tipo de dato, por ejemplo, solamente datos numéricos.

El siguiente código crea un vector numérico:

```
Vector_Numerico <- c(1,2,3,4,5,6,7,8,9)
```

Si prestamos atención, nos daremos cuenta que el vector anterior solo tiene una dimensión y solo tiene valores numéricos:


```
Vector_Numerico
[1] 1 2 3 4 5 6 7 8 9
```

Otro ejemplo se muestra en el siguiente código, que es un vector con caracteres (cadenas):

```
Vector_Cadena ← c("Curso", "Proinnova", "tarea")
```

Si llamamos al objeto obtenemos lo siguiente:

```
Vector_Cadena
[1] "Curso" "Proinnova" "tarea"
```

Se pueden combinar los dos vectores que creamos anteriormente, pero R realizará una coerción de los datos. Puede ver más en [“Coerción de datos”](#).

```
Vector ← c(Vector_Numerico, Vector_Cadena)
```

Si comprobamos el tipo de dato que contiene el Vector veremos que lo transformó a cadena:

```
class(Vector)
[1] "character"
```

Matrices

Son estructuras de dos dimensiones que contienen solo un tipo de dato (numérico, cadena, etc.). Las dimensiones en una matriz se caracterizan por tener un largo (dimensión 1) y un alto (dimensión 2). Las matrices son, por lo tanto, una estructura con forma rectangular, con renglones y columnas en las que solamente hay un tipo de dato.

Dataframe

Los *Dataframes* o marco de datos es una estructura muy parecida a las matrices, sin embargo, a diferencia de las matrices pueden contener varios de tipos de datos en dos dimensiones. Un *Dataframe* es una estructura de datos que organiza los datos en una tabla bidimensional de filas y columnas, como una hoja de cálculo o una base de datos. En “[Definición de un Dataframe](#)” se abordó con un poco más de detalle la definición de este concepto. De momento recuerde que es la forma más intuitiva de almacenar y trabajar con datos ya que, por ejemplo, es muy fácil almacenar los datos en una hoja de cálculo y luego importarla a R como un *Dataframe*. En estas estructuras de datos los renglones contienen los casos, mientras que las columnas tienen las variables. Por ejemplo, en la base de datos `Pima.tr` (ver [tabla 7.1](#)), en los renglones o filas se encuentran cada una de las pacientes y en las columnas cada una de las variables (por ejemplo, `glu`, `bmi`).

TABLA 7.1. EJEMPLO DE UN DATAFRAME

NPREG	GLU	BP	SKIN	BMI	PED	AGE	TYPE
5	86	68	28	30.2	0.364	24	No
7	195	70	33	25.1	0.163	55	Yes
5	77	82	41	35.8	0.156	35	No
0	165	76	43	47.9	0.259	26	No
0	107	60	25	26.4	0.133	23	No
5	97	76	27	35.6	0.378	52	Yes
3	83	58	31	34.3	0.336	25	No
1	193	50	16	25.9	0.655	24	No
3	142	80	15	32.4	0.200	63	No
2	128	78	37	43.3	1.224	31	Yes
0	137	40	35	43.1	2.288	33	Yes
9	154	78	30	30.9	0.164	45	No

Los *Dataframe* son una de las estructuras de datos más comunes utilizadas en el análisis de datos, ciencias de datos y bioestadística.

Subconjuntos de datos

En este apartado se describirá cómo acceder a datos que tienen características específicas dentro de un *Dataframe*. Para ello, vamos a trabajar con la base de datos Vacunación que se describe a continuación:

Dataframe Vacunacion

El *Dataframe* Vacunacion es un archivo con extensión .xlsx que contiene un extracto de los datos acumulados de vacunación hasta el 31 de diciembre de 2021 por países y regiones para el COVID-19. Los datos fueron extraídos del repositorio [Our World in Data](#) en [GitHub para covid-19](#). El *Dataframe* cuenta con las siguientes variables:

- ♦ location: indica la región de la que fueron obtenidos los datos
- ♦ date: fecha
- ♦ vaccine: el tipo de vacuna
- ♦ total_vaccinations: el total acumulado para cada vacuna en específico

Utilice en el menú de Rstudio (ver apartado [[Importar base de datos desde el menú de Rstudio](#)]) para importar la base de datos. Recuerde que se trata de un archivo con extensión .xlsx. Puede descargar el *Dataframe* en [este enlace](#).

Nombre su base de datos como “Vacunacion”. Recuerde que este será nuestro mega objeto al que necesitamos acceder. Puede utilizar la función `View()` para visualizar su *Dataframe*, la cual se alimenta de objetos con estructura de *Dataframe*. El código `View(Vacunacion)` le permitirá después de importar en R poder echar un vistazo a su *Dataframe*. En la [tabla 7.2](#), se muestran los primeros 6 datos del *Dataframe* Vacunacion.

TABLA 7.2. VISTA GENERAL DEL DATAFRAME VACUNACION

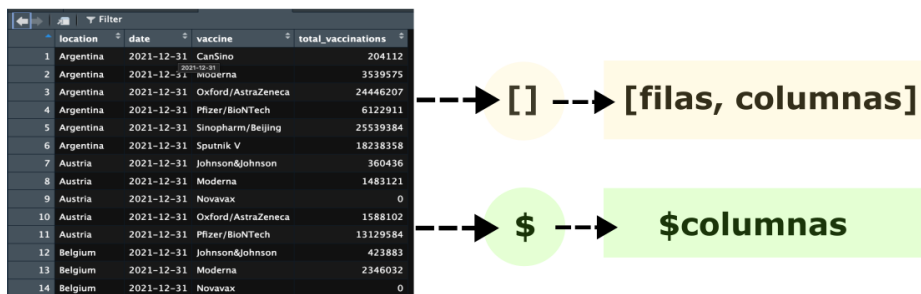
LOCATION	DATE	VACCINE	TOTAL_VACCINATIONS
Argentina	2021-12-31	CanSino	204112
Argentina	2021-12-31	Moderna	3539575
Argentina	2021-12-31	Oxford/AstraZeneca	24446207
Argentina	2021-12-31	Pfizer/BioNTech	6122911
Argentina	2021-12-31	Sinopharm/Beijing	25539384
Argentina	2021-12-31	Sputnik V	18238358
Austria	2021-12-31	Johnson&Johnson	360436
Austria	2021-12-31	Moderna	1483121
Austria	2021-12-31	Novavax	0
Austria	2021-12-31	Oxford/AstraZeneca	1588102
Austria	2021-12-31	Pfizer/BioNTech	13129584
Belgium	2021-12-31	Johnson&Johnson	423883

¿Cómo acceder a un subconjunto de datos?

Para acceder a los subconjuntos de datos o algún dato en específico podemos hacerlo utilizando los símbolos `[]` y el símbolo `$`. El símbolo `[]` nos permite acceder tanto a filas como a columnas y el símbolo `$` solo a columnas (ver [figura 7.1](#)).

FIGURA 7.1. ACCEDER A SUBCONJUNOS DE DATOS

Data frame



Subconjuntos de datos con []

A continuación, se explican algunos ejemplos utilizando corchetes [].

Supongamos que deseamos conocer cual es el dato que se ubica en la fila de 7 y en la columna 4. Para ello, podemos utilizar el siguiente código que devolverá precisamente este dato:

```
Vacunacion[7,4]
# A tibble: 1 × 1
  total_vaccinations
          <dbl>
1             360436
```

Este dato corresponde a la cantidad de individuos vacunados con la vacuna Johnson & Johnson en Austria. Note cómo el primer número indica la fila y el segundo la columna. El código `Vacunacion[7,4]` se podría leer `Vacunacion[ fila 7, columna 4]`.

Ahora suponga que desea conocer cuántas veces aparece cada uno de los países y regiones. Para ello, nos vamos a apoyar de la función `table` utilizando el siguiente código:

```
table(Vacunacion[,1])
location
Argentina   Austria   Belgium   Bulgaria   Chile
6           5         5         4           4
Croatia     Cyprus     Czechia   Denmark     Ecuador
4           4         3         4           4
Estonia     European Union Finland   France     Germany
4           9         4         4           5
Hong Kong   Hungary     Ireland   Italy       Latvia
2           6         3         3           3
Liechtenstein Lithuania Luxembourg Malta     Netherlands
3           4         4         4           5
Norway      Peru      Poland    Portugal   Romania
4           3         5         6           4
```

Slovakia	Slovenia	South Korea	Spain	Sweden
3	4	5	4	4
Switzerland	Ukraine	Uruguay		
3	5	3		

Note cómo dejamos el número de las filas vacío y pusimos el número 1 en las columnas. Este código indica que queremos los conteos de la primera columna, tomando en cuenta todas las filas. Si, por ejemplo, quisiéramos los conteos de las primeras 50 filas, el código sería el siguiente:

```
table(Vacunacion[1:50, 1])
```

location							
Argentina	Austria	Belgium	Bulgaria	Chile	Croatia	Cyprus	Czechia
6	5	5	4	4	4	4	3
Denmark	Ecuador		Estonia		Finland		
4	4		4		3		

Si no conoce el número de la fila o la columna puede utilizar su nombre, siempre y cuando se encuentre entrecomillado:

```
table(Vacunacion[1:50, "location"])
```

location							
Argentina	Austria	Belgium	Bulgaria	Chile	Croatia	Cyprus	Czechia
6	5	5	4	4	4	4	3
Denmark	Ecuador		Estonia		Finland		
4	4		4		3		

Subconjuntos de datos con \$

Otra forma de acceder a los subconjuntos de datos es utilizando el símbolo de \$. Por ejemplo, supongamos que deseamos conocer cuántas veces aparece cada uno de los países y regiones utilizando la función `table`. Para ello empleamos el siguiente código:

```
table(Vacunacion$location)
```

Argentina	Austria	Belgium	Bulgaria	Chile
6	5	5	4	4
Croatia	Cyprus	Czechia	Denmark	Ecuador
4	4	3	4	4
Estonia	European Union	Finland	France	Germany
4	9	4	4	5
Hong Kong	Hungary	Ireland	Italy	Latvia
2	6	3	3	3
Liechtenstein	Lithuania	Luxembourg	Malta	Netherlands
3	4	4	4	5
Norway	Peru	Poland	Portugal	Romania
4	3	5	6	4
Slovakia	Slovenia	South Korea	Spain	Sweden
3	4	5	4	4
Switzerland	Ukraine	Uruguay		
3	5	3		

Note cómo el símbolo \$ nos ayuda acceder a los objetos contenidos en nuestro megaobjeto (*Dataframe*). Hasta cierto punto esto nos puede ayudar a sustituir la función `attach()`.

Se pueden combinar el uso de `[]`, `$` y el uso de operadores relacionales, para acceder a datos en específico. Por ejemplo, suponga que deseamos filtrar la cantidad de individuos vacunados de Bélgica. Para ello empleamos el siguiente código:

```
Vacunacion$total_vaccinations[Vacunacion$location=="Belgium"]
[1] 423883 2346032 0 2846647 16078523
```

La explicación del código es la siguiente: Primero accedimos a la columna de total de vacunación. Posteriormente, dado que es de nuestro interés seleccionar solamente los datos de Bélgica, filtramos utilizando los `[]` para acceder a la columna `location` que a su vez filtramos utilizando los operadores lógicos para seleccionar los datos únicamente de aquello que en la columna `location` es exactamente igual a `Belgium`. Note cómo se utilizaron comillas y un signo de doble igual.

Ejercicios R lab 04

Utilizando la base de datos de vacunación responda los siguientes ejercicios:

- ♦ **Ejercicio 7.1:** identifique cuál es el dato que se encuentra en la fila 50 y columna 3.

Resolución ejercicio 7.1

Importe la base de datos Vacunacion y llámela Vacunacion. Si guardó su base de datos en su carpeta de bases el siguiente código debería de funcionar:

```
Vacunacion[50,3]
# A tibble: 1 × 1
  vaccine
  <chr>
1 Oxford/AstraZeneca
```

- ♦ **Ejercicio 7.2:** identifique cuál es el dato que se encuentra en la fila 98 y columna 2.

Resolución ejercicio 7.2

```
Vacunacion[98,2]
```

- ♦ **Ejercicio 7.3:** si quisiéramos acceder a toda la columna 3, ¿cuál sería el código que debería utilizarse? ¿Hay solo una opción?

Resolución ejercicio 7.3

```
table(Vacunacion[,3])
# Para facilitar la visualización se utilizó la función table
```

En este caso la función `table()` nos permite contar cuántos FALSE o cuántos TRUE hay.

- ♦ **Ejercicio 7.4:** suponga que solamente le interesan los datos contenidos en las filas 4 a la 10. ¿Qué código necesita para ello?

Resolución ejercicio 7.4

```
Vacunacion[4:10,]
```

- ♦ **Ejercicio 7.5:** del total de individuos vacunados acceda solamente a los datos de vacunación para Slovakia.

Resolución ejercicio 7.5

```
Vacunacion$total_vaccinations[Vacunacion$location="Slovakia"]
```

- ♦ **Ejercicio 7.6:** calcule el total de individuos vacunados en Slovakia. Puede utilizar la función `sum` o realizar la suma manualmente.

Resolución ejercicio 7.6

```
sum(Vacunacion$total_vaccinations[Vacunacion$location="Slovakia"])
```

La función `sum()` en R se utiliza para calcular la suma de los valores de un vector o una secuencia de números. Puede utilizar esta función para encontrar la suma de todos los elementos en un vector numérico.

- ♦ **Ejercicio 7.7:** acceda al total de los individuos vacunados en Lithuania y Germany.

Resolución ejercicio 7.7

```
Vacunacion$total_vaccinations[Vacunacion$location="Lithuania"|  
Vacunacion$location="Germany"]
```

- ♦ **Ejercicio 7.8:** cree un nuevo objeto con que contenga todas las columnas del *Dataframe* `vacunacion`, pero que solo contenga las filas de Lithuania. Llame a este objeto como `Lithuania`.

Resolución ejercicio 7.8

```
Lithuania <- Vacunacion[Vacunacion$location="Lithuania", ]  
Lithuania
```

Referencias

- Gandrud, C. (2013). *Learning R: A Step-by-Step Function Guide to Data Analysis*. O'Reilly Media, Inc.
- Long, J. y Teetor, P. (2019). *R Cookbook* (2nd ed.). <https://rc2e.com>
- Mailund, T. (2017). Introduction to R programming. En *Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist* (pp. 1-28).
- Moscarelli, M. (2023). Exploratory Data Analysis in “R”. En *Biostatistics With ‘R’: A Guide for Medical Doctors* (pp. 23-40). Springer Cham. <https://doi.org/10.1007/978-3-031-33073-5>
- R Core Team. (2024). *R: A language and environment for statistical computing*.

Laboratorio de R 5

Edsaúl Emilio Pérez Guerrero

Generar miles de líneas de código puede resultar muy tedioso e implica un arduo trabajo, por lo tanto, la eficiencia y claridad en el código son cruciales para el procesamiento de conjuntos de datos complejos y la realización de análisis estadísticos elaborados. La familia de funciones `apply` y los operadores `pipe`, `%>%` (introducido por el paquete `dplyr`) y `|>` (de R base), son herramientas fundamentales que ofrecen una sintaxis más limpia, intuitiva y fácil de leer. Además, permite ahorrarnos varias líneas de código.

Las funciones de la familia `apply` permiten la aplicación de diversas funciones a todo un *Dataframe*. Por otro lado, los operadores permiten generar código mucho más legible. Así pues, en este capítulo se revisarán las funciones de la familia `apply` y el operador `pipe` `%>%` y `|>`.

Funciones de la familia `apply()`

La familia de funciones `apply()` pertenecen al paquete base de R. Son funciones creadas para manipular segmentos de datos de matrices, arreglos, listas y marcos de datos de forma repetitiva. Por ejemplo, con estas funciones podemos repetir una función para todas las columnas de un *Dataframe*. También, podemos utilizar funciones para grupo de datos en específico, por ejemplo, comparar los casos de los controles. Otras funciones de esta familia realizan funciones de transformación o subconjunto.

La familia de funciones `apply` en R incluye `apply()`, `lapply()`, `sapply()`, `vapply()`, `mapply()`, `rapply()` y `tapply()`. Estas funciones están diseñadas para simplificar las operaciones repetitivas sobre estructuras de datos, como matrices y listas, sin la necesidad de recurrir a bucles `for` (que no se revisan en este libro). Estas funciones permiten aplicar una función a los elementos de un objeto de datos de manera eficiente, lo que resulta en código más compacto y legible.

En este texto nos enfocaremos únicamente en las funciones `apply()`, `lapply()` y `tapply()` de forma superficial. Sin embargo, puede encontrar un tutorial más avanzado en: <https://www.datacamp.com/tutorial/r-tutorial-apply-family>.

La función `apply()`

Esta función nos permite aplicar una misma función a todos los elementos de una matriz, ya sea columnas o filas.

La estructura general de la función `apply()` es la siguiente:

```
apply(X, MARGIN, FUN)
```

Si prestamos atención la función `apply()` necesita de tres argumentos para alimentarse. `X` es un arreglo de datos, el segundo parámetro, `MARGIN`, determina la dimensión (o margen) sobre la cual se agruparán los elementos de `X` para aplicarles una función específica. Estas dimensiones se identifican mediante números: el 1 para las filas y el 2 para las columnas. Por último, el tercer parámetro, `FUN`, especifica la función que se aplicará sobre la dimensión seleccionada. Para una mayor claridad sobre cómo opera esta función, es útil examinar un ejemplo práctico.

Supongamos que deseamos conocer el promedio de todas las variables del *Dataframe* `Pima.tr`, para ello podemos aplicar la función `mean()` empleando el siguiente código:

```
# Cargamos la base de datos
library(MASS)
data(Pima.tr)

# Aplicamos la función Apply a las columnas
apply(X=Pima.tr, MARGIN=2, FUN=mean)
```



```

apply(X=Pima.tr, MARGIN=2, FUN=mean, na.rm=T)
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA
Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
returning NA

```

npreg	glu	bp	skin	bmi	ped	age	type
NA	NA	NA	NA	NA	NA	NA	NA

Note cómo en el argumento `na.rm=T` se aplica la función `mean()` y no `apply()`. Todos los argumentos que se encuentren después del argumento `FUN` de `apply()` serán aplicados a la función que estemos utilizando dentro `apply()` y no a `apply()` en sí. Sin embargo, de nueva cuenta presentamos un error, por lo tanto, es necesario indicarle las columnas a las que queremos que se aplique la función, para ello vamos a utilizar los corchetes. No todas las variables dentro de `Pima.tr` son numéricas. Con el siguiente código, estamos aplicando una función a cada elemento de nuestro *Dataframe*. Los elementos son las columnas de la 1 a la 7. Con ello omitiremos la columna 8, que es un factor y que, por tanto, no se puede estimar la media.

```
apply(X=Pima.tr[,1:7], MARGIN=2, FUN=mean, na.rm=T)
```

npreg	glu	bp	skin	bmi	ped	age
3.570000	123.970000	71.260000	29.215000	32.310000	0.460765	32.110000

Note cómo el filtro se realizó para el argumento `X` y no para otros de los argumentos. En el [capítulo 7](#) se revisó a detalle cómo filtrar columnas y filas.

¿Y si quisiéramos la suma de los datos para cada una de las columnas de `Pima.tr`? El siguiente código nos ayudará:

```
apply(X=Pima.tr[,1:7], MARGIN=2, FUN=sum, na.rm=T)
```

npreg	glu	bp	skin	bmi	ped	age
714.000	24794.000	14252.000	5843.000	6462.000	92.153	6422.000

La función `lapply()`

Esta función es muy parecida a la función `apply()`, sin embargo, esta realiza una operación para una lista y devuelve una lista. De forma simple se puede entender a una lista, con un conjunto de datos de una sola dimensión. La función `lapply()` se puede aplicar a *Dataframe*, ya que R puede coercionar los *Dataframe* a lista en algunas situaciones.

El siguiente código ejemplifica cómo utilizar la función `lapply`

```
lapply(X=Pima.tr[,1:7], FUN=sd)
```

```
$npreg
[1] 3.366268
```

```
$glu
[1] 31.66723
```

```
$bp
[1] 11.4796
$skin
[1] 11.72459
```

```
$bmi
[1] 6.130212
$ped
[1] 0.3072248

$age
[1] 10.97544
```

Si notamos en el código no es necesario utilizar el argumento `MARGIN`, ya que las listas solo tienen una dimensión. Obtenemos el mismo resultado que el que se obtuvo con la función `apply` pero en forma de lista.

La función `tapply`

Esta función nos permite obtener resultados agrupados. La estructura básica de las funciones es la siguiente:

```
tapply(X, INDEX, FUN)
```

Donde `X` es la variable a la que se aplicará la función. `INDEX` es la variable de agrupación y `FUN` es la función que se aplicará a los datos agrupados. Por ejemplo, supongamos que deseamos conocer la media de edad para las pacientes con diabetes y sin diabetes, para ello empleamos el siguiente código:

```
tapply(X=Pima.tr$age, INDEX=Pima.tr$type, FUN=mean)
      No      Yes
29.23485 37.69118
```

Note cómo fue necesario utilizar el símbolo `$`, de lo contrario R no sabría dónde buscar el objeto para hacer las operaciones.

Operadores pipe

En R podemos encontrar varios operados conocidos como pipe. Los más utilizados es el símbolo `%>%` que se encuentra en el paquete `dplyr` y el símbolo `|>` que se encuentra en el paquete base de R. Un pipe puede definirse como un símbolo que permite realizar llamadas o funciones encadenadas. De una manera más simple se puede entender como un símbolo que le permite pasar un resultado intermedio a la siguiente función y el resultado de esta función a la siguiente.

Comenzaremos explicando el operador `%>%` que es un poco más sencillo de entender que el operador `|>`.

Operador `%>%`

La estructura básica del operador `%>%` es la siguiente:

```
df %>%
  operación_1 %>%
  operación_2 %>%
  operación_3
```

El código anterior permite seleccionar una variable de un *Dataframe* (`df`), realizar una operación (operación 1) y tomar los resultados de esta operación para realizar la operación 2. Después toma los resultados de la operación 2 para realizar la última operación (operación 3).

El operador pipe simplemente alimenta los resultados de una operación a la siguiente operación debajo de ella. La ventaja de usar el operador de tubería es que hace que el código sea extremadamente fácil de leer.

El siguiente código permite estimar la media de la edad de los pacientes incluidos en la base `melanoma` y redondear el resultado a dos dígitos:

```
# Llamar la librería MASS para poder cargar la base melanoma
library(MASS)
data("Melanoma")
```

```
# Estimar la media redondeando a dos dígitos
round(mean(Melanoma$age), 2)

[1] 52.46
```

Note cómo necesitamos de la función `round` y dentro de ella estimamos la media. Utilizando el operador pipe el código podría simplificarse de la siguiente forma:

```
library(dplyr) #Librería para poder utilizar el operador pipe

Attaching package: 'dplyr'
The following object is masked from 'package:MASS':
  select
The following objects are masked from 'package:stats':
  filter, lag
The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

Melanoma$age %>%
  mean() %>%
  round(2)

[1] 52.46
```

El código anterior tomó el objeto `Melanoma$age` para estimar la media y luego redondearlo a 2 dígitos.

Existen algunas funciones en las que no es necesario utilizar el símbolo `$` para que R sepa dónde buscar la variable sobre la cual va ejecutar la operación. Estas funciones pertenecen principalmente al mundo de `tidyverse` y a la librería `rstatix` que utilizaremos en futuros capítulos.

Operador `|>`

Se podría decir que los operadores son iguales y hasta cierto punto funcionan de la misma manera. Sin embargo, al tener orígenes distintos, en ciertas situaciones difieren en su funcionamiento. Profundizar en las diferencias de

estos operados se escapa de los objetivos de este texto, pero para fines prácticos podríamos decir que el operador `|>` siempre necesita tener declarado la base de datos de donde tomar las variables. Por ejemplo, si quisiéramos estimar una regresión (ver “[Regresión lineal múltiple](#)”) utilizando el operador `%>%` podríamos utilizar el siguiente código:

```
Melanoma %>%
  lm(age~thickness, data=.)

Call:
lm(formula = age ~ thickness, data = .)

Coefficients:
(Intercept)  thickness
  48.968    1.197
```

Sin embargo, con el operador `|>` necesitamos declarar la base de datos y su uso no se justificaría, ya que el resultado sería un error.

```
Melanoma |>
  lm(age~thickness, data=.)# No declarar la base de datos
```

El código anterior daría como resultado: `Error in is.data.frame(data) : object ‘.’ not found.`

Si volvemos al ejemplo de la estimación de la media con dos dígitos, no tendríamos problema en usar los pipe de forma indistinta y el código sería prácticamente igual.

```
Melanoma$age |>
  mean() |>
  round(2)

[1] 52.46
```

El operador `%>%` también nos puede ayudar a filtrar bases de datos utilizando la función `subset()`. Esta función se utiliza para extraer subconjuntos de datos de un *Dataframe* u otros objetos similares (como matrices o listas), según condiciones específicas. Esta función es muy útil para filtrar filas basadas en criterios lógicos y también permite seleccionar columnas específicas. Vamos a ver un ejemplo.

- ♦ **Ejemplo 8.1:** suponga que desea estimar la media de edad para las pacientes con más de dos partos. Este resultado deberá estar redondeado a 4 cifras.

```
Pima.tr %>%
  subset(npreg>2) %>%
  .$age %>%
  mean(na.rm = TRUE) %>%
  round(4)
[1] 37.4124
```

Explicación del código:

- ♦ `Pima.tr %>%`: inicia el pipe con la base de datos `Pima.tr`.
- ♦ `subset(npreg > 2) %>%`: filtra las filas donde el número de embarazos (`npreg`) es mayor a dos.
- ♦ `.$age %>%`: extrae la columna de edad (`age`) de las filas filtradas. De lo contrario R pensará que la función `mean()` se debe alimentar con un *Dataframe*, lo cual es incorrecto. El operador `.$` es una forma de acceder a una columna específica de un *Dataframe* dentro de un flujo de pipe. Aquí se está utilizando para extraer una columna específica (`age`) del *Dataframe* resultante.
- ♦ `mean(na.rm = TRUE) %>%`: calcula la media de las edades, omitiendo los valores NA.
- ♦ `round(4)`: redondea el resultado a 4 cifras.

Conclusión sobre los dos pipe

Aunque es más complejo, podríamos decir que ambos operadores pueden funcionar para lo mismo. En este texto utilizaremos ambos operadores, aunque la mayoría de las ocasiones utilizaremos ambos operadores, pero le daremos prioridad al operador `|>`.

Información extra

Puede encontrar mayor información sobre el uso de los pipe en:

- ♦ <https://www.statology.org/pipe-in-r/>
- ♦ <https://www.r-bloggers.com/2017/12/pipes-in-r-tutorial-for-beginners/>
- ♦ <https://towardsdatascience.com/an-introduction-to-the-pipe-in-r-823090760d64>
- ♦ <https://towardsdatascience.com/understanding-the-native-r-pipe-98dea6d8b61b>
- ♦ <https://towardsdatascience.com/understanding-the-native-r-pipe-98dea6d8b61b>

Ejercicios

Ejercicios de utilizando las funciones de la familia apply

- ♦ **Ejercicio 8.1:** utilizando la base de datos `Pima.tr` de la librería `MASS` estime la media, la mediana, la desviación estándar y el rango intercuartil para cada una de las variables del *Dataframe*, agrupando a las pacientes con y sin diabetes. Para ello, utilice las siguientes funciones:
 - » `mean()` para la media
 - » `median()` para la mediana
 - » `sd()` para la desviación estándar
 - » `IQR()` para el rango intercuartil

Este tipo de estimaciones serán revisadas más adelante en el libro, aquí solo buscamos que comprenda el funcionamiento de las funciones.

Resolución [ejercicio 8.1](#)

```
library(MASS)
data(Pima.tr)
apply(Pima.tr[,1:7], 2, tapply, Pima.tr$type, mean)
apply(Pima.tr[,1:7], 2, tapply, Pima.tr$type, median)
apply(Pima.tr[,1:7], 2, tapply, Pima.tr$type, sd)
```

- ♦ **Ejercicio 8.2:** utilizando la base de datos Melanoma de la librería MASS estime la media, la mediana, la desviación estándar y el rango intercuartil para cada una de las variables del *Dataframe* que sean cuantitativas.

Resolución ejercicio 8.2

No tiene sentido estimar datos descriptivos del año de inclusión del paciente.

```
library(MASS)
data("Melanoma")
apply(Melanoma[, c(1,4,6)], MARGIN = 2, FUN = mean)
apply(Melanoma[, c(1,4,6)], MARGIN = 2, FUN = median)
apply(Melanoma[, c(1,4,6)], MARGIN = 2, FUN = sd)
apply(Melanoma[, c(1,4,6)], MARGIN = 2, FUN = IQR)
```

- ♦ **Ejercicio 8.3:** utilizando la base de datos Melanoma de la librería MASS estime la media, la mediana, la desviación estándar y el rango intercuartil para cada una de las variables del *Dataframe* agrupando por la variable status.

Resolución ejercicio 8.3

```
apply(Melanoma[, c(1,4,6)], 2, tapply, Melanoma$status, mean)
apply(Melanoma[, c(1,4,6)], 2, tapply, Melanoma$status, median)
apply(Melanoma[, c(1,4,6)], 2, tapply, Melanoma$status, sd)
apply(Melanoma[, c(1,4,6)], 2, tapply, Melanoma$status, IQR)
```

Ejercicios para utilizar pipes

- ♦ **Ejercicio 8.4:** utilizando el pipe `%>%` de R estime la media y la mediana para todas variable numéricas de `Pima.tr`, pero solo incluya a las pacientes con un `gluosa > 120`. Utilice la función `subset()`. Le será de ayuda.

Resolución [ejercicio 8.4](#)

```
Pima.tr %>%
  subset(glu>120) %>%
  .$age %>%
  mean(na.rm = TRUE)
```

Una explicación puntal del código anterior se dio en el [ejemplo 8.1](#).

```
Pima.tr %>%
  subset(glu>120) %>%
  .$age %>%
  median(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$npreg %>%
  mean(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$npreg %>%
  median(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$bp %>%
  mean(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$bp %>%
  median(na.rm = TRUE)
```

```
Pima.tr %>%
  subset(glu>120) %>%
  .$skin %>%
  mean(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$skin %>%
  median(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$bmi %>%
  mean(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$bmi %>%
  median(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$ped %>%
  mean(na.rm = TRUE)

Pima.tr %>%
  subset(glu>120) %>%
  .$ped %>%
  median(na.rm = TRUE)
```

Referencias

- Gandrud, C. (2013). *Learning R: A Step-by-Step Function Guide to Data Analysis*. O'Reilly Media, Inc.
- Long, J. y Teetor, P. (2019). *R Cookbook* (2nd ed.). <https://rc2e.com>
- Mailund, T. (2017). Introduction to R programming. En *Beginning Data Science in R: Data Analysis, Visualization, and Modelling for the Data Scientist* (pp. 1-28).

- Moscarelli, M. (2023). Data types in 'R'. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 41-46). Springer Cham. <https://doi.org/10.1007/978-3-031-33073-5>
- R Core Team (2024). *R: A language and environment for statistical computing*.
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics*.
- Wickham, H. (2019). *Advanced R* (2nd ed.). Chapman & Hall/CRC The R Series.
- Wickham, H., Çetinkaya-Rundel, M. y Grolemund, G. (2023). Workflow: Code style. En *R for Data Science* (2nd ed.). O'Reilly Media, Inc. <https://r4ds.hadley.nz/workflow-style.html#sec-pipes>

Laboratorio 6 de R

Edsaúl Emilio Pérez Guerrero

En el apartado “Tipos de datos en R” presentamos los factores. Establecimos que son tipos de datos que contienen niveles y que estos siguen una jerarquía alfanumérica. Sin embargo, ¿qué pasaría si quisiéramos modificar el orden de estos niveles? ¿O qué pasaría si quisiéramos transformar una variable numérica a un factor y modificar las etiquetas de los niveles? Este laboratorio 6 de R se enfoca precisamente en ello, en la modificación de factores utilizando la función `factor()`.

La función `factor()`

La función `factor()` permite:

- ♦ Crear factores
- ♦ Especificar niveles
- ♦ Reordenar niveles
- ♦ Cambiar etiquetas
- ♦ Especificar etiquetas

Esta función se encuentra en el paquete base de R, por lo que no es necesario instalar ninguna otra librería. Aunque existe la librería `forcats` que se especializa en la modificación de factores, en este libro, salvo algunas excepciones, siempre utilizaremos la función `factor()` para trabajar con factores.

No olvide consultar la ayuda de la función con `help("factor")`. Vamos a ejemplificar cómo utilizar la función `factor()`.

Creación de factores utilizando la función `factor()`

La función `factor()` convierte un vector de datos en un factor.

Supongamos que tenemos un vector de caracteres con el nivel escolar de un grupo de personas que queremos convertir a factor. Para ello podemos seguir los siguientes pasos:

- ♦ Paso 1: crear el vector de datos

Supongamos que tienes un vector de caracteres con el nivel escolar de un grupo de personas. Los niveles incluidos son “Primaria”, “Secundaria” y “Universidad”:

```
nivel_escolar ← c("Universidad", "Primaria",
                  "Secundaria", "Universidad",
                  "Primaria", "Secundaria", "Primaria")
```

- ♦ Paso 2: convertir a factor

Ahora, vamos a convertir este vector en un factor. Además, queremos asegurarnos de que los niveles del factor estén en un orden lógico, desde la educación básica hasta la superior:

```
factor_nivel_escolar ← factor(nivel_escolar,
                              levels = c("Primaria", "Secundaria",
                              "Universidad"))
```

- ♦ Paso 3: verificar el factor

Para verificar que el vector ha sido convertido correctamente en un factor con los niveles ordenados adecuadamente, podemos imprimir y verificar la “estructura” del factor:

```
print(factor_nivel_escolar) # Para imprimir el factor
```

```
[1] Universidad Primaria Secundaria Universidad Primaria Secundaria
[7] Primaria
Levels: Primaria Secundaria Universidad
str(factor_nivel_escolar) # Para verificar la estructura
Factor w/ 3 levels "Primaria","Secundaria",..: 3 1 2 3 1 2 1
```

Note cómo el resultado de la función `str()` nos dice entre otras cosas cuáles son y cómo están ordenados los niveles de los factores.

No olvide que siempre puede utilizar la función `as.factor()` para coercionar una variable a un factor.

Especificar y modificar etiquetas de los niveles de un factor con la función `factor()`

Vamos a retomar el objeto creado en la sección anterior para modificar las etiquetas de los niveles.

- ♦ **Ejemplo 9.1:** supongamos que queremos ser más descriptivos o cambiar las etiquetas de los nombres de los niveles. Para ello utilizamos el siguiente código:

```
levels(factor_nivel_escolar) ← c("Educación Primaria",
                                "Educación Secundaria",
                                "Educación Universitaria")

print(factor_nivel_escolar)
[1] Educación Universitaria Educación Primaria Educación Secundaria
[4] Educación Universitaria Educación Primaria Educación Secundaria
[7] Educación Primaria
Levels: Educación Primaria Educación Secundaria Educación Universitaria
```

Ahora bien, también podemos especificar los niveles y convertir una variable numérica a factor.

- ♦ **Ejemplo 9.2:** suponga que quiere convertir a factor la variable `low` del *Dataframe* `birthwt` de la librería `MASS`. Además, es de su interés especificar los niveles de la siguiente manera: “Peso normal” para el 0, y “Bajo peso” para el 1.

» Paso 1: importar base de datos

```
library(MASS) # Cargar librería que incluye el Dataframe
data("birthwt") #Importar Dataframe
```

» Paso 2: convertir la variable a factor

```
birthwt$low <- factor(birthwt$low,
                      levels = c(0, 1),
                      labels = c("Peso normal",
                                "Bajo peso"))
```

En el código anterior, `levels = c(0, 1)` especifica el orden de los niveles en los datos originales, y `labels = c("Peso normal", "Bajo peso")` asigna una etiqueta a cada uno de esos niveles.

La parte del código merece una explicación adicional:

- ♦ `‘birthwt$low’` antes del signo de asignación (`<-`): selecciona la columna `low` del *Dataframe* `birthwt`.
- ♦ Mientras que:

```
<- factor(birthwt$low,
          levels = c(0, 1),
          labels = c("Peso normal",
                    "Bajo peso"))
```

Reemplaza la columna `low` original por otra variable también llamada `low`, pero que se convirtió en factor con sus respectivas etiquetas asociadas. Es decir, estamos reemplazando nuestra variable original con otra llamada igual, pero con características diferentes. El usar `$` y llamar al objeto `low` desde el *Dataframe* permite que el reemplazo se haga en la base de datos directamente. Esta técnica la vamos a utilizar mucho en este libro.

» Paso 3: verificar la variable

```
str(birthwt$low)
Factor w/ 2 levels "Peso normal", ..: 1 1 1 1 1 1 1 1 1 1 ...
table(birthwt$low) # Permite mediante conteos conocer los niveles
```

Peso normal	Bajo peso
130	5
	9

Reordenar niveles de un factor con la función factor()

Vamos a retomar el objeto `factor_nivel_escolar` que creamos anteriormente para el siguiente ejemplo.

- ♦ **Ejemplo 9.3:** suponga que para un análisis específico requiere invertir los niveles de su objeto `factor_nivel_escolar` para que el nivel más bajo sea “Educación Universitaria” y el nivel más alto “Educación Primaria”. Para ello, vamos a utilizar la función `factor()` como se muestra a continuación:

```
# Imprimir el factor original para visualizar los datos
print("Factor original:")
[1] "Factor original:"
print(factor_nivel_escolar)
[1] Educación Universitaria Educación Primaria Educación Secundaria
[4] Educación Universitaria Educación Primaria Educación Secundaria
[7] Educación Primaria
Levels: Educación Primaria Educación Secundaria Educación Universitaria
# Reordenar los niveles del factor
# Creando un nuevo factor
factor_nivel_escolar2 <- factor(factor_nivel_escolar,
                                levels = c("Educación Universidad", "Educación Primaria",
                                             "Educación Secundaria"))
```

```
# Verificar el nuevo orden de los niveles
print("Factor con niveles reordenados:")
[1] "Factor con niveles reordenados:"
print(factor_nivel_escolar2)
[1] <NA>      Educación Primaria      Educación Secundaria
[4] <NA>      Educación Primaria      Educación Secundaria
[7] Educación Primaria
Levels: Educación Universidad Educación Primaria Educación Secundaria
# Estructura del factor
str(factor_nivel_escolar2)
Factor w/ 3 levels "Educación Universidad", ..: NA 2 3 NA 2 3 2
# Tabla de frecuencias
table(factor_nivel_escolar2)
factor_nivel_escolar2
Educación Universidad      Educación Primaria      Educación Secundaria
0                          3                          2
```

Vamos a explicar el código anterior:

- ♦ El primer `print()` muestra un mensaje que indica que lo que sigue es el factor original. El segundo `print()` muestra los niveles y los datos del factor original.
- ♦ El factor original `factor_nivel_escolar` se pasa nuevamente a través de la función `factor()`, pero esta vez con un nuevo orden de niveles especificado por el argumento `levels`. Se establece que “Universidad” sea el primer nivel, seguido por “Primaria” y “Secundaria”. Esto reordena los niveles.
- ♦ Finalmente se verifica el objeto.
También se pueden verificar los niveles del objeto con la función `levels()`:

```
levels(factor_nivel_escolar)
[1] "Educación Primaria"      "Educación Secundaria"
[3] "Educación Universitaria"
```

Ejercicios

- ♦ **Ejercicio 9.1:** convertir un vector de datos en un factor.
 Datos: `c("rojo", "azul", "verde", "azul", "rojo")`. Convierta este vector en un factor, utilice `print()` para mostrar el resultado y compruebe que es un factor.

Resolución ejercicio 9.1

```
colores <- c("rojo", "azul", "verde", "azul", "rojo")
colores_factor <- as.factor(colores)
print(colores_factor)
is.factor(colores_factor)
```

- ♦ **Ejercicio 9.2:** crear un factor y especificar el orden de los niveles manualmente.
 Datos: `c("bajo", "alto", "medio", "bajo", "alto")`. Convierta este vector en un factor, especifica el orden de los niveles como "alto", "medio", "bajo", y utiliza `print()` para mostrar el resultado.

Resolución ejercicio 9.2

```
datos <- c("bajo", "alto", "medio", "bajo", "alto")
factor_datos <- factor(datos, levels = c("alto", "medio", "bajo"))
print(factor_datos)
```

- ♦ **Ejercicio 9.3:** cambiar el orden de los niveles de un factor existente.
 Datos: factor con niveles `c("frío", "caliente", "templado")` en ese orden. Reordene los niveles para que "templado" sea el primer nivel y muestra el factor reordenado.

Resolución ejercicio 9.3

```
temperaturas <- factor(c("frío", "caliente", "templado", "frío", "caliente"), levels = c("frio", "caliente", "templado"))
temperaturas <- factor(temperaturas, levels = c("templado", "frío", "caliente"))
print(temperaturas)
```


- ♦ **Ejercicio 9.4:** cambiar las etiquetas de los niveles de un factor.
Datos: factor con niveles c("sí", "no"). Cambie las etiquetas a "Afirmativo" y "Negativo" y muestre el resultado.

Resolución ejercicio 9.4

```
respuestas <- factor(c("sí", "no", "sí", "no", "sí"), levels = c("sí",
"no"))
levels(respuestas) <- c("Afirmativo", "Negativo")
print(respuestas)
```

- ♦ **Ejercicio 9.5:** integrar un factor en un *Dataframe* y realizar un conteo utilizando la función `table()`.
Utilice la base de datos `Pima.tr2` de la librería `MASS` para crear una variable que introduzca datos aleatorios sobre si las pacientes tienen o no hipertensión. Esta variable debe de ser un factor. Utilice la función `table()` para contar y asigne una semilla de 1234.

```
library(MASS)
data("Pima.tr2")
set.seed(1234)
hipertension <- sample(x=c("Sí", "No"),
                      size = 300, replace = T)
Pima.tr2$hipertension <- as.factor(hipertension)

table(hipertension)
```

Resolución ejercicio 9.5

```
library(MASS)
data("Pima.tr2")
set.seed(1234)
hipertension <- sample(x=c("Sí", "No"), size = 300, replace = T)
Pima.tr2$hipertension <- as.factor(hipertension)
table(hipertension)
```

- ♦ **Ejercicio 9.6:** utilice el *Dataframe* `birthwt` de la librería `MASS` para convertir a objeto la variable `smoke`. Sustituya el 0 por “No fumó en el embarazo” y el 1 por “Sí fumó en el embarazo”. No deberá crear una nueva variable sino sustituir la original.

```
library(MASS)
data("birthwt")

print("Factor original:")
print(table(birthwt$smoke))

birthwt$smoke <- factor(birthwt$smoke, levels = c(0,1),
                        labels = c("No fumó en el embarazo", "Sí fumó en el embarazo"))

print("Factor modificado:")
print(table(birthwt$smoke))
levels(birthwt$smoke)
```

Resolución ejercicio 9.6

```
library(MASS)
data("birthwt")

print("Factor original:")
print(table(birthwt$smoke))

birthwt$smoke <- factor(birthwt$smoke, levels = c(0,1),
                        labels = c("No fumó en el embarazo", "Si fumó en el embarazo"))

print("Factor modificado:")
print(table(birthwt$smoke))
levels(birthwt$smoke)
```

- ♦ **Ejercicio 9.7:** resuelva el ejercicio anterior, pero utilizando la librería `forcats`.

Resolución [ejercicio 9.7](#)

```
install.packages("forcats") # En el caso que lo requiera
library(MASS)
data("birthwt")
birthwt$smoke ← forcats::fct_recode(as.factor(birthwt$smoke),
  "No fumó en el embarazo"="0", "Sí fumó en el embarazo"="1")
levels(birthwt$smoke)
```

Visualización de datos

Miryam Rosario Guillén Medina

Edsaúl Emilio Pérez Guerrero

En “[Etapas en el análisis estadístico](#)” se describieron las etapas del análisis estadístico. Una de estas etapas es la exploración de datos que a su vez se compone de la visualización de datos y de la estadística descriptiva.

La visualización de datos es un aspecto fundamental en el análisis estadístico, ya que permite la comprensión rápida de la información, facilita el análisis y la comunicación de resultados. Además, es más fácil entender los datos gráficamente que entenderlos mediante tablas. En este capítulo se revisarán los aspectos básicos para la construcción de gráficas en R, para ello utilizaremos el paquete base que viene incluido en R, pero también utilizaremos la librería `ggplot` que es el estándar actual para la construcción de gráficos en R.

Los objetivos de aprendizaje del capítulo son:

- ♦ Realizar gráficos mediante el paquete Base de R
- ♦ Comprender y utilizar los argumentos básicos de las funciones que permiten crear gráficos en R
- ♦ Comprender la sintaxis utilizada en `ggplot`
- ♦ Construir gráficos básicos mediante `ggplot`

En este capítulo se utilizan muchos términos que son definidos y utilizados en capítulos posteriores. No se preocupe si no comprende estos términos, más adelante seguro lo hará. Este capítulo solo busca que el lector pueda crear gráficos, los cuales utilizaremos más adelante en este libro. Cuando necesite construir una gráfica, siempre puede regresar a este capítulo.

Siempre recuerde: en R las gráficas son gratis.

Visualización utilizando el paquete base

Para la creación de gráficas necesitaremos importar algunas bases de datos que ya hemos utilizado anteriormente:

- ♦ Pima.tr (descrita en el [Ejercicio 3.5](#))
- ♦ birthwt (descrita en el [Ejercicio 3.4](#))
- ♦ Vacunación (descrita en “[Dataframe Vacunacion](#)”)

Histogramas

Uno de los gráficos más utilizados para visualizar variables cuantitativas son los histogramas. Hasta cierto punto, un histograma es similar a un gráfico de barras después de que los valores de las variables se agrupan en un número finito de intervalos de clase. En los histogramas para cada intervalo la altura de la barra corresponde a la frecuencia de observaciones en ese intervalo. Los histogramas permiten tener la siguiente información:

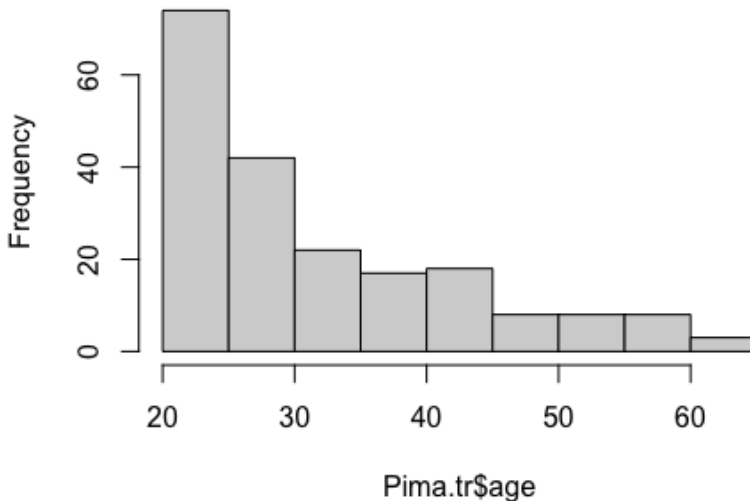
- ♦ Distribución de los datos (ver [capítulo 12](#))
- ♦ Dispersión de la variable cuantitativa
- ♦ Ayudan a identificar donde están concentrados los datos
- ♦ Moda. Nos pueden permitir identificar cual es el dato que más se repite
- ♦ Valores atípicos. Aunque un histograma no muestra datos atípicos con tanta claridad como un [boxplot](#), se puede inferir su presencia si hay barras que están aisladas del resto de la distribución.

Los histogramas se crean con la función `hist()`. Vamos a comenzar a construir gráficos, del más sencillo a los más complejos. Es importante que identifique los argumentos que se van añadiendo, ya que estos son comunes para los gráficos construidos desde el paquete base.

- ◊ Ejemplo 10.1: vamos a construir un histograma básico. Para ello, vamos a utilizar la variable `age` del *Dataframe* `Pima.tr` de la librería `MASS`.

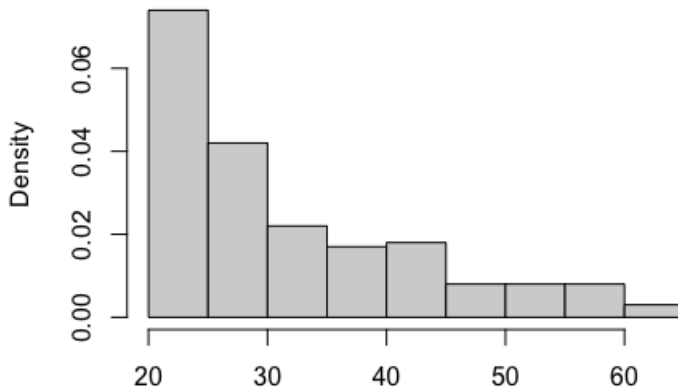
```
# Importar base de datos
library(MASS)
data("Pima.tr")
# Construcción del histograma
hist(Pima.tr$age)
```

Histogram of Pima.tr\$age



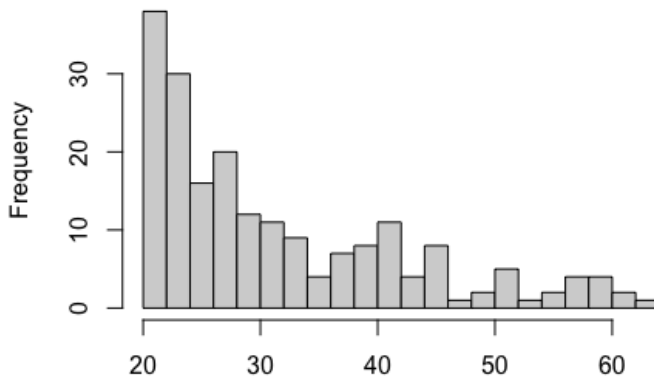
Una vez construido nuestro histograma, vamos a comenzar a modificarlo. Con el argumento `freq = FALSE` podemos obtener las densidades:

```
hist(Pima.tr$age, freq = F)
```

Histogram of Pima.tr\$age

El argumento `breaks` nos permite cambiar los intervalos de clase por el número que establezcamos:

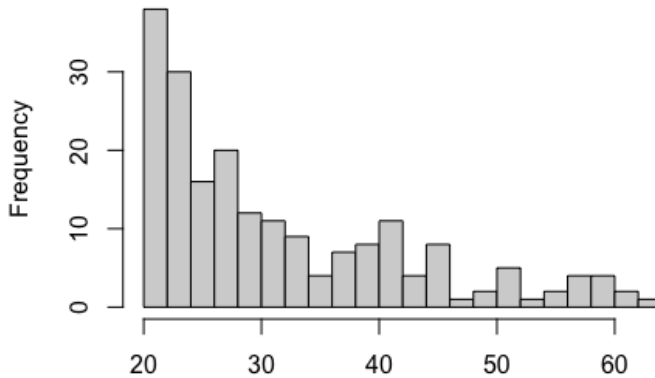
```
hist(Pima.tr$age, breaks = 20) #Histograma con 20 intervalos de clase
```

Histogram of Pima.tr\$age

Podemos cambiar el título, para ello utilizamos el argumento `main` y entre comillas podemos declarar el título que nos convenga:

```
hist(Pima.tr$age, breaks = 20, main = "Mi primer histograma")
```

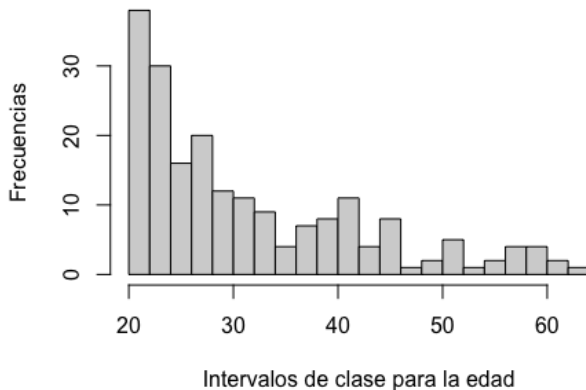
Mi primer histograma



También, con los argumentos `ylab` y `xlab` podemos cambiar los nombres de los ejes “y” y “x” respectivamente:

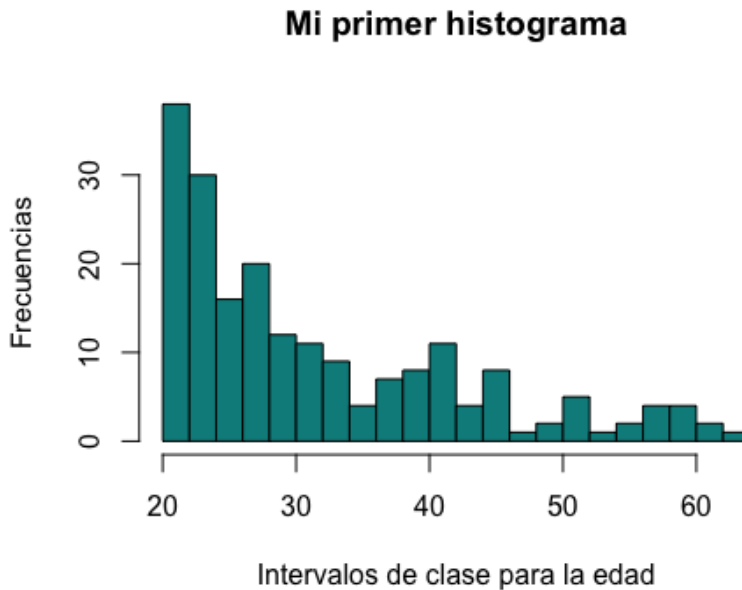
```
hist(Pima.tr$age, breaks = 20, main = "Mi primer histograma",  
     ylab = "Frecuencias", xlab= "Intervalos de clase para la edad")
```

Mi primer histograma



También puede cambiar el color de las barras, por ejemplo, podemos cambiarla por el color “cyan4” utilizando el argumento `col`:

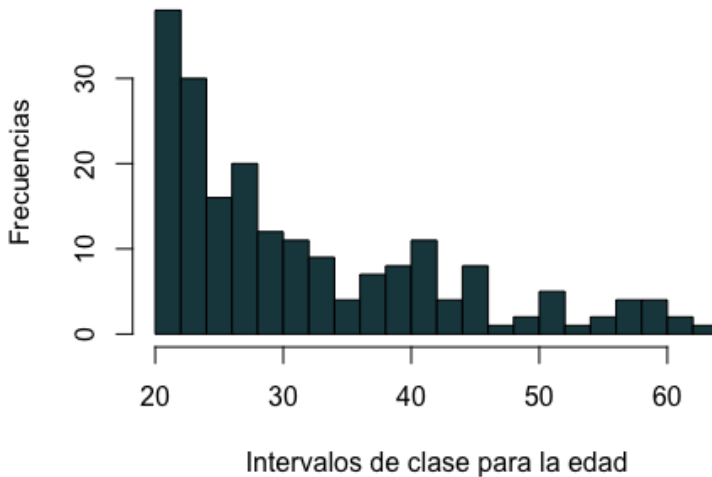
```
hist(Pima.tr$age, breaks = 20, main = "Mi primer histograma",
     ylab = "Frecuencias", xlab= "Intervalos de clase para la edad",
     col = "cyan4")
```



Note como cuando R reconoce el color, el texto cambia a ese color. En R los colores pueden ser leídos mediante nombres (como en el código anterior), mediante números o mediante el código de colores de RGB. Por ejemplo, con el siguiente código vamos a modificar el color “cyan4” y vamos a utilizar el color “#1C464D” que corresponde al color *Midnight green*:

```
hist(Pima.tr$age, breaks = 20, main = "Mi primer histograma",
     ylab = "Frecuencias", xlab= "Intervalos de clase para la edad",
     col = "#1C464D")
```

Mi primer histograma



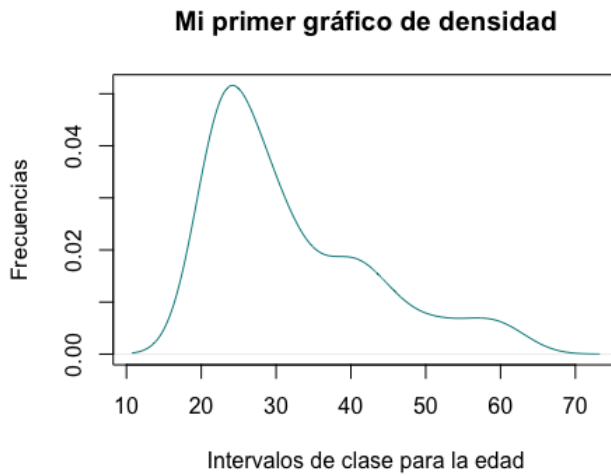
La universidad de [Columbia desarrolló un manual](#) que contiene algunos de los colores que pueden ser utilizados en R. Usted puede aumentar esta gama de colores mediante la instalación de librerías con paletas de colores.

Gráficos de densidad

La forma de los histogramas se ve fuertemente afectada por la cantidad de intervalos de clase utilizados, por lo que a veces es útil trazar un gráfico de densidad en su lugar. En R, podemos crear un gráfico de densidad usando la función de `density()`.

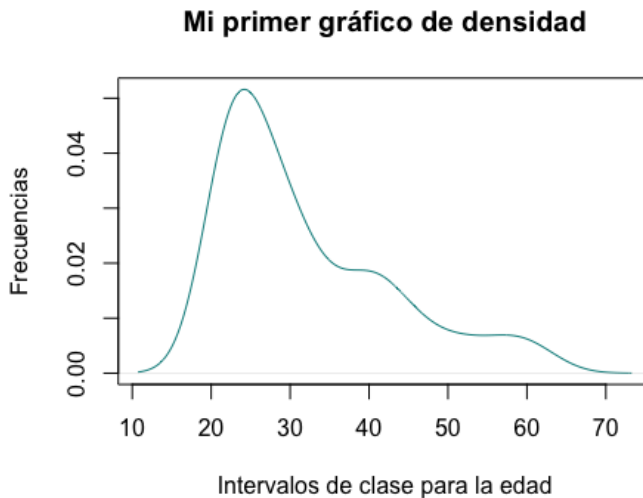
♦ Ejemplo 10.2

```
plot(density(Pima.tr$age), main = "Mi primer gráfico de densidad",
     ylab = "Frecuencias",
     xlab= "Intervalos de clase para la edad", col = "cyan4")
```



Si utiliza pipes, el código puede ser construido de la siguiente forma:

```
density(Pima.tr$age)|>
  plot(main = "Mi primer gráfico de densidad",
        ylab = "Frecuencias",
        xlab = "Intervalos de clase para la edad", col = "cyan4")
```



Este código merece algunas aclaraciones. La primera es que se utilizó la función `plot()`, esta función es genérica en el que R tratará de graficar lo más ideal para los datos con las que fue alimentada. Por ejemplo, la podríamos utilizar para graficar un boxplot, si la alimentamos con los argumentos correctos. Segundo, note cómo los argumentos para modificar los aspectos estéticos de la gráfica, se encuentran aplicados a la función `plot()` y no a la función de densidad.

La función `plot()` dada su complejidad merece que se dedique un apartado especial a ella ([“La función plot\(\)”](#))

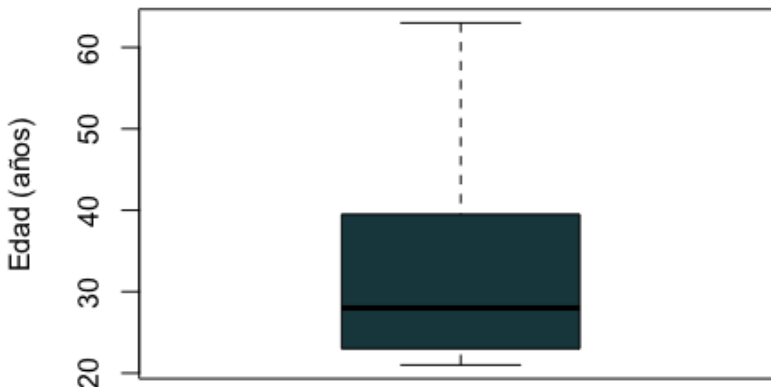
Boxplot

Ya hemos utilizado la función `boxplot()` con anterioridad en el apartado [“Introducción a las funciones”](#). La función `boxplot()` contiene ciertas peculiaridades que es importante aclarar.

- ♦ **Ejemplo 10.3:** para realizar un boxplot de las edades de las pacientes de la base de datos `Pima.tr` puede utilizar el siguiente código:

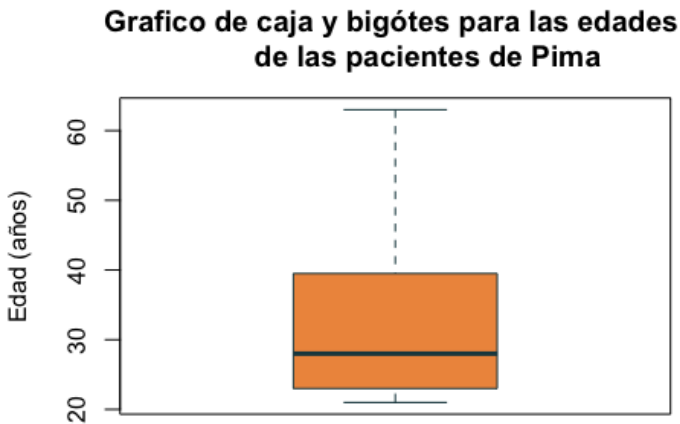
```
boxplot(Pima.tr$age, main="Gráfico de caja y bigotes para las edades
de las pacientes de Pima", ylab="Edad (años)", col="#1C464D")
```

Gráfico de caja y bigotes para las edades de las pacientes de Pima



Note cómo hay un “enter” en el título, que se ve reflejado en la gráfica. También podemos cambiar el color del contorno de la caja con el argumento `border`.

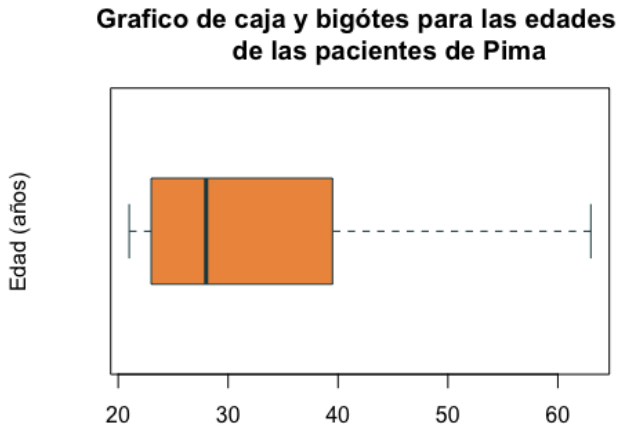
```
boxplot(Pima.tr$age,
main="Gráfico de caja y bigótes para las edades
      de las pacientes de Pima",
ylab="Edad (años)", col="#EE964B",
border="#1C464D")
```



Note cómo ahora nos referimos a los colores con un código que comienza por un símbolo de `#`. Puede encontrar mayor información de los colores [aquí](#).

En algunas ocasiones es de nuestro interés tener una gráfica de forma horizontal. Para ello, utilizamos el argumento `horizontal = T`. Recuerde que la `T` en R es igual a `TRUE`. Puede utilizar ambas:

```
boxplot(Pima.tr$age,
main="Gráfico de caja y bigótes para las edades
      de las pacientes de Pima",
ylab="Edad (años)", col="#EE964B",
border="#1C464D", horizontal = T)
```

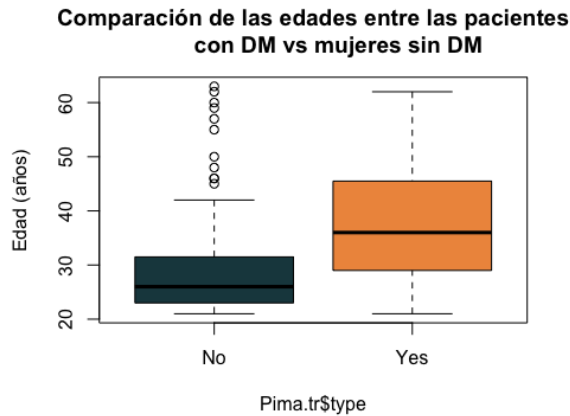


Note que las funciones por defecto tienen predefinidos algunos argumentos, en este caso el argumento horizontal se encuentra predefinido como FALSE.

Boxplot por grupos

- ♦ **Ejemplo 10.4:** ¿qué pasaría si queremos graficar en un boxplot las edades de las pacientes de Pima.tr y agruparlas de acuerdo con la presencia de diabetes? Para ello, podemos emplear el siguiente código (que después explicaré):

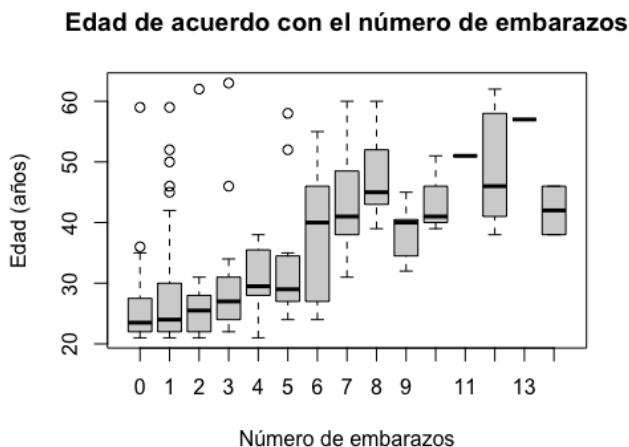
```
boxplot(Pima.tr$age~Pima.tr$type,
  main="Comparación de las edades entre las pacientes
    con DM vs mujeres sin DM",
  ylab="Edad (años)",
  col = c("#1C464D", "#EE964B"))
```



La principal peculiaridad del código es: `Pima.tr$age~Pima.tr$type`. Esta estrategia se conoce en R como “fórmula” y se lee como: la variable `age` agrupada por `type`. En este caso el símbolo `~` significa agrupación.

- ♦ **Ejemplo 10.5:** ahora supongamos que queremos saber si hay una relación entre la edad y el número de embarazos:

```
boxplot(Pima.tr$age~Pima.tr$npreg,
        main="Edad de acuerdo con el número de embarazos",
        ylab = "Edad (años)", xlab="Número de embarazos")
```

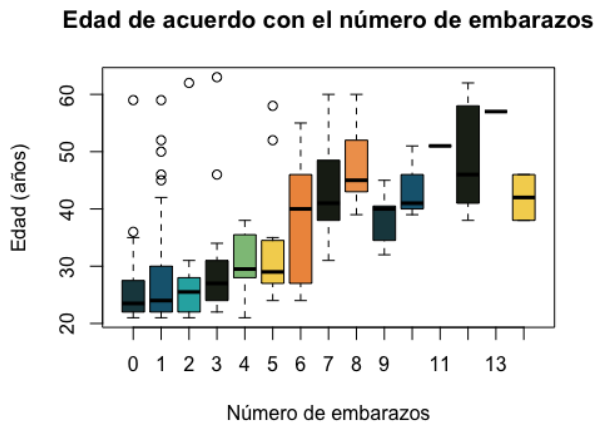


- ♦ **Ejemplo 10.6:** ahora, vamos a crear una paleta de colores para cambiar los colores de las cajas. Vamos a crear un vector con los nombres de los colores y después vamos a utilizarlo:

```
Mi_Paleta <- c("#1C464D", "#19647E", "#28AFB0", "#1F271B",
               "#8EC187", "#F4D35E", "#EE964B", "#1C2319",
               "#F0A05B")
```

Ahora al utilizar el argumento `col` lo asignaremos a nuestra paleta:

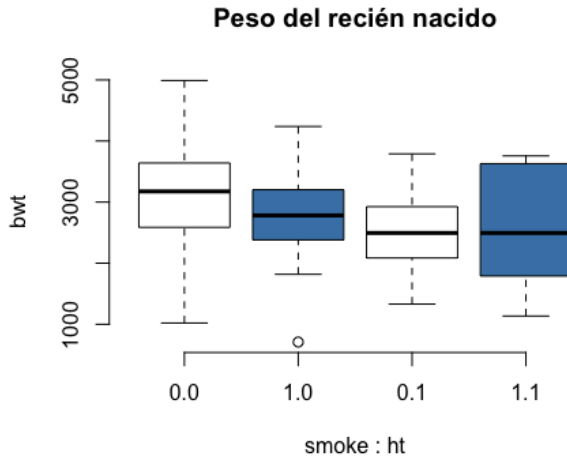
```
boxplot(Pima.tr$age~Pima.tr$npreg,
        main="Edad de acuerdo con el número de embarazos",
        ylab = "Edad (años)", xlab="Número de embarazos",
        col=Mi_Paleta)
```



Note cómo los colores se repitieron, ya que hay 14 cajas y solo 9 colores. Las agrupaciones pueden ser mas complejas y tener dos variables de agrupación. Utilicemos el *Dataframe* `birwth` para explicarlo mejor.

- ♦ **Ejemplo 10.7:** supongamos que queremos realizar una gráfica del peso del bebé al nacer (`bwt`) agrupado por el antecedente de tabaquismo en el embarazo (`smoke`) y la historia de hipertensión (`ht`). Podemos emplear el siguiente código:


```
boxplot(bwt~smoke*ht,
        data = birthwt, main="Peso del recién nacido",
        col = c("white", "steelblue"), frame = FALSE)
```



La sección de código `bwt~smoke*ht` indica el peso del recién nacido agrupado por el antecedente de tabaquismo y de hipertensión de la madre. Con el argumento `data` podemos indicar de dónde puede R obtener los datos y, por tanto, no es necesario utilizar el símbolo `$`.

Gráficos de violín

Existe una alternativa a los `boxplot` que permite una apreciación de mayor calidad de la distribución de los datos. Estos gráficos se conocen como gráficos de violín o *violin plots*. Los cuales conjuntan la distribución (forma) de los datos y un `boxplot`. La librería `vioplot` contiene la función `vioplot()` que requiere de un vector numérico o de una fórmula para alimentarse.

♦ **Ejemplo 10.8:** procedemos a instalar y a cargar la librería:

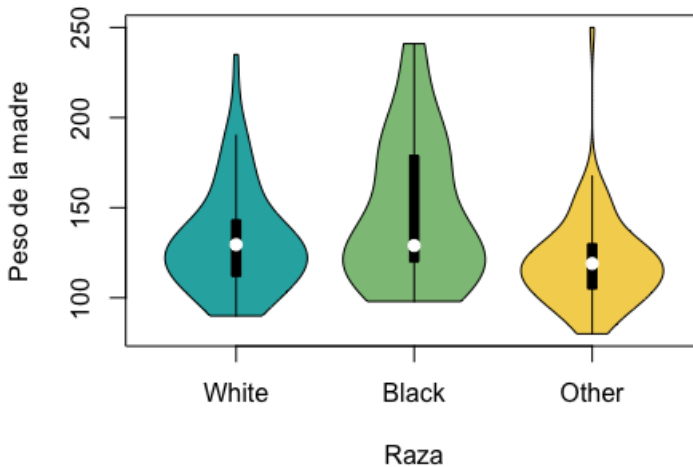
```
install.packages("vioplot")
library("vioplot")
```

Cambiamos las etiquetas de la variable race y la convertimos a factor:

```
birthwt$race <- factor(birthwt$race,
                       levels = c(1,2,3),
                       labels = c("White", "Black", "Other"))
```

Las etiquetas de la variable race se obtuvieron después de utilizar `help(birthwt)`:

```
vioplot(birthwt$lwt~birthwt$race,
        ylab="Peso de la madre",
        xlab = "Raza",
        col=c("#28AFB0", "#8EC187", "#F4D35E"))
```



La función plot()

La función `plot()` es una de las funciones más genéricas, funcionales y grandiosas que existen en R. Me explico. Esta función permite generar diferentes tipos de gráficas, simplemente haciendo modificaciones en dos argumentos (`x` y `y`),

permite generar desde gráficos de puntos (Scatterplot), boxplot y hasta resultados de ciertas pruebas de hipótesis. La función `plot()` necesita al menos del argumento `x` y el resultado de la gráfica dependerá de las características de este argumento y de si viene o no acompañado de otro argumento, el argumento `y`. Esta función entiende lo que necesitamos siempre y cuando la alimentemos de forma correcta. La [tabla 10.1](#) muestra el comportamiento general de la función `plot`, según los argumentos `x` y `y`.

TABLA 10.1. COMPORTAMIENTO DE LA FUNCIÓN `PLOT()`

ARGUMENTO X	ARGUMENTO Y	RESULTADO
Ninguno		Error (x siempre es necesario)
Entero/numérico	Entero/numérico	Scatterplot
Entero/numérico	Ninguno	Gráficos de puntos por variable de indexación
Factor	Numérico	Boxplot
Factor	Factor	Gráfico de mosaico
Factor	Ninguno	Gráfico de barras/columnas

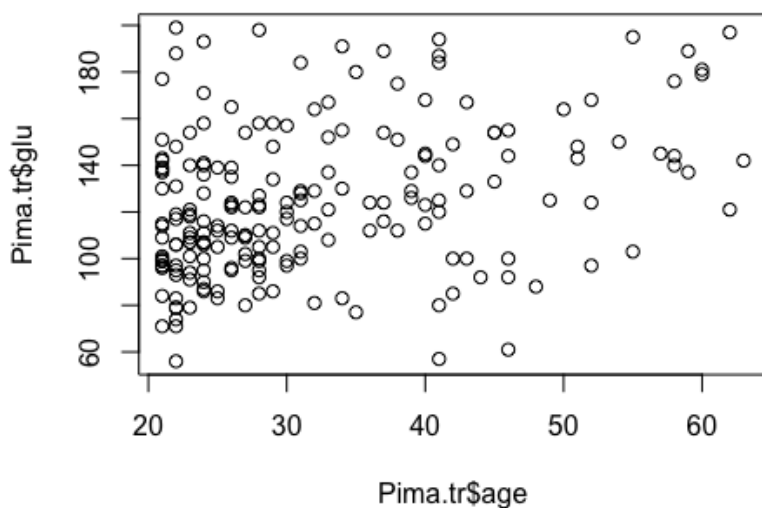
Scatterplot

Los Scatterplot son representaciones que permiten graficar dos variables cuantitativas, una para el eje de las “x” y otra para el eje de las “y”. Estas gráficas son útiles para identificar la relación entre dos variables cuantitativas, identificar tendencias de una variable a lo largo del tiempo, detección valores atípicos y dispersión de los datos.

En el caso de R estos gráficos se construyen utilizando la función `plot()`.

- ♦ **Ejemplo 10.9:** ¿existirá relación entre la edad y las concentraciones de glucosa en las mujeres de `Pima.tr`? Podemos comenzar con construir un Scatterplot:

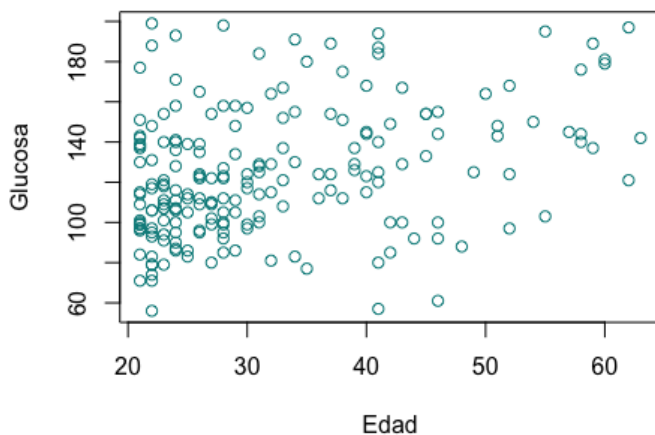
```
plot(Pima.tr$age, Pima.tr$glu)
```



Al igual que las gráficas anteriores, podemos cambiar la estética de la gráfica añadiendo otros argumentos:

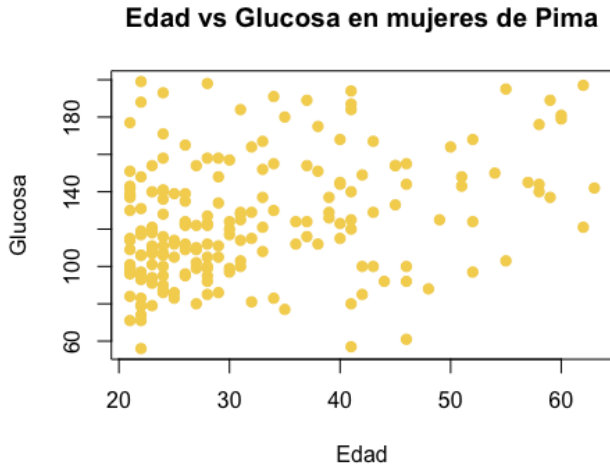
```
plot(Pima.tr$age, Pima.tr$glu,
     main = "Edad vs Glucosa en mujeres de Pima",
     xlab="Edad", ylab="Glucosa", col="cyan4")
```

Edad vs Glucosa en mujeres de Pima



Podemos modificar la forma de los puntos con el argumento `pch`:

```
plot(Pima.tr$age, Pima.tr$glu,
     main = "Edad vs Glucosa en mujeres de Pima",
     xlab="Edad", ylab="Glucosa",
     col="#F4D35E", pch=19) #Se cambio el color
```



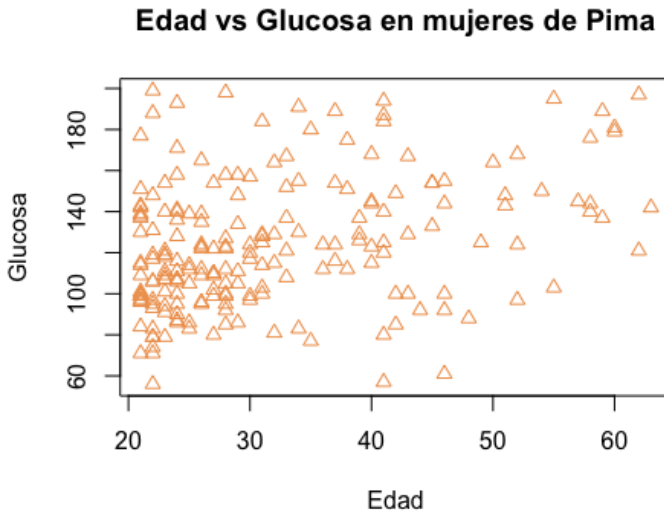
En la [figura 10.1](#) se muestran las opciones de puntos disponibles en R.

FIGURA 10.1. OPCIONES PARA EL ARGUMENTO PCH DE LA FUNCIÓN PLOT



- ♦ **Ejemplo 10.10:** podemos cambiar ahora la forma de los puntos de nuestra gráfica. Pruebe la forma que usted prefiera:

```
plot(Pima.tr$age, Pima.tr$glu,
     main = "Edad vs Glucosa en mujeres de Pima",
     xlab="Edad", ylab="Glucosa",
     col="#F0A05B", pch=24) #Se cambio el color y forma
```



- ♦ **Ejemplo 10.11:** las gráficas en R se pueden construir en capas (aunque en realidad son varias gráficas). Por ejemplo, a la gráfica anterior le podemos agregar texto, líneas, etcétera.

De nuevo, no se preocupe si no comprende muchos de los términos. Cuando necesite construir una gráfica, siempre puede regresar a este capítulo. Vamos a calcular el coeficiente Pearson:

```
cor.test(Pima.tr$age, Pima.tr$glu)
```

Pearson's product-moment correlation

data: Pima.tr\$age and Pima.tr\$glu

t = 5.1451, df = 198, p-value = 6.415e-07

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.2149050 0.4602209

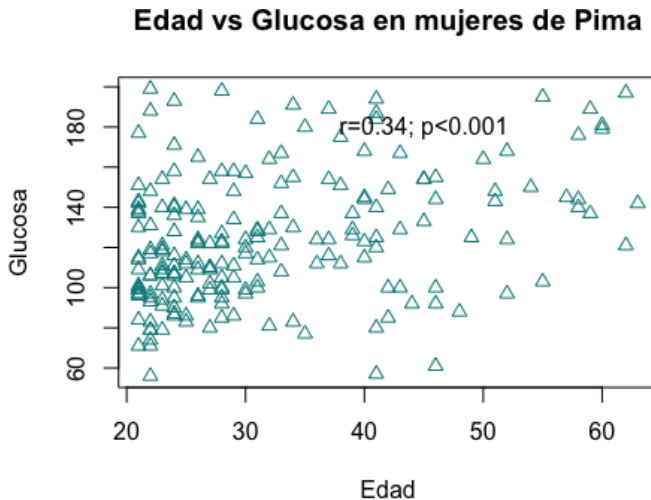
sample estimates:

cor

0.343407

Ahora vamos a agregar el texto a nuestra gráfica, utilizando la función `text`. Recuerde que necesitamos capas, por lo que primero necesitamos una gráfica a la cual agregar el texto:

```
#Capa principal
plot(Pima.tr$age, Pima.tr$glu,
     main = "Edad vs Glucosa en mujeres de Pima",
     xlab="Edad", ylab="Glucosa", col="cyan4", pch=24)
# Segunda capa
text(x=45, y=180, label="r=0.34; p<0.001")
```



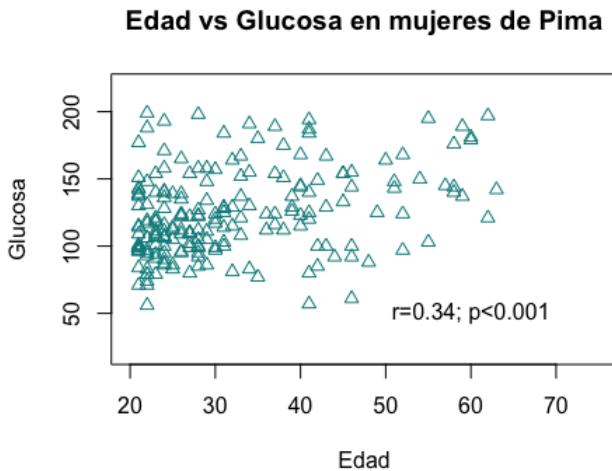
Podemos notar que la función `text()` requiere que indiquemos las coordenadas en donde será colocado el texto (argumentos “x” y “y”) y el texto *per se*, en este caso con el argumento `label`. Note cómo el texto va entre comillas.

Ejemplo 10.12: ¿qué pasaría si queremos que el texto no se encime con los puntos? Una de las opciones es cambiar los límites de los ejes y cambiar las coordenadas del texto, colocándole donde no se encimen:

```
#Capa principal
plot(Pima.tr$age, Pima.tr$glu,
```

```
main = "Edad vs Glucosa en mujeres de Pima",
xlab="Edad", ylab="Glucosa", col="cyan4", pch=24,

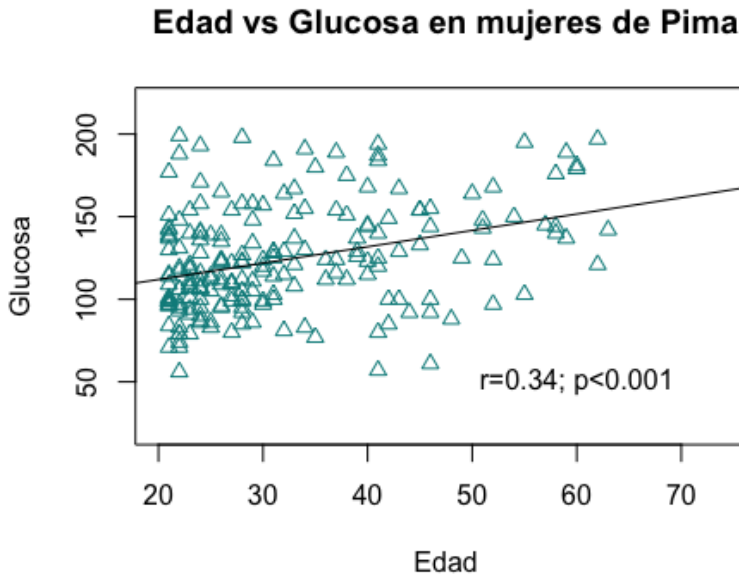
xlim = c(20,75), ylim=c(20,220))
# Segunda capa
text(x=60, y=50, label="r=0.34; p<0.001")
```



Observe como los argumentos `xlim` y `ylim` requieren de un valor mínimo y un valor máximo que deben de ir concatenados con la función `c()`.

- ♦ **Ejemplo 10.13:** podemos ahora agregar la línea de tendencia de la regresión utilizando la función `lm()`, la cual revisaremos más adelante:

```
#Capa principal
plot(Pima.tr$age, Pima.tr$glu,
     main = "Edad vs Glucosa en mujeres de Pima",
     xlab="Edad", ylab="Glucosa", col="cyan4", pch=24,
     xlim = c(20,75), ylim=c(20,220))
# Segunda capa
text(x=60, y=50, label="r=0.34; p<0.001")
# Tercera capa
abline(lm(Pima.tr$glu~Pima.tr$age))
```

La función `abline` agrega una línea sobre una gráfica ya construida, en este caso agregamos la línea de regresión para marcar la tendencia. Tanto la función `abline()` como la función `text()` no tienen sentido sin una gráfica previa. De hecho, sin una gráfica previa producirán un error.

Barplot

Los barplots o gráficos de barras se utilizan para representar datos categóricos. Estos gráficos permiten mostrar y comparar conteos, frecuencias, porcentajes de factores agrupados o sin agrupas. En su forma clásica, estos gráficos muestran en el de las “x” y en el eje de las “y” conteos o porcentajes. Aunque también existe una versión horizontal para estos. Los autores de este libro consideramos que los gráficos de barras no son los más adecuados para la representación de variables cuantitativas.

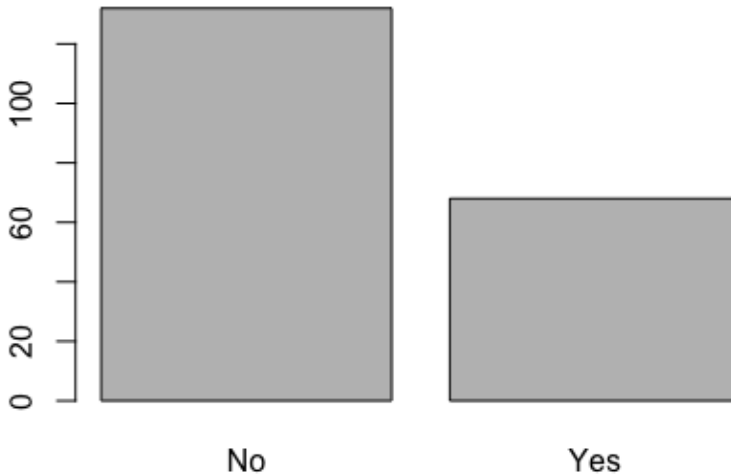
Para construir un gráfico de barras, usted debe de utilizar la función `barplot()` o en su defecto la función `plot()`. Para poder funcionar adecuadamente `barplot()` necesita de conteos, por lo que primero debemos de aplicar la función `table()` para obtener estos conteos.

- ♦ **Ejemplo 10.14:** vamos a utilizar la función `barplot()` para representar cuántas pacientes tienen diabetes y cuántas pacientes no lo tienen, para ello utilizamos el siguiente código:

```
# Si no ha importado la base hágalo

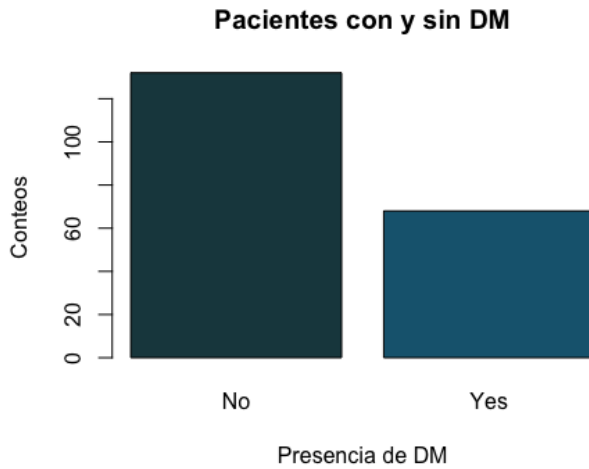
# Necesitamos declarar y crear una tabla con los conteos
# Creamos un objeto llamado tab con los conteos
tab <- table(Pima.tr$type)

# Ahora ya podemos graficar nuestro barplot
barplot(tab)
```



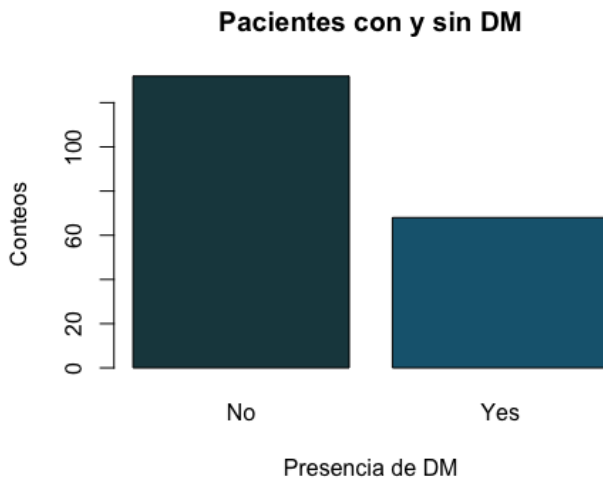
Podemos mejorar esta gráfica poco a poco:

```
barplot(tab, main = "Pacientes con y sin DM",
        xlab="Presencia de DM",
        ylab="Conteos", col= Mi_Paleta)
```



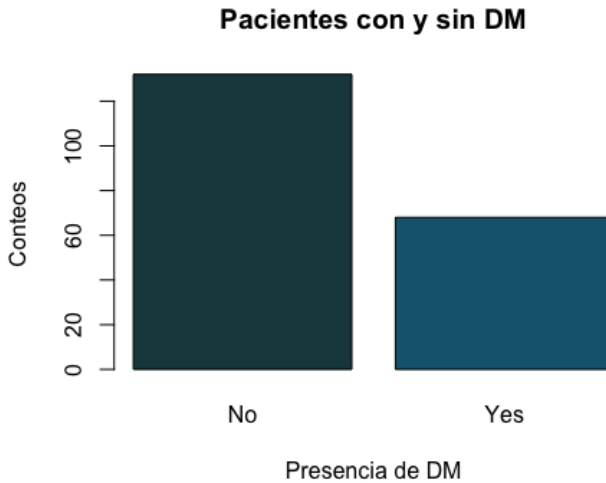
La gráfica puede crearse utilizando pipes:

```
table(Pima.tr$type) |>  
  barplot(main = "Pacientes con y sin DM",  
          xlab="Presencia de DM",  
          ylab="Conteos", col= Mi_Paleta)
```



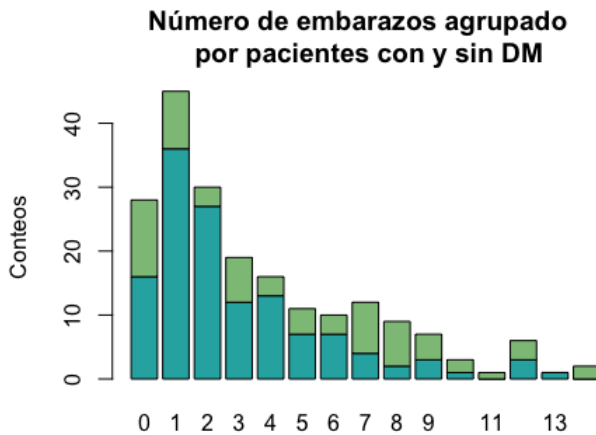
Utilizando la función `plot()` el código sería:

```
plot(Pima.tr$type, main = "Pacientes con y sin DM",
     xlab="Presencia de DM",
     ylab="Conteos", col= Mi_Paleta)
```



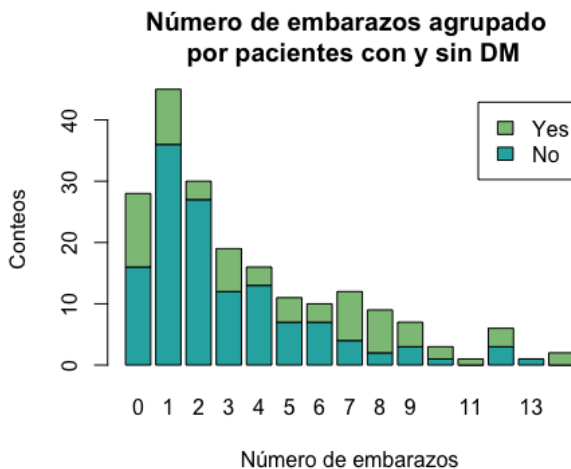
- ♦ **Ejemplo 10.15:** podemos aumentar el número de clasificaciones que pueden ser representadas en el barplot, si aumentamos la complejidad de nuestra tabla. Por ejemplo, la siguiente tabla muestra el número de embarazos agrupado entre las pacientes con y sin DM:

```
table(Pima.tr$type, Pima.tr$npreg)>
barplot(
  main="Número de embarazos agrupado \n por pacientes con y sin DM",
  ylab = "Conteos",
  xlab="Número de embarazos",
  col = c("#28AFB0", "#8EC187"))
```



- ♦ **Ejemplo 10.16:** el resultado es poco entendible y necesita ser trabajado para producir una gráfica más amigable. Vamos a agregar leyendas para los colores:

```
table(Pima.tr$type, Pima.tr$npreg)|>
  barplot(
    main="Número de embarazos agrupado \n por pacientes con y sin DM",
    ylab = "Conteos", xlab="Número de embarazos",
    col = c("#28AFB0", "#8EC187"), legend=T)
```

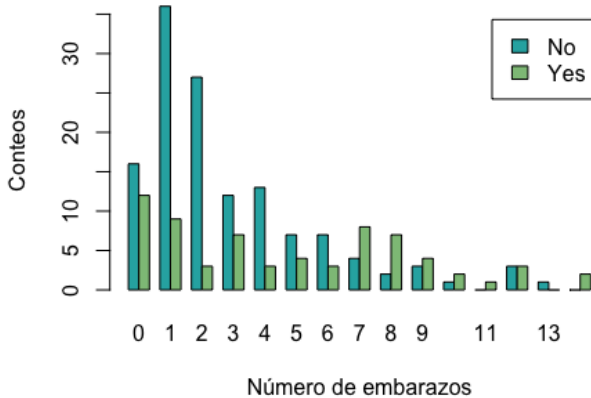


El argumento `legend=T` nos ayuda a mostrar las leyendas.

- ♦ **Ejemplo 10.17:** podemos poner las barras a un lado de la otra utilizando el argumento `beside = T`:

```
table(Pima.tr$type, Pima.tr$npreg)|>
  barplot(
    main="Número de embarazo agrupado por pacientes con y sin DM",
    ylab = "Conteos", xlab="Número de embarazos",
    col = c("#28AFB0", "#8EC187"),
    legend=T, beside = T)
```

Número de embarazo agrupado por pacientes con y sin



- ♦ **Ejemplo 10.18:** vamos a agregar algunos colores, pero esta vez utilizando una librería, en este caso `grDevices`:

```
# Si es necesario instale
# install.packages("grDevices")
library(grDevices)
```

Tener muchos colores no tendría sentido en la comparación del número de embarazos entre las pacientes con y sin diabetes. Por lo tanto, creamos una nueva tabla únicamente con el número de embarazos:

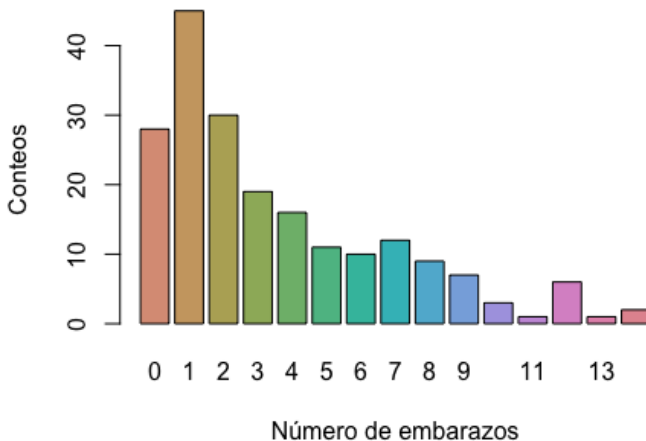
```
tab2 <- table(Pima.tr$npreg)
```

Posteriormente, podemos crear nuestras propias paletas de 14 colores (los que necesitamos para cada número de embarazo):

```
n <- 15 # Cuantos colores necesito
p1 <- hcl.colors(n, palette = "Dynamic")
#Tomamos 15 colores de la paleta dynamic
p2 <- hcl.colors(n, palette = "Earth")
p3 <- hcl.colors(n, palette = "Berlin")
p4 <- hcl.colors(n, palette = "Fall")
p5 <- hcl.colors(n, palette = "Sunset")
```

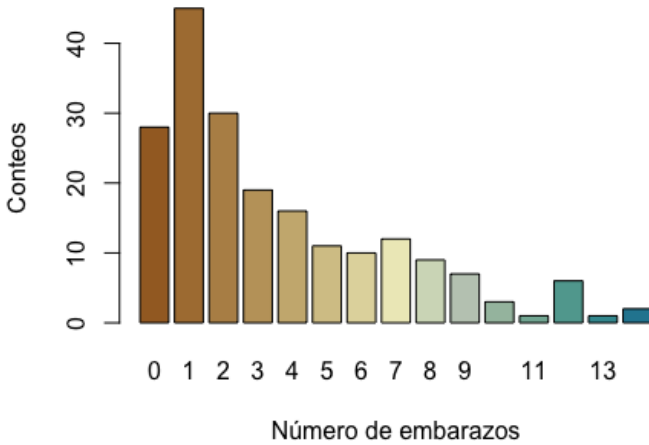
Ahora podemos probar cómo queda nuestra gráfica:

```
barplot(tab2, xlab="Número de embarazos",
        ylab="Conteos", col=p1)
```

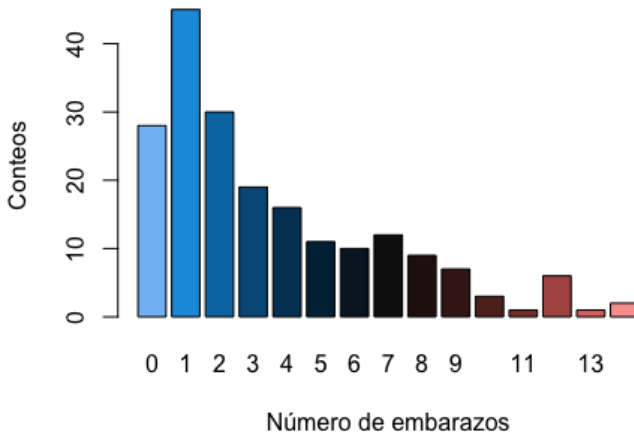


Podemos probar otras paletas:

```
barplot(tab2, xlab="Número de embarazos",
        ylab="Conteos", col=p2)
```

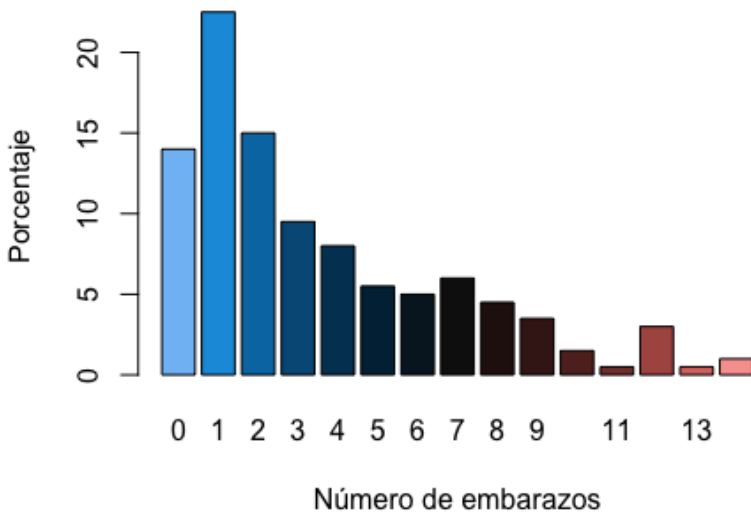


```
barplot(tab2, xlab="Número de embarazos",
        ylab="Conteos", col=p3)
```



- ♦ **Ejemplo 10.19:** podemos utilizar la función `prop.table()` en el caso que nuestro interés sea graficar porcentajes en lugar de conteos:

```
tab3 <- prop.table(tab2)*100
# Para obtener porcentajes es necesario multiplicar por 100
barplot(tab3, xlab="Número de embarazos",
        ylab="Porcentaje", col=p3)
```



En R existen muchas librerías mediante las cuales se pueden generar paletas. Algunas de ellas son:

- ♦ **RColorBrewer:** esta librería es una de las más populares para manejar paletas de colores en R. Proporciona paletas de colores diseñadas para ser estéticamente agradables y también perceptualmente uniformes.
- ♦ **viridis:** ofrece paletas de colores que son perceptualmente uniformes en términos de brillo y color. Estas paletas también se reproducen de manera consistente en impresiones en blanco y negro.
- ♦ **scales:** esta librería proporciona funciones para crear paletas de colores personalizadas y también para ajustar las propiedades de las paletas, como la luminosidad y la saturación.

- ♦ **colorspace**: permite crear una gran variedad de paletas de colores, incluyendo secuenciales, divergentes y cualitativas. También es posible ajustar estas paletas para mejorar la accesibilidad.
- ♦ **ggsci**: esta librería contiene paletas de colores basadas en las populares paletas utilizadas en publicaciones científicas, como las de *Nature*, *Lancet*, y *JCO*. Es ideal para quienes desean replicar estilos de gráficos de publicaciones específicas.
- ♦ **paletteer**: proporciona una interfaz completa para acceder a una amplia gama de paletas de colores de diversas librerías de R, facilitando la exploración y aplicación de diferentes paletas sin necesidad de cargar múltiples paquetes.

Mosaic plot (gráfico de mosaicos)

Es otra forma de representar datos categóricos, sobre todo cuando los datos provienen de una tabla cruzada. Se podría decir que el gráfico de mosaicos es un gráfico de barras segmentado. Aunque revisaremos más a detalle los conceptos de tabla de cruzadas y tablas de contingencia en el [capítulo 19](#) y en el [capítulo 20](#), realizaremos una descripción sencilla de las mismas. Las tablas cruzadas permiten analizar la relación entre dos variables y permiten conocer valores que cumplen con determinada característica al representar los datos en filas y columnas.

Imagine que estamos interesados en saber cuántas de las madres que fuman tuvieron bebés con bajo peso al nacer. Para ello utilizamos la base de datos `birthwt` de la librería `MASS`:

```
library(MASS) # Cargar la librería, en el
              # caso que no la tenga en su ambiente de trabajo
data("birthwt") # Llamar la base de datos
```

Si requiere recordar las variables de la base `birthwt` y su significado, puede utilizar `help("birthwt")`.

Vamos a modificar las etiquetas de los factores:

```
birthwt$smoke ← factor(birthwt$smoke,
                        labels=c("No fumó", "Sí fumó"))
# Como se conocen el orden los niveles no es necesario especificarlos
birthwt$low ← factor(birthwt$low,
                     labels= c("Peso normal", "Bajo peso"))
```

Si queremos conocer cuántos pacientes fuman y de forma separada cuántos recién nacidos presentaron bajo peso al nacer, podemos emplear el siguiente código:

```
table(birthwt$smoke)

No fumó      Sí fumó
115          74

table(birthwt$low)

Peso normal  Bajo peso
130          59
```

Las anteriores son solo tablas de frecuencias, no una tabla de contingencia. Se pueden crear tablas de contingencia con la función `table` ingresando dos variables:

```
table(birthwt$smoke, birthwt$low)

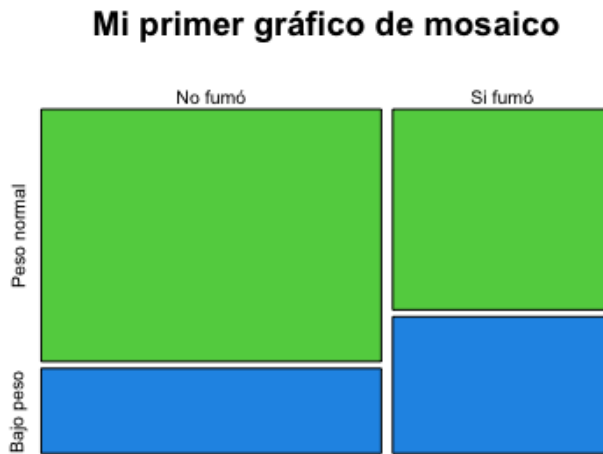
      Peso normal Bajo peso
No fumó    86      29
Sí fumó    44      30
# smoke queda en la filas y low en las columnas

table(birthwt$low, birthwt$smoke)

      No fumó  Sí fumó
Peso normal   86     44
Bajo peso     29     30
# low queda en las filas y smoke en las columnas
```

Ahora para realizar el gráfico de mosaico utilizamos el siguiente código:

```
table(birthwt$smoke, birthwt$low)|>
  mosaicplot(col=3:6, main= "Mi primer gráfico de mosaico")
```



La explicación del código es la siguiente: `table(birthwt$smoke, birthwt$low)` nos permitió generar la tabla de contingencia, con `|>` trasladamos ese resultado a la siguiente función, la función `mosaicplot()` nos permitió generar el gráfico utilizando los datos de la tabla de contingencia. En esta ocasión seleccionamos los colores del 3 al 6.

Note cómo el gráfico nos permite identificar rápidamente las proporciones de cada una de las categorías.

Gráficos de sectores

Los gráficos de sectores ayudan a representar proporciones de una variable cualitativa en una circunferencia. Representa las categorías de una variable mediante proporciones. Para estimar la proporción de cada una de las categorías se utiliza la siguiente fórmula:

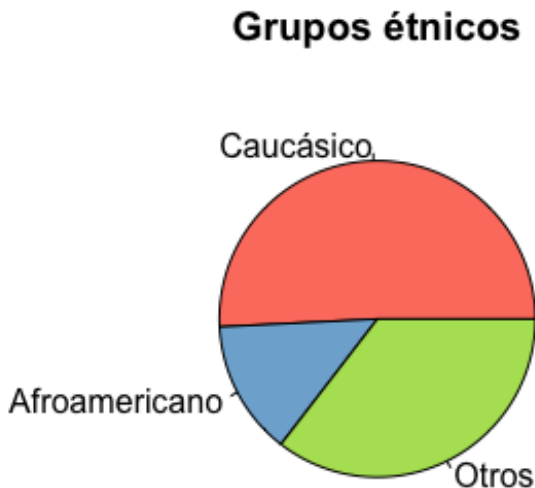
Donde es la frecuencia absoluta y el total de las observaciones de la muestra. El 360 estará siempre en la fórmula y no variará porque es necesario para que el diagrama de sectores tenga una forma circular.

El gráfico de sectores, también conocido como gráfico circular o gráfico de pastel, es una representación visual utilizada para mostrar la proporción o distribución de diferentes categorías dentro de un conjunto de datos. Cada categoría se representa como un sector circular en el gráfico, y el tamaño de cada sector es proporcional a la frecuencia o proporción de esa categoría en relación con el total.

- ♦ **Ejemplo 10.20:** vamos realizar un diagrama de sectores para la variable `race` de `birthwt`.

Primero, modificamos la variable con las etiquetas adecuadas:

```
birthwt$race <- factor(birthwt$race,
                        labels = c("Caucásico", "Afroamericano", "Otros" ))
table(birthwt$race)|>
  pie(main = "Grupos étnicos",
      col = c("#fd7f6f", "#7eb0d5", "#b2e061"))
```



Note cómo de nuevo utilizamos la función `table()` para obtener los conteos de las categorías.

- ♦ **Ejemplo 10.21:** podemos modificar aún más nuestra gráfica, como se muestra con el siguiente código:

```
# Generar una tabla de frecuencias para la
# variable 'race' en el dataset 'birthwt'
tabla_frecuencias <- table(birthwt$race) |>

# Crear un gráfico circular con la tabla de frecuencias
pie(
  main = "Grupos étnicos", # Título del gráfico
  col = c("#fd7f6f", "#7eb0d5", "#b2e061"), # Colores
  density = 50, # Densidad de las líneas de sombreado en el gráfico
  angle = 45 # Ángulo de las líneas de sombreado
)

# Añadir una leyenda al gráfico para facilitar la interpretación
legend(
  "topright", # Posición de la leyenda dentro del gráfico
  c("Caucásico", "Afroamericano", "Otros"), # Etiquetas
  fill = c("#fd7f6f", "#7eb0d5", "#b2e061") # Colores
```



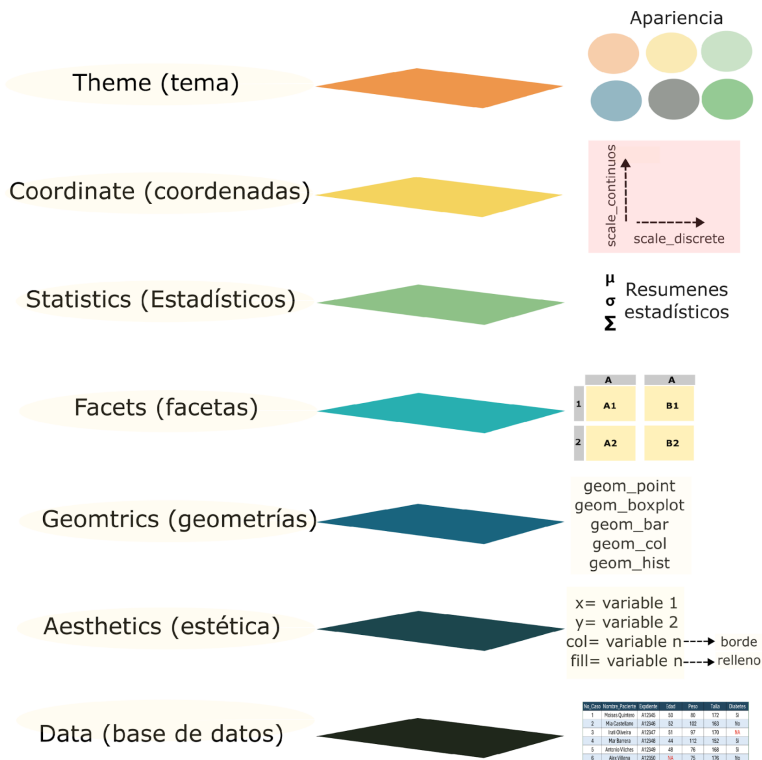
ggplot2

`ggplot2` es una de las librerías más utilizadas en R, de hecho podría considerarse el estándar para construcción de gráficos, ya que permite la construcción de diferentes tipos de gráficos. Puede echar un vistazo sobre las gráficas que se pueden hacer utilizando `ggplot2` en el siguiente [enlace](#). Además, hay diversas paqueterías que están construidas sobre `ggplot2` y que aprovechan su sintaxis para mejorar aún más las gráficas, por ejemplo: `ggpurbr`, `ggstastplot`, `gggenomes`, `plotly`, `ggbridges`, `ggrepel`, `cowplot`, `ggforce`, `gganimate`, `ggtext`, `patchwork`. Relacionado a ciencias de la salud hay diversas librerías que le pudieran ser de utilidad: `ggsurvplot`, `ggforest`, `ggmosaic`, `ggpmisc` y `gggortify`. Por si no fuera poco, gran parte de los gráficos relacionados con genómica, RNA-seq y expresión génica se realizan en el contexto de `ggplot2`.

Ahora bien, ¿qué es `ggplot2`? Es una librería que permite la creación de gráficos de forma estructurada y que se encuentra basada en *The Grammar of Graphics* (de ahí la `gg`), “*Create Elegant Data Visualisations Using the Grammar of Graphics*” (s. f.). Simplemente usted aporta los datos, define variables y la estética como las variables y `ggplot2` se encarga de lo demás.

`ggplot2` funciona por medio de capas, cada una de ellas aporta un componente en específico a su gráfico. Es importante que entienda el comportamiento de estas capas, ya que le permitirá modificar y crear gráficas de una forma más eficiente. El error más común en `ggplot2` es querer modificar la capa superior, pero utilizando un código para modificar la capa más profunda. La [figura 10.2](#) muestra las principales capas y sus principales características.

FIGURA 10.2. CAPAS DE GGPLOT



Vamos a explicar de forma muy general el funcionamiento de las capas en ggplot2. Para ello, vamos a utilizar el *Dataframe* `Pimar.tr` de la librería `MASS`.

- ♦ **Ejemplo 10.22:** supongamos que es nuestro interés observar la relación entre el IMC y la glucosa en ayunas de las mujeres de Pima. Para ello, vamos a construir una gráfica utilizando `ggplot2`.

Si lo requiere instale la librería ggplot2:

```
# Si lo requiere
install.packages("ggplot2")
```

Llamamos la librería:

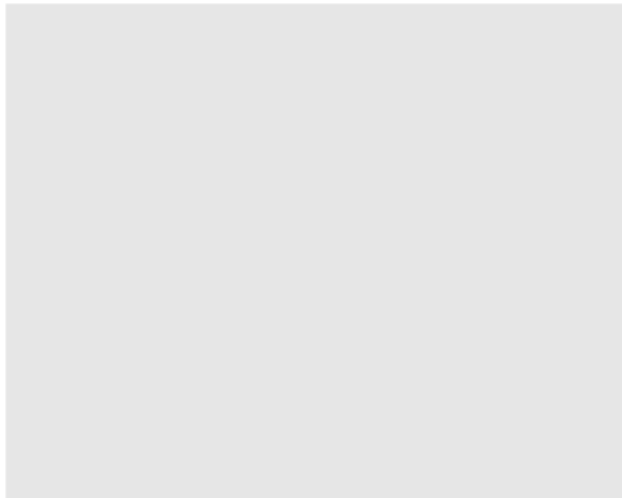
```
library(ggplot2)
```


Importamos la base de datos:

```
library(MASS)
data("Pima.tr")
```

La primera capa de ggplot2 son los datos. ggplot2 necesita un *Dataframe* para poder construir un gráfico, en ciertas circunstancias no es necesario un *Dataframe*, pero sí la mayoría de los casos. Esta es una de las principales diferencias entre construir gráficas en el paquete base y en ggplot2:

```
# Primera capa de ggplot2
ggplot(data = Pima.tr)
```

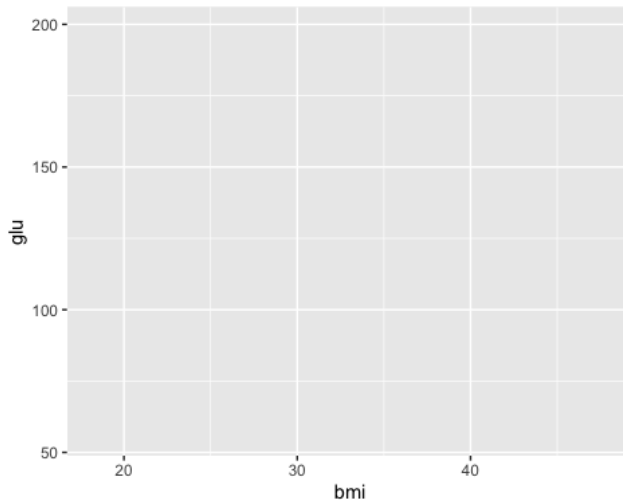


Si utiliza pipes, el código es el siguiente:

```
Pima.tr|>
  ggplot()
```

Como lo habrá notado tenemos un lienzo en blanco (en gris en este caso), hemos construido nuestra primera capa. Ahora tenemos que indicarle qué variables deben de ir en el eje de las “x” y cuál en el de las “y” utilizando la siguiente capa (aesthetic):

```
ggplot(data = Pima.tr, mapping = aes(x=bmi, y=glu))
```



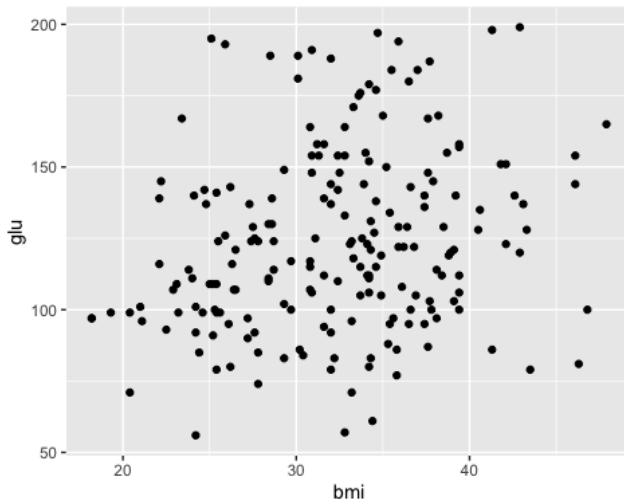
O bien:

```
Pima.tr|>
  ggplot(mapping = aes(x=bmi, y=glu))
```

Para indicarle a ggplot2 qué variables deben de conformar el eje de las “x” y el eje de las “y” utilizamos el código: `mapping = aes(x=bmi, y=glu)`, donde indicamos la estética con `aes` mediante el “mapeo” `mapping`. Una explicación más técnica de esto sería: el argumento `mapping` en `ggplot2` se utiliza para definir el “mapeo” estético de las variables de su conjunto de datos a las propiedades visuales de los elementos del gráfico, como los ejes, colores, tamaños, formas, etc. Se especifica mediante la función `aes()`, que crea una lista de mapeos estéticos.

Hasta ahora no le hemos pedido a `ggplot2` que muestre una forma o geometría. La siguiente capa nos ayuda precisamente a esto, a indicarle a `ggplot2` mediante qué forma quiero que se muestren mis datos:

```
Pima.tr|>
  ggplot(mapping = aes(x=bmi, y=glu))+
  geom_point()
```



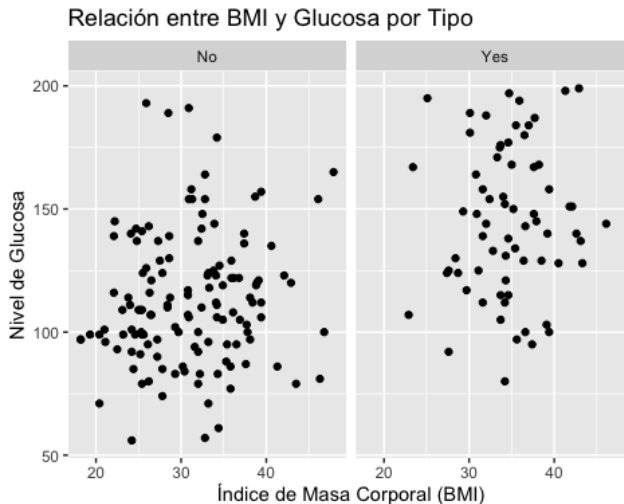
En `ggplot2`, los “geoms” (abreviatura de *geometries* o geometrías) son los componentes que determinan cómo se visualizan los datos en el gráfico, es decir, mediante qué geometría serán representados. Cada geom corresponde a un tipo específico de gráfico o elemento gráfico, como puntos, líneas, barras, cajas, etc. Los geoms se añaden a un objeto `ggplot` mediante funciones específicas que comienzan con `geom_`.

Note cómo las geometrías fueron añadidas utilizando `+`. En `ggplot2`, el signo `+` se utiliza para agregar capas y componentes adicionales a un objeto `ggplot`. Podríamos decir que el signo de `+` nos permite concatenar estos componentes del gráfico.

Hasta aquí tenemos nuestro primer gráfico utilizando `ggplot2`. Las tres capas que son necesarias para la construcción de un gráfico son: data, aesthetic y geometrics (ver [figura 10.2](#)).

- ♦ **Ejemplo 10.23:** ahora, supongamos que necesitamos crear dos gráficos de dispersión (uno al lado del otro) como el construido en el ejemplo anterior. Debe de mostrarse un gráfico para las mujeres con diabetes y otro para las mujeres sin diabetes. El siguiente código nos dará este gráfico:

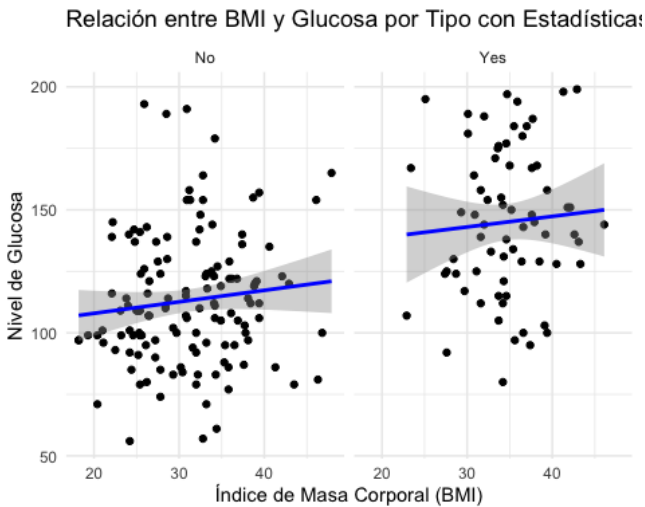
```
Pima.tr |>
  ggplot(mapping = aes(x = bmi, y = glu)) +
  geom_point() +
  facet_wrap(~ type) + # Agregar la capa de facetas
  labs(title = "Relación entre BMI y Glucosa por Tipo", # Cambiar títulos
       x = "Índice de Masa Corporal (BMI)", y = "Nivel de Glucosa")
```



La función `facet_wrap(~ type)` divide el gráfico en múltiples paneles basados en la variable `type`. Recuerde que el símbolo `~` se utiliza para agrupar. Cada panel muestra una subpoblación de los datos, permitiendo la comparación de la relación entre BMI y glucosa entre diferentes tipos. Además, en este código añadimos títulos a los ejes y al gráfico.

Imagine ahora que es de su interés agregar estimaciones estadísticas a su gráfico, por ejemplo, una línea con la ecuación de la recta que indique la tendencia de los datos, además queremos modificar el estilo del gráfico haciéndolo más minimalista:

```
# Crear el gráfico con facetas y capas estadísticas
Pima.tr |>
  ggplot(mapping = aes(x = bmi, y = glu)) +
  geom_point() +
  geom_smooth(method = "lm",
             se = TRUE, color = "blue") + # Línea de tendencia
  facet_wrap(~ type) + # Facetas
  labs(title = "Relación entre BMI y Glucosa por Tipo con Estadísticas",
       x = "Índice de Masa Corporal (BMI)", y = "Nivel de Glucosa") +
  theme_minimal()
`geom_smooth()` using formula = 'y ~ x'
```

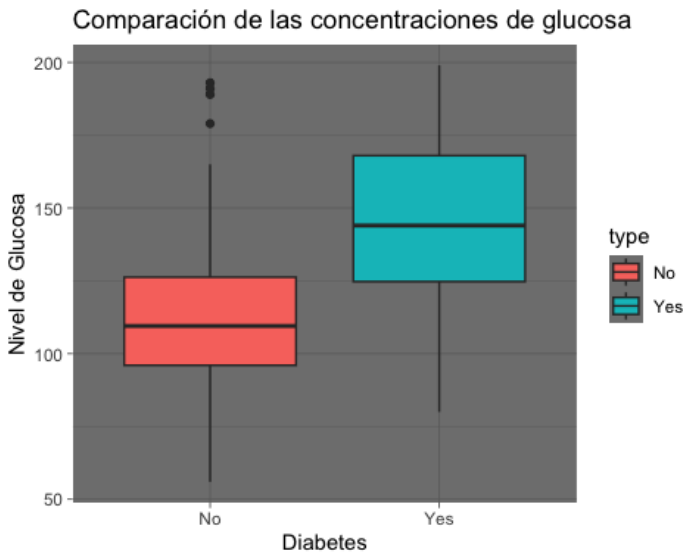


Parte de las nuevas líneas de código corresponden a la capa `statistics`, con lo cual agregamos la línea de tendencia. `geom_smooth()` añade una capa de línea de tendencia utilizando el método `method = "lm"` para regresión lineal. Note cómo a pesar de que `geom_smooth` está antes de la capa de faceta, `ggplot2` supo exactamente el orden jerárquico de las capas y primero separó por la variable `type` y luego estimó la línea de tendencia.

Finalmente, la última línea del código (`theme_minimal()`) nos permitió modificar la apariencia del gráfico final.

- ♦ **Ejemplo 10.24:** vamos ahora a crear un gráfico de boxplot para las concentraciones de glucosa (*glu*) separado por las mujeres con y sin diabetes, del *Dataframe* *Pima.tr*:

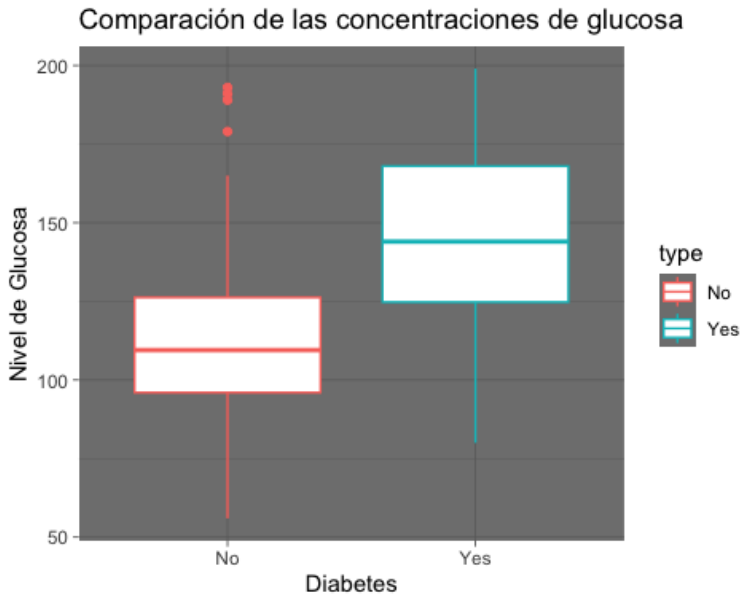
```
Pima.tr|>
  ggplot(mapping = aes(x= type, y= glu, fill = type))+
  geom_boxplot()+
  labs(title = "Comparación de las concentraciones de glucosa",
       x = "Diabetes", y = "Nivel de Glucosa") +
  theme_dark()
```



En el código anterior añadimos el argumento `fill = type` para `aes`, lo que nos permitió separar a las pacientes con y sin diabetes. El argumento `fill` en este caso colorea el interior de los boxplot, uno para las mujeres con diabetes y otro para las mujeres sin diabetes. Además, añadimos la geometría `geom_boxplot` y el tema `dark` mediante `theme_dark`.

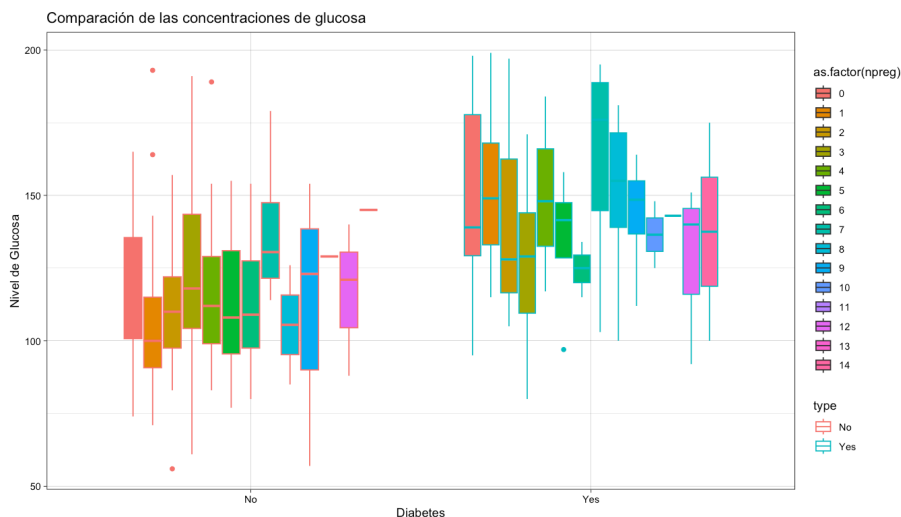
Si en el código anterior sustituimos el `fill = type` por `col = type` en lugar del interior, se colorea el contorno de las figuras:

```
Pima.tr|>
  ggplot(mapping = aes(x= type, y= glu, col = type))+
  geom_boxplot()+
  labs(title = "Comparación de las concentraciones de glucosa",
        x = "Diabetes", y = "Nivel de Glucosa") +
  theme_dark()
```



- ♦ **Ejemplo 10.25:** vamos a combinar dos argumentos en aes, lo cual nos permitirá tener dos agrupaciones, una para fill y otra para col:

```
Pima.tr|>
  ggplot(mapping = aes(x= type, y= glu, col = type, fill = as.factor(npreg)))+
  geom_boxplot()+
  labs(title = "Comparación de las concentraciones de glucosa",
        x = "Diabetes", y = "Nivel de Glucosa") +
  theme_linedraw()
```



Este código nos permitió separar tanto por la variable `type` como para el número de embarazos.

La última capa que nos falta por ejemplificar es la de `coordinate`, la cual permite modificaciones en los ejes, por ejemplo, que se muestre el logaritmo de las concentraciones de glucosa en lugar de las concentraciones originales. La capa de coordenadas en `ggplot2`, manejada por funciones como `coord_cartesian()`, `coord_fixed()`, `coord_flip()`, `coord_polar()`, entre otras, se utiliza para ajustar la escala y los límites del área de dibujo, y para cambiar la forma en que los datos se proyectan en el gráfico. En este libro no se revisará esta capa, ya que no se abordarán modificaciones sobre los ejes (salvo para la creación de gráficos de sectores).

Material complementario de ggplot2

Aquí solo se revisaron conceptos básicos y superficiales de `ggplot2`. Con ellos pretendemos que los lectores tengan una noción y puedan construir las gráficas que se necesitarán en este libro. Revisar a profundidad `ggplot2` podría requerir estudiar un libro completo, como el escrito por Wickham, Navarro y Lin Perderse (2016). Recomendamos algunos tutoriales y blogs que pueden ayudarle a comprender más sobre `ggplot2`:

- ◇ <https://cedricscherer.netlify.app/2019/08/05/a-ggplot2-tutorial-for-beautiful-plotting-in-r/>
- ◇ https://r-charts.com/ggplot2/#google_vignette
- ◇ <https://github.com/jennybc/ggplot2-tutorial>
- ◇ <https://rpubs.com/anlope10/562981>
- ◇ https://www.youtube.com/watch?v=HPJn1CMvtmI&list=PLtL57Fd-bwb_C6RSoJtBojTNOMVlgpeJkS

Por último, le recomendamos tener siempre a la mano el cheat sheet de ggplot:

- ◇ [Versión en español](#)
- ◇ [Versión original](#)

Ejemplos de sintaxis en ggplot2

- ◇ **Scatter plots:** Utilizados para mostrar la relación entre dos variables cuantitativas:

```
ggplot(data, aes(x = variable1, y = variable2)) +  
  geom_point()
```

- ◇ **Gráficos de líneas (line plots):** útiles para mostrar tendencias a lo largo del tiempo o de otra variable continua:

```
ggplot(data, aes(x = variable1, y = variable2)) +  
  geom_line()
```

- ◇ **Gráficos de barras (bar plots):** pueden ser utilizados tanto para datos categóricos como para datos continuos:

```
ggplot(data, aes(x = variable_factor, y = variable_numerica)) +  
  geom_bar(stat = "identity")
```

- ♦ **Histogramas:** para visualizar la distribución de una variable continua:

```
ggplot(data, aes(x = variable_numerica)) +  
  geom_histogram()
```

- ♦ **Box plots:** para mostrar la distribución de una variable continua a través de sus cuartiles y posibles valores atípicos:

```
ggplot(data, aes(x = variable_factor, y = variable_numerica)) +  
  geom_boxplot()
```

- ♦ **Diagramas de violín:** combinan características de los diagramas de caja y los histogramas:

```
ggplot(data, aes(x = variable_factor, y = variable_numerica)) +  
  geom_violin()
```

- ♦ **Diagramas de densidad:** para estimar la densidad de probabilidad de una variable continua:

```
ggplot(data, aes(x = variable_numerica)) +  
  geom_density()
```

- ♦ **Diagramas de área:** similares a los gráficos de líneas, pero con el área debajo de la línea rellena:

```
ggplot(data, aes(x = variable1, y = variable2)) +  
  geom_area()
```

- ♦ **Heatmaps:** para mostrar la intensidad de valores en una matriz:

```
ggplot(data, aes(x = variable1, y = variable2,  
                 fill = intensity_variable)) +  
  geom_tile()
```

- ♦ **Gráficos de facetas (faceted plots):** permiten dividir un gráfico en múltiples paneles, según los niveles de una variable:

```
ggplot(data, aes(x = variable1, y = variable2)) +
  geom_point() +
  facet_wrap(~ variable_factor)
```

- ♦ **Gráficos de burbujas (bubble plots):** una extensión de los diagramas de dispersión donde el tamaño de los puntos refleja una tercera variable:

```
ggplot(data, aes(x = variable1, y = variable2,
                 size = variable3)) +
  geom_point()
```

- ♦ **Gráficos de barras apiladas (stacked bar plots):** para mostrar la composición de cada categoría:

```
ggplot(data, aes(x = variable_factor,
                 y = variable_numerica,
                 fill = variable_agrupacion)) +
  geom_bar(stat = "identity", position = "stack")
```

Ejercicios

- ♦ **Ejercicio 10.1:** utilizando la base de datos `base de datos birtwht` de la librería `MASS` realice un histograma con 10 intervalos de clase para la variable `age`. Cambie el color de las barras, agregue un título y rótulos en los ejes.

Resolución [ejercicio 10.1](#)

```
# Si lo requiere
library(MASS)
data("birthwt")
hist(birthwt$age, breaks = 10,
```

```
main = "Histograma para la variable edad",
col = 1:10, xlab = "Intervalos de clase",
ylab = "Frecuencia")
```

- ♦ **Ejercicio 10.2:** realice una gráfico de densidad para la variable bwt. Agregue un título y rótulos en los ejes.

Resolución ejercicio 10.2

```
density(birthwt$bwt)|>
plot(main = "Gráfico de densidad para la variable bwt", col="cyan4",
      ylab = "Densidad", xlab = "bwt")
```

- ♦ **Ejercicio 10.3:** repita los ejercicios anteriores, pero esta vez utilizando ggplot2.

Resolución ejercicio 10.3

```
# Si es necesario
library(ggplot2)
birthwt|>
  ggplot(mapping = aes(x=age))+
  geom_histogram(bins = 10, fill = 1:10)+
  labs(title = "Histograma para la variable age",
        x = "Conteos",
        y = "Intervalos de clase")+
  theme_dark()
birthwt|>
  ggplot(mapping = aes(x=age))+
  geom_density()+
  labs(title = "Gráfico de densidad para la variable age")+
  theme_dark()
```

- ♦ **Ejercicio 10.4:** utilizando ggplot2 genere dos gráficos de densidad para la variable bwt (peso de los recién nacidos en g) de la base de datos birthwt, separando a la madres que fumaron y las que no fumaron en el embarazo

(variable `smoke`). Uno de los gráficos deberá de hacer la división utilizando `aes` y otro mediante la capa `faceta`.

Resolución [ejercicio 10.4](#)

```
birthwt$smoke ← factor(birthwt$smoke,
                        labels = c("No fumó", "Si fumó"))

birthwt|>
  ggplot(mapping = aes(x=bwt, col=smoke, fill = smoke))+
  geom_density(alpha=0.5)+ # 1= sin transparencia
  labs(title = "Gráfico de densidad de las madres
             que fumaron vs las madres que no fumaron",
        x = "Peso de los recién nacidos en g")+
  theme_minimal()

birthwt|>
  ggplot(mapping = aes(x=bwt, fill=smoke))+
  geom_density(alpha=0.5)+ # 1= sin transparencia
  labs(title = "Gráfico de densidad de las madres
             que fumaron vs las madres que no fumaron",
        x = "Peso de los recién nacidos en g")+
  facet_grid(~smoke)+
  theme_minimal()
```

- ♦ **Ejercicio 10.5:** realice un boxplot para la variable `glu` agrupada por la variable `type`. Utilice la base `Pima.tr`. Cambie colores y agregue títulos.

Resolución [ejercicio 10.5](#)

```
library(MASS)
data("Pima.tr")
boxplot(Pima.tr$glu~Pima.tr$type,
        main="Comparación de glucosa agrupo por la variable type",
        col = c("#999999", "#E69F00"), xlab = "Grupos de estudio",
        ylab = "Glucosa mg/dL")
```

- ♦ **Ejercicio 10.6:** realice un boxplot para la variable `skin` agrupada por la variable `type`. Utilice la base `Pima.tr`. Cambie colores y agregue títulos:
Resolución [ejercicio 10.6](#)

```
boxplot(Pima.tr$skin~Pima.tr$type,
        main="Comparación de skin agrupado por la variable type",
        col = c("#999999", "#E69F00"),
        xlab = "Grupos de estudio", ylab = "skin")
```

- ♦ **Ejercicio 10.7:** utilizando la base `birthwt` realice un boxplot en el que agrupe la edad de la madre (`age`) de acuerdo con el antecedente de hipertensión (`ht`) y el antecedente de tabaquismo de la madre (`smoke`).
Resolución [ejercicio 10.7](#)

```
boxplot(birthwt$age~birthwt$ht*birthwt$smoke,
        main="Comparación de la edad agrupad por ht y smoke",
        col = c("#999999", "#E69F00"),
        xlab = "Grupos de estudio", ylab = "Edad",
        names=c("Hipertensión \n no smoke",
                 "Hipertensión \n smoke",
                 "No hipertensión \n no smoke",
                 "No hipertensión \n smoke"))
```

- ♦ **Ejercicio 10.8:** utilizando la base de datos `Vacunacion` seleccione las primeras 50 filas para realizar un boxplot del total de pacientes `total_vaccinations` vacunados agrupado por el tipo de vacuna `vaccine`. ¿Tiene sentido hacer esta gráfica?
Resolución [ejercicio 10.8](#)

```
Vacunacion <- readxl::read_excel("Bases/Vacunacion.xlsx")
boxplot(total_vaccinations~vaccine, data = Vacunacion[1:50,])
```

- ♦ **Ejercicio 10.9:** utilizando la base `birthwt` realice un `boxplot` en el que agrupe el peso de los recién nacidos (`bwt`) de acuerdo con el antecedente de hipertensión (`ht`) y el antecedente de tabaquismo de la madre (`smoke`). Esta vez realice el ejercicio utilizando `ggplot2`, cambie títulos y rótulos de los ejes.

Resolución ejercicio 10.9

```
birthwt$smoke ← factor(birthwt$smoke,
                      labels = c("No fumó", "Si fumó"))
birthwt$ht ← factor(birthwt$ht,
                   labels = c("Sin HTA", "HTA"))
birthwt|>
  ggplot(mapping = aes(x=smoke, y=bwt, fill=ht))+
  geom_boxplot()+
  labs(title = "Comparación de la edad agrupado por ht y smoke",
       x = "Antecedente de tabaquismo",
       y = "Peso del recién nacido en g")+
  theme_minimal()
```

- ♦ **Ejercicio 10.10:** realice un gráfico de violín para la variable `glu` agrupada por la variable `type`. Utilice la base `Pima.tr`. Cambie colores y agregue títulos, utilice R base y luego `ggplot2`.

Resolución ejercicio 10.10

```
library(MASS)
data("Pima.tr")
vioplot::vioplot(Pima.tr$glu~Pima.tr$type,
                 main="Comparación de glucosa agrupo por la variable type",
                 col = c("#999999", "#E69F00"),
                 xlab = "Grupos de estudio", ylab = "Glucosa mg/dL")
```

El símbolo `::` permite acceder a las funciones de una librería sin necesidad de cargarla en el ambiente de trabajo. Para este ejemplo, accedimos a la función `vioplot()` de la librería `vioplot`.

```
Pima.tr|>
  ggplot(mapping = aes(x=type, y=glu))+
  geom_violin(aes(col=type, fill=type, alpha = 0.5))+
  geom_boxplot(aes(fill=type))+
  theme_minimal()+
  labs(title = "Comparación de glucosa agrupo por la variable type",
        x = "Grupos",
        y = "Glucosa (mg/dL)")
```

- ♦ **Ejercicio 10.11:** realice un gráfico de violín para la variable skin agrupada por la variable type. Utilice la base Pima.tr. Cambie colores y agregue títulos, utilice R base.

Resolución ejercicio ejercicio 10.11

```
vioplot::vioplot(Pima.tr$skin~Pima.tr$type,
  main="Comparación de skin agrupo por la variable type",
  col = c("#999999", "#E69F00"),
  xlab = "Grupos de estudio", ylab = "skin")
```

- ♦ **Ejercicio 10.12:** utilizando la base birthwt realice un gráfico de violín en el que agrupe la edad de la madre (age) de acuerdo con el antecedente de hipertensión (ht) y el antecedente de tabaquismo de la madre (smoke).

Resolución ejercicio 10.12

```
vioplot::vioplot(birthwt$age~birthwt$ht*birthwt$smoke,
  main="Comparación de la edad agrupad por ht y smoke",
  col = c("#999999", "#E69F00"), xlab = "Grupos de estudio", ylab = "Edad",
  names=c("Hipertensión \n no smoke", "Hipertensión \n smoke",
    "No hipertensión \n no smoke", "No hipertensión \n smoke"))
```

- ♦ **Ejercicio 10.13:** utilizando la base de datos Cushings de la librería MASS genere un gráfico de barras y otro de sectores para la variable Type. Cambie los colores, títulos de los ejes y el gráfico.

Resolución ejercicio 10.13

```
library(MASS)
data(Cushings)
# Crear paleta de colores
colores <- c("#fd7f6f", "#7eb0d5", "#b2e061", "#bd7
be", "#ffb55a", "#ffee65", "#beb9db", "#fdcce5", "#8bd3c7")

table(Cushings$Type)→tab # Objeto con los conteos
barplot(tab, col = colores,
        ylab = "Frecuencia",
        xlab = "Tipo de Cushing",
        main = "Frecuencia de los tipos de Cushings")
pie(tab,
    col = colores,
    main = "Frecuencia de los tipos de Cushings")
```

- ♦ **Ejercicio 10.14:** repita el ejercicio anterior pero esta vez utilizando ggplot2.

Resolución ejercicio 10.14

```
# Si es necesario
library(MASS)
data("Cushings")
library(ggplot2)
Cushings|>
  ggplot(mapping = aes(x=Type, fill = Type))+
  geom_bar()+
  labs(title = "Gráfico de barras para el tipo de Cushing",
       x = "Conteos",
       y = "Tipo de Cushing")+
  theme_bw()
```

La creación de un gráfico de sectores utilizando `ggplot2` es un tanto laboriosa, lo primero que se tiene que realizar es tener un *Dataframe* con los conteos de tipo de Cushing en una variable y otra con el tipo de Cushing.

```
Tabla_conteos <- data.frame(
  Tipo= c("a", "b", "c", "u"),
  Conteo= c(6,10,5,6)
)

Tabla_conteos|>
  ggplot(mapping = aes(x="", y=Conteo, fill = Tipo))+
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0)+
  labs(title = "Gráfico de sectores para el tipo de Cushing",
       x = "Conteos",
       y = "Tipo de Cushing")+
  theme_bw()
```

- ♦ **Ejercicio 10.15:** utilizando la base de datos `birthwt` de la librería `MASS` genere un gráfico de barras y otro de sectores para la variable `race`.

Resolución ejercicio 10.15

```
data("birthwt")
table(birthwt$race)→tab1
barplot(tab1, col=colores,
        xlab="Étnia",
        ylab="Frecuencias",
        main="Étnia de las pacientes")
pie(tab1, col = colores)
```

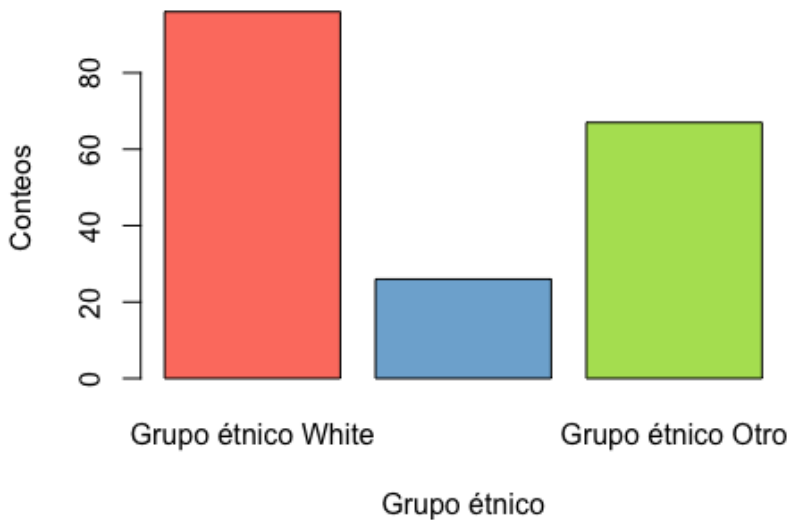
- ♦ **Ejercicio 10.16:** repita la gráfica de barras del ejercicio anterior, utilizando la función `plot()` y luego mediante `ggplot2`.

Resolución ejercicio 10.16

```
# Si es necesario
library(MASS)
data("birthwt")
library(ggplot2)
birthwt$race ← factor(birthwt$race,
                      labels = c("Grupo étnico White",
                                "Grupo étnico Black",
                                "Grupo étnico Otro"))

colores ← c("#fd7f6f", "#7eb0d5", "#b2e061", "#bd7ebe",
            "#ffb55a", "#ffee65", "#beb9db", "#fdcce5",
            "#8bd3c7")

plot(birthwt$race,
     col=colores,
     xlab="Grupo étnico",
     ylab= "Conteos")
```



```
birthwt|>
  ggplot(mapping = aes(x = race, fill = race))+
  geom_bar()+
  theme_classic()+
  labs(title = "",
        x = "Grupo étnico",
        y = "Conteo")
```

- ♦ **Ejercicio 10.17:** utilizando la base de datos `birthwt` de la librería `birthwt` genere un gráfico de mosaico con las proporciones de las variables `low` y `ht`. Realice una interpretación de su gráfico.

Resolución ejercicio 10.17

```
birthwt$low <- factor(birthwt$low,
                      labels = c("Peso normal", "Bajo peso"))
birthwt$ht <- factor(birthwt$ht,
                     labels = c("No HTA", "Si HTA"))

table(birthwt$low, birthwt$ht)|>
  mosaicplot(col=colores,
             main="Gráfico de mosaicos para bajo peso e HTA")
```

- ♦ **Ejercicio 10.18:** se tiene un conjunto de datos que incluye el tipo de tratamiento recibido por pacientes con una enfermedad específica: Tratamiento A, Tratamiento B o Tratamiento C.
 - a) Cree un vector que represente el tipo de tratamiento recibido por 150 pacientes, asegurándose de que los tratamientos estén distribuidos de manera no uniforme. Siembre la semilla: 123.
 - b) Utilice la función `table` para obtener la frecuencia de cada tipo de tratamiento.
 - c) Emplee `barplot` para crear un gráfico de barras que muestre visualmente estas frecuencias. Cambie los títulos de los ejes y el título del gráfico.

Resolución ejercicio 10.18

```
set.seed(123)
tratamientos <- sample(x=c("Tratamiento A",
                           "Tratamiento B",
                           "Tratamiento C"),
                      size = 150,
                      replace = T)

table(tratamientos)
barplot(table(tratamientos), col=colores,
        xlab="Tratamientos",
        ylab="Frecuencias",
        main="Tratamientos de una enfermedad")
```

- ♦ **Ejercicio 10.19:** ¿qué necesita para poder recrear esta gráfica en `ggplot2`? Cree la gráfica del ejercicio anterior en `ggplot2`.

Resolución ejercicio 10.19

Puede crear el gráfico directamente:

```
ggplot(mapping = aes(x=tratamientos, fill = tratamientos))+
  geom_bar()+
  labs(title = "Tratamientos de una enfermedad",
       x = "Tratamientos",
       y="Frecuencias")+
  theme_light()
```

- ♦ **Ejercicio 10.20:** cree un gráfico de dispersión que muestre la relación entre el índice de masa corporal (`bmi`) y los niveles de glucosa (`glu`) en la base de datos `Pima.tr`. Añadir una línea de tendencia con su intervalo de confianza.
- Cargue las librerías necesarias (`ggplot2` y `MASS`).
 - Utilice el conjunto de datos `Pima.tr`.
 - Cree un gráfico de dispersión de `bmi` vs. `glu`.
 - Añada una línea de tendencia con método de regresión lineal (`lm`).
 - Incluya etiquetas en los ejes y un título.

Resolución ejercicio 10.20

```
# Si es necesario
library(ggplot2)
library(MASS)

# Crear el gráfico de dispersión con línea de tendencia
ggplot(Pima.tr, aes(x = bmi, y = glu)) +
  geom_point() + # Agregar puntos
  geom_smooth(method = "lm", se = TRUE,
              color = "blue") + # Línea de tendencia con intervalo de confianza
  labs(title = "Relación entre BMI y Glucosa",
        x = "Índice de Masa Corporal (BMI)",
        y = "Nivel de Glucosa") + # Añadir etiquetas
  theme_minimal() # Aplicar tema minimalista
```

- ♦ **Ejercicio 10.21:** cree un gráfico de barras que muestre la distribución del número de personas con diabetes (type) en función de la clase de peso (bmi). Utilice la capa faceta para agrupar por la variable type.

Resolución ejercicio 10.21

```
# Si es necesario
library(ggplot2)
library(MASS)

# Crear el gráfico de barras facetado por 'type'
ggplot(Pima.tr, aes(x = bmi)) +
  geom_bar(fill = "skyblue") + # Gráfico de barras con color de relleno
  facet_wrap(~ type) + # Facetar por la variable 'type'

  labs(title = "Distribución de BMI por Tipo de Diabetes",
        x = "Índice de Masa Corporal (BMI)", y = "Frecuencia") +
  # Añadir etiquetas
  theme_minimal() # Aplicar tema minimalista
```

- ♦ **Ejercicio 10.22:** cree un boxplot que muestre la distribución de la presión arterial diastólica (bp) en función del tipo de diabetes (type). Aplicar un tema y personalizar el gráfico.
- Cargue las librerías necesarias (ggplot2 y MASS).
 - Utilice el conjunto de datos Pima.tr.
 - Cree un boxplot de bp en función de type.
 - Personalice el gráfico aplicando un tema y añadiendo etiquetas en los ejes y un título.

Resolución ejercicio 10.22

```
# Si es necesario
library(ggplot2)
library(MASS)

# Crear el boxplot de 'bp.diast' en función de 'type'
ggplot(Pima.tr, aes(x = type, y = bp, fill = type)) +
  geom_boxplot() + # Crear boxplot
  labs(title = "Distribución de la Presión Arterial Diastólica por
Tipo de Diabetes",
       x = "Tipo de Diabetes", y = "Presión Arterial Diastólica (bp)") +
  # Añadir etiquetas
  theme_minimal() + # Aplicar tema minimalista
  theme(legend.position = "none") # Eliminar la leyenda
```

Referencias

- Chan, B. K. C. (2018). Data Analysis Using R Programming. En *Biostatistics for Human Genetic Epidemiology* (pp. 47-122). Springer International Publishing. https://doi.org/10.1007/978-3-319-93791-5_2
- Coleman, A., Bose, A. y Mitra, S. (2023). Metagenomics Data Visualization Using R. *Methods in Molecular Biology*, 2649, 359-392. https://doi.org/10.1007/978-1-0716-3072-3_19

- Jeong, H.-W., Hernández-Rodríguez, B., Kim, J., Kim, K.-P., Enriquez-Gasca, R., Yoon, J., Adams, S., Schöler, H. R., Vaquerizas, J. M. y Adams, R. H. (2017). Transcriptional regulation of endothelial cell behavior during sprouting angiogenesis. *Nature Communications*, 8(1), 726. <https://doi.org/10.1038/s41467-017-00738-7>
- Kassambara, A. (2023). *ggpubr: «ggplot2» Based Publication Ready Plots*. <https://CRAN.R-project.org/package=ggpubr>
- Sievert, C. (2020). *Interactive web-based data visualization with R, plotly, and shiny*. Chapman and Hall/CRC. <https://plotly-r.com/>
- Wickham, H., Çetinkaya-Rundel, M. y Grolemund, G. (2018). *R for Data Science: Import, Tidy, Transform, Visualize, and Model*. <https://r4ds.hadley.nz/>
- Wickham, H., Navarro, D. y Lin Perderse, T. (2016). *ggplot2: Elegant Graphics for Data Analysis* (3.a ed.). <https://ggplot2-book.org/>

Estadística descriptiva

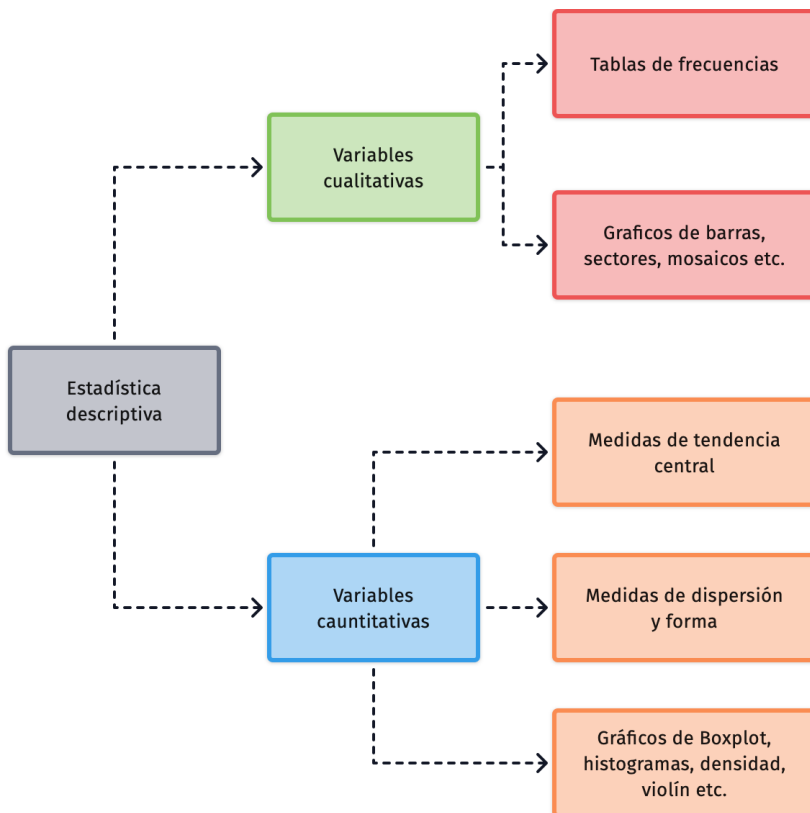
Ana Gabriela Flores Vargas
Miryam Rosario Guillén Medina
Edsaúl Emilio Pérez Guerrero

Introducción

La estadística descriptiva tiene por misión resumir la información que proporciona un conjunto de datos y de las variables. Consiste en presentar y resumir los datos, para ello, empleamos gráficos, tablas, estimación de estadísticos. En este libro incluimos a la estadística descriptiva dentro de la exploración de datos ([figura 3.2](#), punto 4).

En este capítulo primeramente abordaremos la estadística descriptiva para variables cualitativas, donde revisaremos tablas de frecuencia y gráficas que permitan dar a conocer y presentar este tipo de variables. Posteriormente, nos centraremos en la descripción y presentación de las variables cuantitativas en donde abordaremos las medidas de tendencia central, dispersión y de forma, además de los gráficos más representativos para estas variables. La [figura 11.1](#) muestra una visión general sobre la estadística descriptiva y sus principales medidas y gráficos agrupados por tipo de variables.

FIGURA 11.1. VISIÓN GENERAL DE LA ESTADÍSTICA DESCRIPTIVA



Estadística descriptiva para variables cualitativas

La estadística descriptiva en variables cualitativas es extremadamente sencilla. Básicamente consiste en presentar los datos en forma de conteos o frecuencias y en proporciones o porcentajes. Las frecuencias proporcionan una rápida visión del comportamiento de los datos.

La [tabla 11.1](#) muestra un ejemplo de la descripción de variables cualitativas en un grupo de pacientes en los que les fue evaluado el dolor después de una cirugía por apendicitis.

TABLA 11.1. EJEMPLO DE TABULACIÓN (ESTADÍSTICA DESCRIPTIVA) DE VARIABLES CUALITATIVAS

CATEGORÍA DEL DOLOR	FRECUENCIA	FRECUENCIA RELATIVA	FRECUENCIA ACUMULADA	FRECUENCIA RELATIVA ACUMULADA
Severo	4	0.70	60	1.00
Moderado	8	0.13	56	0.93
Leve	17	0.28	48	0.80
Ninguno	31	0.52	31	0.52

Existen algunas medidas que podemos utilizar para presentar más ampliamente nuestros datos:

- ♦ Frecuencia (conteos)
- ♦ Frecuencia relativa (proporción)
- ♦ Frecuencia acumulada
- ♦ Frecuencia relativa

Describiremos cada una de estas medidas a continuación:

- ♦ **Frecuencia absoluta:** la frecuencia absoluta (f_a) es el número de veces que se ha observado dicha modalidad. Son los conteos que podemos hacer a las categorías de las variables cualitativas

$$\sum f_a = n$$

- ♦ **Frecuencia relativa (proporciones):** la frecuencia relativa (f_r) es la frecuencia absoluta dividida entre el número de casos. Se puede expresar como proporción o como porcentaje

$$\sum f_r = \frac{f_a}{n}$$

- ♦ **Frecuencia acumulada:** la frecuencia acumulada es el resultado que se obtiene de la suma sucesiva de las frecuencias absolutas o relativas, cuando se realiza de menor a mayor, según sus valores. En la frecuencia acumulada, los datos se pueden ir sumando en orden ascendente o descendente.

- ♦ **Ejemplo 11.1:** en un servicio de traumatología, con objeto de realizar una correcta planificación, interesa conocer la localización de la patología principal de los pacientes atendidos en urgencia, para lo cual se estudia una muestra de 186 elegida entre los pacientes atendidos durante los últimos 6 meses.

La variable de interés es la zona afectada. Es una variable cualitativa dividida en 5 modalidades: rodilla, cadera, tobillo, cráneo y otras. Al final del estudio se obtuvo que en los 6 meses se atendieron 30 problemas relacionados con rodilla, 28 con cadera, 41 con tobillo, 34 cráneo y 53 otras. La [tabla 11.2](#) muestra cómo deberían de presentarse los datos. Para poder llegar a ellos, simplemente tiene que aplicar cada una de las formulas descritas anteriormente. En el caso particular de esta variable, a menos de que usted lo establezca, no existe un orden específico para las cirugías, por lo tanto, las frecuencias acumuladas pueden variar. De hecho, si en su variable no existe un orden, las frecuencias acumuladas quizá no le digan mucho.

TABLA 11.2. RESOLUCIÓN EJEMPLO, ESTADÍSTICA DESCRIPTIVA EN VARIABLES CUALITATIVAS

ZONA AFECTADA	FA	FR(N)	FR(%)	FA ACUMULADA
Rodilla	30	0.161	16.1	30
Cadera	28	0.151	15.1	58
Tobillo	41	0.220	22.0	99
Cráneo	34	0.183	18.3	133
Otras	53	0.285	28.5	196
Total	186	1.000	100.0	–

¿Cómo obtener una tabla de frecuencias con R?

En R se utilizan, las siguientes funciones:

1. `table()` permite conocer las frecuencias de un objeto, incluso si los tipos de datos no son cualitativos.

2. `prop.table()` permite obtener las proporciones de un objeto. Es necesario que el objeto sea una tabla.
3. `cumsum()` realiza la suma acumulativa de las frecuencias de un objeto.

No olvide que puede consultar la ayuda de cada una de las funciones utilizando la función `help()`.

- ♦ **Ejemplo 11.2:** para ejemplificar cómo estimar los valores de una tabla de frecuencias en R, primero partiremos de un objeto llamado `grados` y después estimaremos cada uno de los valores:

```
set.seed(346597) # Sembrando la semilla
grados <- sample(LETTERS[1:4], 1000, TRUE)
```

Con el código anterior creamos un objeto con 1000 datos de las letras de la A a la D con remplazo. Ahora podemos visualizar y contar cuantos grados (letras) hay:

```
table(grados) # Permite visualizar el conteo de los grados
grados
```

A	B	C	D
268	252	233	247

Ahora, podemos ver la proporción de cada uno de los grados con la función `prop.table()`:

```
# Se requiere un objeto que sea una tabla
table(grados)|>
  prop.table() # Permite conocer las proporciones de cada uno de los grados
grados
```

A	B	C	D
0.268	0.252	0.233	0.247

Para obtener los porcentajes, podemos multiplicar nuestra tabla con las proporciones por 100:

```
prop.table(table(grados))*100# Se multiplica por 100
```

grados

A	B	C	D
26.8	25.2	23.3	24.7

Para obtener las frecuencias acumuladas podemos utilizar la función `cumsum`:

```
cumsum(table(grados))
```

A	B	C	D
268	520	753	1000

- ♦ **Ejemplo 11.3:** si se parte de la creación de un objeto a partir de una tabla, puede utilizar el siguiente código:

```
x <- table(grados)
```

```
table(x) # Conteos
```

x

```
233 247 252 268
```

```
1 1 1 1
```

```
prop.table(x) # Proporciones, Se requiere un objeto que sea una tabla
```

grados

A	B	C	D
0.268	0.252	0.233	0.247

```
prop.table(x)*100 # Porcentajes
```

grados

A	B	C	D
26.8	25.2	23.3	24.7

```
cumsum(x) # Frecuencias acumuladas
```

A	B	C	D
268	520	753	1000

- ♦ **Ejemplo 11.4:** puede crear varios objetos y posteriormente organizarlos en forma de tabla para obtener de una forma más organizada, en este ejemplo, se presentará primero el código y después se explicará más adelante:

```
data_freq ←
  data.frame(
    Freq = as.numeric(x),
    Freq_Rela = as.numeric(prop.table(x)),
    Freq_Acum = as.numeric(cumsum(x)),
    Freq_Rela_Acum = as.numeric(cumsum(prop.table(x)))
  )
rownames(data_freq) ← LETTERS[1:4]
data_freq
```

	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum
A	268	0.268	268	0.268
B	252	0.252	520	0.520
C	233	0.233	753	0.753
D	247	0.247	1000	1.000

Explicación del código anterior: el código de forma general, permitió la creación de un objeto `data_freq()` que contiene una tabla de frecuencias con los conteos (objeto `Freq`), la frecuencia relativa (objeto `Freq_Rela`), la frecuencia acumulada (objeto `Freq_Acum`) y la frecuencia relativa acumulada (objeto `Freq_Rela_Acum`). Este objeto tomó los datos del objeto llamado “x” que creamos en el [ejemplo 11.3](#). Para que los datos pudieran ser organizados de forma adecuada en una tabla, es necesario hacer la coerción a valores numéricos de los objetos.

En el código se utilizó la función `data.frame()`, la cual permite crear marcos de datos cuando es alimentada por objetos y los organiza en columnas (puede revisar el siguiente [enlace](#)). Finalmente la función `rownames()` permite cambiar los nombres del *Dataframe*, en este caso por las primeras cuatro letras del abecedario (igual que los grados).

Note cómo en este código no ordena explícitamente los datos, pero al estar utilizando las primeras cuatro letras del abecedario, los datos se ordenan mediante el alfabeto. Si quiere ordenar los datos, deberá utilizar la función `sort()`.

La tabla de frecuencias también puede crearse con el siguiente código, en el que creo una nueva columna con los nombres de la tabla `Nombres=names(x)`. Recuerde que puede copiar y pegar el código en su consola o script:

```
data_freq <-
data.frame(
  Nombres= names(x),
  Freq =as.numeric(x),
  Freq_Rela = as.numeric(prop.table(x)),
  Freq_Acum = as.numeric(cumsum(x)),
  Freq_Rela_Acum = as.numeric (cumsum(prop.table(x))))
print(data_freq)
```

Funciones para resumen de variables cualitativas

Si bien, en el apartado se describió cómo obtener una tabla de frecuencias utilizando el paquete base de R, existen algunas funciones más especializadas para obtener este tipo de información. Vamos a revisar la `freq()` de la librería `questionr` y la función `tabyl()` de la librería `janitor`.

- ♦ **Ejemplo 11.5:** para este ejemplo vamos a importar la base de datos `Base Descriptivos.rds` que puede descargar de [este enlace](#).

Este es un archivo `.rds` por lo que deberá de emplear la función `readRDS()`. Le recomendamos, sino lo ha hecho ya, crear dentro de su proyecto una carpeta llamada `Bases` y guardar ahí el *Dataframe* que descargó. Si siguió las indicaciones anteriores podrá emplear el siguiente código para importar la base de datos:

```
df <- readRDS(file = "Bases/Base Descriptivos.rds")
```

Del código anterior identifique los siguientes puntos:

- ♦ Se nombró el objeto con el *Dataframe* como `df`. `df` es una abreviatura de *Dataframe* que se utiliza comúnmente. La vamos a utilizar mucho en este libro.
- ♦ La función `readRDS()` se alimentó de un archivo que en este caso fue asignado a un archivo de nuestra computadora siguiendo la ruta “`Bases/Base Descriptivos.rds`”. Esta ruta debe de ir entrecomillada.

Una vez importada nuestra base de datos vamos a utilizar la función `freq()` de la librería `questionr` para hacer una tabla de frecuencia para la variable `Escolaridad`.

```
# Si es necesario instale la librería
install.packages("questionr")
```

Ahora llame la librería:

```
library(questionr)
```

Y aplique la función `freq()`:

```
freq(df$Escolaridad)
```

	n	%	val%
Licenciatura	50	14.3	15.0
Posgrado	62	17.7	18.6
Preparatoria	52	14.9	15.6
Primaria completa	45	12.9	13.5
Primaria incompleta	63	18.0	18.9
Secundaria	61	17.4	18.3
NA	17	4.9	NA

- ♦ **Ejemplo 11.6:** ahora vamos a utilizar a utilizar la función `tabyl()` de la librería `Janitor` para obtener nuestra tabla de frecuencia, aunque en realidad no obtenemos una tabla de frecuencias, pero sí una tabla muy estética para obtener frecuencias y porcentajes:

```
#Instale la librería de ser necesario
install.packages("janitor")
library(janitor)
tabyl(df$Escolaridad)
```

df\$Escolaridad	n	percent	valid_percent
Licenciatura	50	0.14285714	0.1501502
Posgrado	62	0.17714286	0.1861862
Preparatoria	52	0.14857143	0.1561562
Primaria completa	45	0.12857143	0.1351351
Primaria incompleta	63	0.18000000	0.1891892
Secundaria	61	0.17428571	0.1831832
<NA>	17	0.04857143	NA

Gráficos utilizados para la exploración de variables cualitativas

Existen diferentes gráficos que nos permiten representar las variables, entre ellos se incluyen:

- ♦ Gráficos de barras, función `barplot()` y `geom_bar`
- ♦ Gráficos de sectores, función `pie()`
- ♦ Gráficos de mosaicos, función `mosaicplot()`

La construcción y descripción de estos gráficos fue descrita en las secciones: “[Barplot](#)”, “[Mosaic plot \(gráfico de mosaicos\)](#)”, “[Gráficos de sectores](#)” y “[Material complementario de ggplot2](#)”.

La función `cut`

En ocasiones, es de nuestro interés fraccionar o segmentar una variable de cuantitativa para poder obtener categorías de la misma. De hecho, este proceso es común en la estadística descriptiva. Por ejemplo, supongamos que es de nuestro interés conocer cuántos de los pacientes de un estudio presentan una edad de 0 a 10 años, de 11 a 20 años y de 21 a 35 años. Tome en cuenta que el categorizar una variable numérica puede hacer que se pierda información y no en todas las situaciones es recomendable.

Para obtener esta variable fraccionada o “cortada” en R utilizamos la función `cut()` que permite precisamente categorizar una variable de tipo cuantitativo. Esta función tiene la siguiente estructura:

```
cut(x, breaks, labels = NULL,
    include.lowest = FALSE, right = TRUE, dig.lab = 3,
    ordered_result = FALSE, ...)
```

- ♦ `x`: vector numérico
- ♦ `breaks`: puntos de corte, debe ser mayor a 2
- ♦ `labels`: etiquetas de cada categoría
- ♦ `right`: indica hacia donde debe cerrarse el intervalo `]`. El valor predeterminado es el izquierdo `[`, es decir no incluirá el número del lado izquierdo pero si el del lado derecho.

Para resolver los siguientes ejemplos y entender cómo funciona `cut()`, vamos a crear un vector numérico de datos ficticios que contendrán las edades de 800 pacientes. El código para crear este vector es el siguiente:

```
# Creando un vector
x ← (1:40)
# Sembrando la misma semilla
set.seed(1234)
# Obteniendo una muestra
edad ← sample(x, size = 800, replace = T)
```

Del código anterior conviene explicar lo siguiente: 1) se creó un vector `x` con una secuencia de números del 1 al 40, 2) se utilizó la función `set.seed()` para sembrar la semilla. La función `set.seed()`. recuerde que R crea números aleatorios siguiendo un algoritmo, en este caso con la función `set.seed()` define bajo qué parámetros se mueve este algoritmo. Puede revisar más detalles de esta función en: <https://www.statology.org/set-seed-in-r/> y en <https://r-coder.com/set-seed-r/>. 3) Se creó un objeto llamado `edad`, utilizando el objeto `x` se creó una muestra con reemplazo de 800 datos.

Ahora sí, vamos a resolver un ejemplo.

- ♦ **Ejemplo 11.7:** utilizando el objeto `edad` que creó anteriormente realice cortes de manera que pueda identificar cuantos pacientes tienen una edad menor o igual a 10, de 11 a 30 años incluyendo el 30 y el resto de los pa-

cientes. No cambie las etiquetas de los niveles. Nombre este nuevo objeto como `edad1`, pida los conteos de los niveles utilizando la función `table()`, realice una gráfica de barras con esta nueva variable.

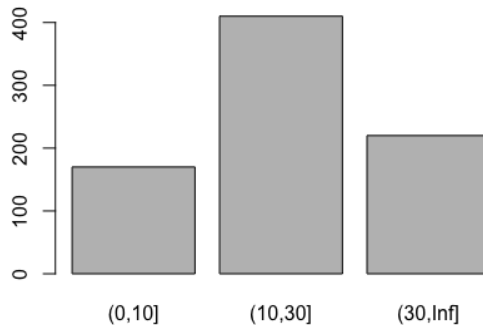
El código para resolver este ejercicio es el siguiente:

```
edad1 <- cut(x = edad, breaks = c(0, 10, 30, Inf),
             right=T)
table(edad1)
edad1
      (0,10]      (10,30]      (30,Inf]
      170       410       220
table(edad1==30)

FALSE
800
```

Se crearon 4 cortes del objeto `edad`, el primero va del 0 al 10 sin incluir el 0 pero incluyendo el 10, el segundo corte va del 10 al 30 sin incluir el 10 e incluyendo el 30, el último corte va del 30 (sin incluirlo) al infinito. Utilizar `Inf` simplifica la selección del número más grande y es muy recomendado cuando no se conoce el número mayor. Aunque puede construir una gráfica y visualizar su nueva variable:

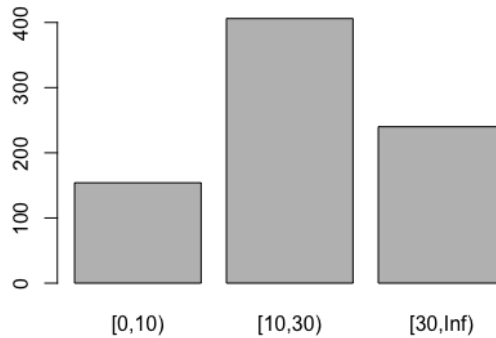
```
table(edad1)|>
  barplot()
```



- ♦ **Ejemplo 11.8:** cree un nuevo objeto llamado `edad2` que contenga los mismos cortes del ejercicio anterior, pero cambie el argumento `right = F`:

```
edad2 <- cut(x = edad, breaks = c(0, 10, 30, Inf),
             right=F)
table(edad2)
edad2
  [0,10)  [10,30) [30,Inf)
      154      406      240
table(edad2==30)

FALSE
      800
barplot(table(edad2))
```



- ♦ **Ejemplo 11.9:** utilizando la función `cut` cambie las etiquetas de los niveles `edad1` y `edad2` por: “Grupo edad 1”, “Grupo de edad 2”, “Grupo de edad 3”. Para ello utilice el argumento `labels`:

```
edad1 <- cut(x = edad, breaks = c(0, 10, 30, Inf),
             labels = c("Grupo edad 1", "Grupo de edad 2", "Grupo de edad 3" ),
             right=T)
table(edad1)
edad1
```

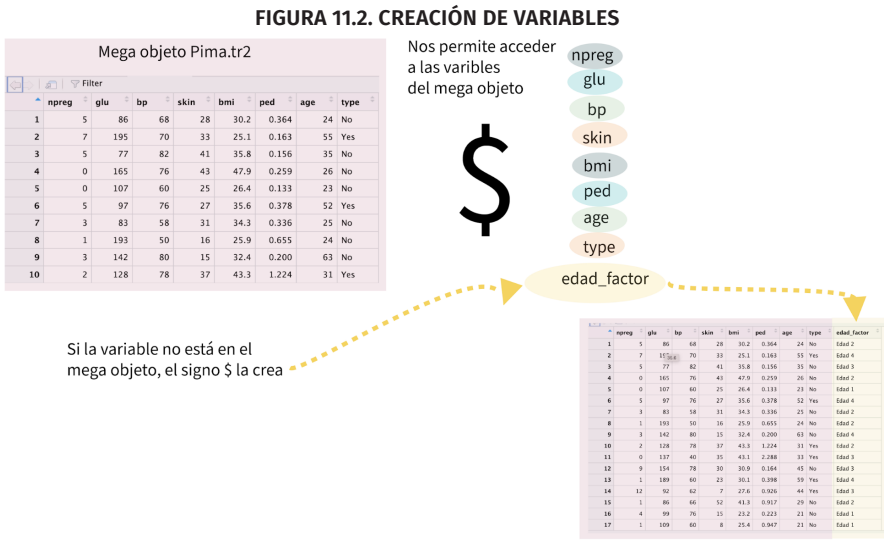
```

Grupo edad 1      Grupo de edad 2      Grupo de edad 3
170               410                  220

levels(edad1)# Visualizar los niveles de un factor
[1] "Grupo edad 1"      "Grupo de edad 2"      "Grupo de edad 3"
    
```

Creación de variables dentro de un Dataframe

Es común que sea de nuestro interés incluir una nueva variable que calculó o modificó en un *Dataframe*, por ejemplo, la variable que recientemente creamos utilizando la función `cut()`. Para ello, podemos utilizar el símbolo `$` que en el caso de que la variable no se encuentre dentro nuestro *Dataframe* (megaobjeto) la crea. Ver [figura 11.2](#).



Por ejemplo, el siguiente código estaría creando una nueva variable dentro del *Dataframe* estimada después de la realización de un “corte” de una variable:

```
base_datos$nueva_variable <- cut(variable_antigua, breaks=4)
```

♦ **Ejemplo 11.10:** utilizando la base de datos `Pima.tr2` de la librería `MASS` corte la variable `age` de acuerdo con los siguientes criterios:

- » Mujeres menores a 24 años
- » Mujeres de 24 a 33, pero menores de 33
- » Mujeres de 33 a 49
- » Mujeres de 50 a 72

Además: 1) cree un nuevo objeto dentro la base `Pima.tr2` y nómbrelo como `edad_factor`, 2) cambie las etiquetas de `edad_factor` por la etiqueta que más le convenga, 3) obtenga un conteo de los niveles del objeto `edad_factor`, 4) cree un gráfico de barras de `edad_factor`:

```
library(MASS) # Cargar librería
data("Pima.tr2") # Llamar base de datos
Pima.tr2$edad_factor <- cut(Pima.tr2$age,
                             breaks = c(0, 24, 33, 50, 73),
                             labels = c("Edad 1", "Edad 2",
                                           "Edad 3", "Edad 4"),
                             right = F) # Crear cortes
# El código anterior permitió crear una nueva variables dentro de
Pima.tr2
table(Pima.tr2$edad_factor) # Conteos de la nueva variable
```

Edad 1	Edad 2	Edad 3	Edad 4
65	120	79	36

Si utiliza el código `View(Pima.tr2)` podrá visualizar su nueva variable al final de la base de datos. La [figura 11.3](#) ejemplifica la creación de la variable `edad_factor` dentro de `Pima.tr2`.

FIGURA 11.3. CREACIÓN DE LA VARIABLE EDAD_FACTOR

	npreg	glu	bp	skin	bmi	ped	age	type	edad_factor
1	5	86	68	28	30.2	0.364	24	No	Edad 2
2	7	157	70	33	25.1	0.163	55	Yes	Edad 4
3	5	77	82	41	35.8	0.156	35	No	Edad 3
4	0	165	76	43	47.9	0.259	26	No	Edad 2
5	0	107	60	25	26.4	0.133	23	No	Edad 1
6	5	97	76	27	35.6	0.378	52	Yes	Edad 4
7	3	83	58	31	34.3	0.336	25	No	Edad 2
8	1	193	50	16	25.9	0.655	24	No	Edad 2
9	3	142	80	15	32.4	0.200	63	No	Edad 4
10	2	128	78	37	43.3	1.224	31	Yes	Edad 2
11	0	137	40	35	43.1	2.288	33	Yes	Edad 3
12	9	154	78	30	30.9	0.164	45	No	Edad 3
13	1	189	60	23	30.1	0.398	59	Yes	Edad 4
14	12	92	62	7	27.6	0.926	44	Yes	Edad 3
15	1	86	66	52	41.3	0.917	29	No	Edad 2
16	4	99	76	15	23.2	0.223	21	No	Edad 1
17	1	109	60	8	25.4	0.947	21	No	Edad 1

También, podemos comprobar que nuestros cortes hayan sido correctos. Puede utilizar el siguiente código:

```
# Comprobamos Edad 1
table(Pima.tr2$age<24)

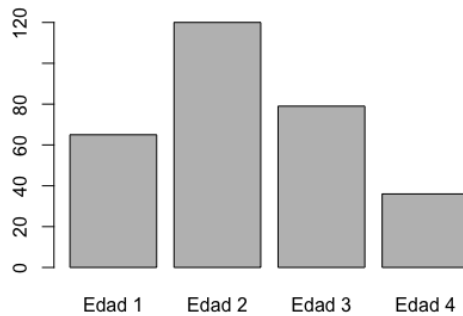
FALSE      TRUE
235         65

# Comprobamos Edad 2
table(Pima.tr2$age ≥ 24 & Pima.tr2$age < 33)

FALSE      TRUE
180         120
```

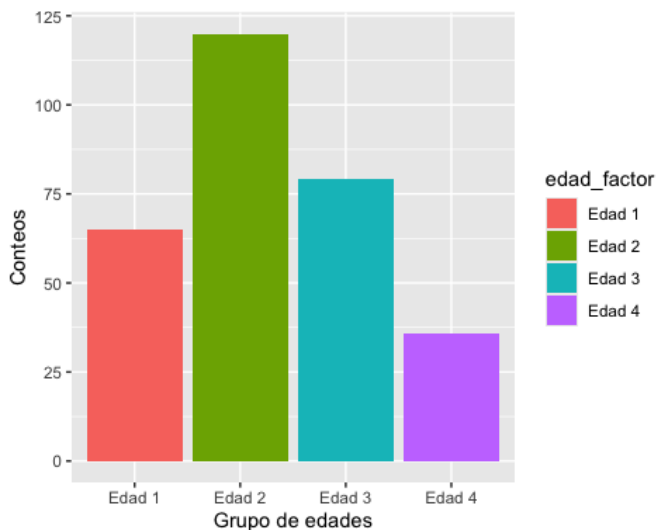
Finalmente, podemos crear una gráfica para visualizar nuestra nueva variable:

```
barplot(table(Pima.tr2$edad_factor))
```

Utilizando `ggplot2` la gráfica se construiría así:

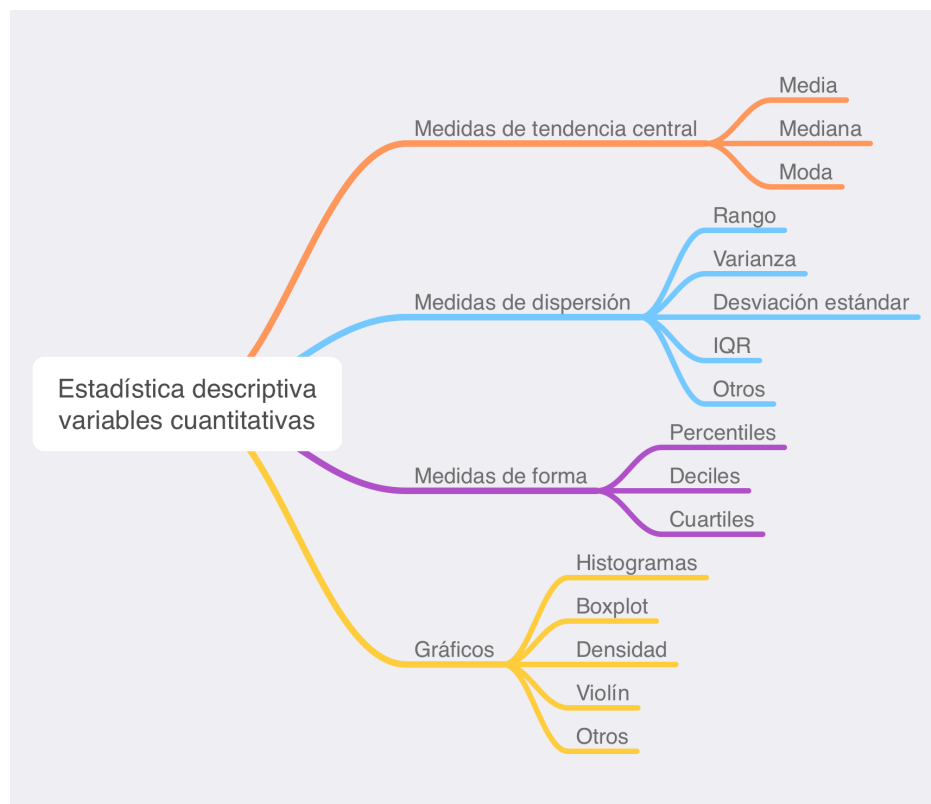
```
library(ggplot2)
Pima.tr2|>
  ggplot(mapping = aes(x=edad_factor, fill = edad_factor))+
  geom_bar()+
  labs("Gráfico de barras para la edades de un grupo de pacientes",
       x="Grupo de edades",
       y= "Conteos")
```



Estadística descriptiva para variables cuantitativas

La presentación, descripción y presentación se puede realizar mediante gráficas, medidas de tendencia central y medidas de dispersión. Además, se pueden incluir otras medidas que reflejan la posición de los datos cuando estos son ordenados. Comenzaremos la descripción de esta sección con las medidas de tendencia central, seguida de las medidas de dispersión para finalizar con las medidas de dispersión. La [figura 11.4](#) muestra una visión general sobre la estadística de descriptiva de las variables cuantitativas.

FIGURA 11.4. VISIÓN GENERAL ESTADÍSTICA DESCRIPTIVA PARA VARIABLES CUANTITATIVAS



Medidas de tendencia central

Las medidas de tendencia central son estadísticas que describen el centro de un conjunto de datos. Estas medidas resumen la posición central de un conjunto de datos, proporcionando un valor alrededor del cual los datos tienden a agruparse. Las principales medidas de tendencia central son: la media aritmética, la mediana y la moda.

Media aritmética

Se refiere al valor que tendría cada elemento de la serie de datos, si todos tuvieran el mismo valor. Presenta las siguientes características:

- ♦ Es el parámetro de centralización más utilizado, su valor es el centro aritmético de los datos.
- ♦ Se suele emplear el símbolo μ para la media poblacional.
- ♦ Se emplea $\bar{x}$ muestral.
- ♦ Es muy sensible a los valores extremos.
- ♦ La media aritmética se puede estimar de otras medias aritméticas.
- ♦ Única. Solo existe una media para un determinado conjunto de datos.
- ♦ Simple. Es fácil de calcular y de comprender.

La fórmula para estimar la media aritmética es la siguiente:

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n}$$

Donde: x_i es el i -ésimo dato; si la suma es desde i es igual a 1 hasta n .

La media aritmética presenta la desventaja de que es un valor muy susceptible a los valores extremos y que cuanto estos están presentes, la media puede no reflejar la realidad. Por ejemplo, la media para los siguientes datos (12, 14, 15, 14, 15, 18) la media es 14.6, en cambio, si modificamos el último dato por 38, la media cambia hasta 18. Para estimar la media en R utilice la función `mean()`.

Mediana

Otra medida de centralización es la mediana. Esta medida refleja el valor central de un conjunto con ordenados de mayor a menor. La mediana es el valor que divide al conjunto de datos en dos partes iguales. Su estimación es bastante sencilla, sin embargo, es distinta cuando se trabaja con n datos impares y cuando se trabaja con n datos pares.

Cuando se trabaja con una n impar la fórmula es la siguiente:

$$\Rightarrow M = \frac{X_{(n+1)/2}}$$

Si es par, la mediana es la media aritmética de los dos valores centrales:

$$\Rightarrow M = \frac{X_{n/2} + X_{n/2+1}}{2}$$

Por ejemplo, para el siguiente conjunto de datos: 2, 4, 6, 8, **9**, **11**, 12, 13, 14, 20, la mediana es 10. Si ordenamos los datos de menor a mayor, notaremos que, al tratarse de un número par, los dos valores del centro serán 9 y 11. Por tanto, 10 corresponde al promedio de **9** y **11**.

En cambio, si el número de datos del conjunto de valores es impar, simplemente debemos de seleccionar el valor del centro. En el siguiente conjunto de datos, la mediana también es 10. Note cómo se agregó un valor más.

Datos ordenados: 2, 4, 6, 8, 9, **10**, 12, 13, 14, 20. El valor central es el 10.

La mediana posee las siguientes características:

1. Única. Una mediana para cada conjunto de datos.
2. Simple. Es fácil de estimar.
3. Los valores extremos no influyen tanto sobre la mediana, ya que solo considera 1 o 2 valores.
4. Divide al grupo de valores en dos partes iguales, cada una con el 50% de las observaciones.
5. Desprecia información, solo toma en cuenta los valores del centro.
6. No es posible estimar la mediana de otros medianas. Se necesita la totalidad de los datos.

En R puede utilizar la función `median()` para estimar la mediana.

Moda

Otro valor de centralización es la moda. Este valor refleja al dato o valor que más se repite en un conjunto de datos. Es una medida que no es muy utilizada en ciencias de la salud, con excepción de algunos casos. Se considera como absoluta al valor que efectivamente es el que más se repite y como moda relativa al valor que se repite considerablemente sin llegar a ser el que más se reitera.

Algunos conjuntos de datos pueden presentar más de una moda. En el caso de que presenten dos, nos referiremos a ellos como bimodales. También se utiliza los términos trimodales, etcétera.

En el paquete base de R no existe ninguna función para estimar la moda, pero la función `table` le podría ayudar fácilmente a conocer cuál es el dato que más se repite en un conjunto de datos.

Medidas de dispersión

Estas medidas ofrecen información sobre el grado de variabilidad de una variable. Además, indican si una variable tiene datos más dispersos (variación) que otra. Junto con las medidas de tendencia central y dispersión son las medidas que se utilizan para presentar una variable cuantitativa.

Las medidas de dispersión más importantes son: rango, varianza, desviación estándar y coeficiente de variación.

Rango

- ♦ Es la diferencia entre el valor máximo y el valor mínimo de los datos observados.
- ♦ Aporta información sobre el recorrido de una variable.
- ♦ Puede ser engañosa con datos extremos.
- ♦ Nunca se debe evaluar solo, se necesitan de otras medidas de dispersión. Puede ser muy engañoso.

Suponga que es de su interés conocer qué tanto varían ciertas presiones arteriales de un grupo de pacientes. Las presiones sistólicas son: 120, 135, 160, 100, 155, 115, 165, 125, 130. Una manera de medir la variabilidad es mediante el rango, que se estima de la siguiente forma:

- ♦ Máximo = 165
- ♦ Mínimo = 100
- ♦ Rango = $165 - 100 = 65$

En R utilice la función `range()`.

Varianza y desviación estándar

Una buena medida de dispersión sería tomar como referencia qué tanto los valores se desvían de la media (centro). Por lo tanto, se podría restar la media a cada uno de los valores de nuestro conjunto de datos y ver qué tanto se alejan de esta. Sin embargo, esta acción tendría como consecuencia que algunos de los valores calculados sean negativos; y con ello, se anularía el efecto de los valores positivos con los negativos. Para lidiar con este problema, una vez que se haya restado la media a cada uno de los valores, se podrían elevar al cuadrado y, con ello, tener siempre valores positivos.

La varianza es la medida que estima el promedio de las diferencias cuadráticas de los datos respecto a la media. Sus unidades son las de los datos al cuadrado, por ejemplo, si medimos las estaturas en *cm* de un grupo de niños, la varianza tendrá unidades de *cm*². Para representar la varianza población se utiliza σ^2 , mientras que para la varianza muestral se utiliza s^2 .

La fórmula para estimar la varianza de una población es:

$$\sigma^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n}$$

Cuando se utiliza como estimador de población se utiliza $n - 1$:

$$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}$$

Una medida más fácil de manejar, dado que no implica datos al cuadrado, es la desviación estándar, la cual es la raíz cuadrada de la varianza. La desviación estándar poblacional está dada por:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{N}}$$

La desviación estándar indica qué tanto se desvían los datos en promedio con respecto a la media de ese conjunto de datos. A diferencia de la varianza, la desviación estándar no tiene unidades. Mientras mayor sea la desviación estándar, mayor será la dispersión de los datos.

No olvidar que:

1. La varianza y la desviación estándar son mayores o iguales a cero.
2. A mayor dispersión de los datos, mayor varianza y mayor desviación estándar.

En R utilice las funciones `var()` y `sd()` para estimar la varianza y la desviación estándar, respectivamente.

Coefficiente de variación

Es una medida de dispersión sin unidades que permite la comparación de variables tan diferentes, como edades e índice de masa corporal. El coeficiente de variación es el cociente de la desviación estándar respecto a la media, multiplicado por 100 $CV = \frac{s}{\bar{x}} \cdot 100$. La principal característica del coeficiente de variación es que permite la comparación de datos en distintas unidades y en distintas poblaciones o muestras.

No existe una función en el paquete base de R para la estimación del coeficiente de variación, sin embargo, su estimación es sencilla si combinamos la función `mean` y `sd`. También puede crear una función con el siguiente código:

```
CV<-function(x){
  (sd(x)/mean(x))*100
}
```

Error estándar

El error estándar es una medida que se utiliza para evaluar la variabilidad de un estadístico entre diferentes muestras. Este error tiende a disminuir conforme aumenta el tamaño de la muestra. Por lo tanto, un menor error estándar indica que la muestra es más representativa de la población en su conjunto.

El error estándar mide la precisión de la estimación de un estadístico a partir de su parámetro. Por ejemplo, para la media el error estándar se calcula como la desviación estándar de la distribución de las medias muestrales obtenidas de todas las posibles muestras de tamaño n extraídas de la población.

Si la variabilidad de las medias muestrales se mide por su desviación estándar, las fórmulas para el error estándar serían:

$$SE = \frac{\sigma}{\sqrt{N}}$$

$$SE = \frac{S}{\sqrt{n}}$$

El error estándar no disminuye de manera directamente proporcional al tamaño de la muestra, debido a que se utiliza la raíz cuadrada de n en su cálculo. Por ejemplo, para reducir el error estándar a la mitad, es necesario aumentar el tamaño de la muestra cuatro veces. Además de la media, el error estándar también puede calcularse para otros estadísticos.

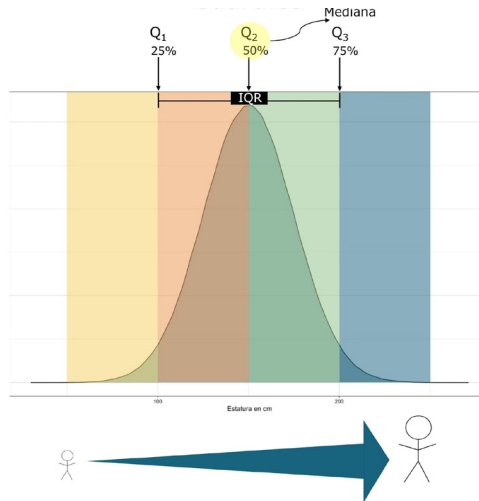
Medidas de posición

Las medidas de posición son aquellas que permiten, una vez ordenados los datos de menor a mayor, ubicar dónde se encuentra el 10%, 20%, 50%, etc., de los datos. Estos datos, además de la posición, nos permiten conocer la dispersión del conjunto de datos. El término n -tiles se refiere a n grupos en los que será dividido el conjunto de datos. Solemos llamarle cuarteliles a los datos que permiten dividir en 4 partes iguales a todo el conjunto de datos, mientras que nos referimos a deciles cuando permiten dividir el conjunto de datos en 10 partes iguales.

Cuartiles

Dividen al conjunto de datos en **cuatro** partes iguales; en cada una de ellas hay 25% de los datos (ver [figura 11.5](#)).

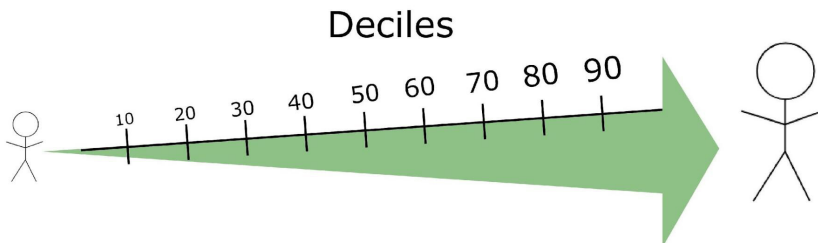
FIGURA 11.5. CUARTILES



Deciles

Se consideran como medidas de dispersión o de posición, las cuales dividen un conjunto de datos en 10 partes iguales en cuanto al número de datos (ver [figura 11.6](#)).

FIGURA 11.6. DECILES



Hay muchas formas de calcular los percentiles. La función `quantile()` en R cuenta con 9 algoritmos para estimar los -tiles. En este compendio no cambiaremos el argumento `type` y utilizaremos el argumento predefinido. Puede encontrar más información sobre como R calcula los -tiles utilizando la función de ayuda: `help("quantile")`. Tome este dato en cuenta si estima percentiles en diferentes programas estadísticos.

Funciones empleadas en la estadística descriptiva para variables cuantitativas

Los aspectos teóricos de la estadística descriptiva para las variables cuantitativas ya fueron explicados en la sección anterior. En esta sección, solo abordaremos ejemplos de cómo realizar la estimación de medidas de centralización y tendencia central utilizando R.

Existen diversas funciones para obtener estadísticos que nos permiten resumir datos cuantitativos. Las más comunes son:

- ♦ `mean()` función que devuelve la media de un vector numérico
- ♦ `median()` devuelve la mediana de una conjunto de datos
- ♦ `sd()` devuelve la desviación estándar de un vector numérico
- ♦ `var()` devuelve la varianza de un vector numérico
- ♦ `quantile()` devuelve los percentiles indicados.
- ♦ `IQR()` devuelve el rango intercuantil de un vector de datos numérico
- ♦ `summary()` devuelve un resumen de cualquier tipo de variable, en el caso de variables cuantitativas devuelve valores mínimo, máximo, mediana y media
- ♦ `min()` devuelve el valor mínimo de conjunto de datos
- ♦ `max()` devuelve el valor máximo de un conjunto de datos

El siguiente es un ejemplo de cómo estimar la estadística descriptiva para variables cuantitativas utilizando R base.

- ♦ **Ejemplo 11.11:** para la resolución de estos ejercicios vamos a utilizar la base de datos de `Pima.tr` de la librería `MASS`. El primer paso, por lo tanto, es llamar a la librería y luego a los datos:

```
library(MASS)
```

```
data(Pima.tr)
```

En esta ocasión no vamos a utilizar la función `attach` y vamos a trabajar todas las funciones con el símbolo de `$` para poder acceder a los objetos contenidos en nuestro mega objeto (*Dataframe*).

Vamos a calcular algunas medidas de tendencia central y algunas de dispersión para la variable `age`:

```
mean(x=Pima.tr$age, rm.na=T)
[1] 32.11
```

Note cómo empleamos dos argumentos en la función `mean()`. Todas las funciones requieren argumentos, en este caso utilizamos uno llamado `x` y otro `rm.na`. El primero hace alusión a un vector numérico y el segundo hace alusión a la omisión de los valores perdidos (NA). En R podemos omitir declarar los argumentos, siempre y cuando estén en el orden correcto. Por ejemplo, el código anterior se podría simplificar:

```
mean(Pima.tr$age, rm.na=T)
[1] 32.11
```

Note cómo omitimos declarar el valor de `x`, sin embargo, no podemos omitir declarar el argumento `na.rm` ya que no es el segundo argumento asignado a la función y R no sabrá qué hacer. El siguiente código daría un error:

```
mean(Pima.tr$age, T)
```

Ahora con una explicación más avanzada del uso de las funciones, podemos estimar el resto de los estadísticos:

```
median(Pima.tr$age)# Para estimar la mediana
[1] 28
sd(Pima.tr$age)# Para estimar la desviación estándar
[1] 10.97544
var(Pima.tr$age)# Para estimar la varianza
```

```
[1] 120.4602
IQR(Pima.tr$age)# Para estimar el rango intercuartil
[1] 16.25
quantile(Pima.tr$age)# Para estimar los cuartiles
  0%   25%   50%   75%  100%
21.00 23.00 28.00 39.25 63.00
quantile(Pima.tr$age, c(0.1,0.2, 0.4, 0.9))
  10%   20%   40%   90%
  21.0   23.0   26.0   49.1
# Para estimar los percentiles 10, 20, 40 y 90
min(Pima.tr$age)# Para estimar el valor mínimo
[1] 21
max(Pima.tr$age)# Para estimar el valor máximo
[1] 63
summary(Pima.tr$age)# Para hacer un resumen numérico más general
  Min.   1st Qu. Median Mean   3rd Qu. Max.
  21.00   23.00   28.00  32.11  39.25  63.00
```

Funciones de otras librerías

La función `describe()` de la librería `psych`

La función `describe()` del paquete `psych` en R es una herramienta muy útil para obtener un resumen detallado de las estadísticas descriptivas de un conjunto de datos, especialmente cuando se trabaja con múltiples variables.

La función `describe()` proporciona una amplia gama de estadísticas descriptivas para cada variable en el conjunto de datos, incluyendo medidas de tendencia central, dispersión, forma de la distribución y otros estadísticos descriptivos relevantes.

Algunos de los estadísticos descriptivos que la función `describe()` proporciona para cada variable incluyen:

- ♦ Media y mediana
- ♦ Desviación estándar y rango

- ◊ Valores mínimos y máximos
- ◊ Curtosis y asimetría
- ◊ Moda
- ◊ Cuántiles
- ◊ Además de estas medidas básicas, `describe()` también puede proporcionar otras estadísticas descriptivas más detalladas, como los valores atípicos, el coeficiente de variación, la entropía y la cantidad de datos faltantes.

◊ Ejemplo 11.12

```
install.packages("psych")
```

Vamos a crear un objeto con las edades hipotéticas de un grupo de pacientes:

```
set.seed(123) # Para reproducibilidad, sembrando una semilla
edades <- sample(18:80, 200, replace = TRUE)
psych::describe(edades)

vars  n  mean  sd   median trimmed mad   min   max  range  skew kurtosis  se
X1    1  200  48.51 16.85  47.5   48.33 22.24 18   80    62   0.07 -1.111.19
# La utilización de los cuatro puntos permite acceder a las
# funciones de una librería sin llamarla
```

La función `ds_tidy_stats()` de la librería `descriptr`

La función `ds_tidy_stats()` de la librería `descriptr` en R es una herramienta útil para calcular estadísticas descriptivas de un conjunto de datos y presentar los resultados en un formato “tidy”, que es un formato tabular organizado donde cada fila representa una observación y cada columna representa una variable o una medida descriptiva.

◊ Ejemplo 11.13

```
install.packages("descriptr")# Instalar librería
library(MASS)
data(Pima.tr)
```

```
# La función necesita un data frame
Pima.tr |>
  descriptr::ds_tidy_stats(glu)
# A tibble: 1 × 16
  vars min max mean t_mean median mode range variance stdev skew
<chr> <int> <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 glu  56 199 124. 123. 120. 100 143 1003. 31.7 0.456
# i 5 more variables: kurtosis <dbl>, coeff_var <dbl>, q1 <dbl>, q3 <dbl>,
# iqrange <dbl>
```

Ejercicios

- ♦ **Ejercicio 11.1:** utilizando la base de datos `Pima.tr` de la librería `MASS`, responda las siguientes preguntas:
 - a) ¿Cuál es el percentil 25 para la variable `glu`?
 - b) ¿Cuál es valor mínimo y máximo para la variable `bmi`?
 - c) Utilizando la función `summary` describa la variable `bp`
 - d) ¿Cuál es la edad promedio de aquellos pacientes con diabetes (`type = "Yes"`)?
 - e) ¿Cuál es el valor de la mediana para la glucosa de las pacientes sin diabetes?
 - f) Utilice la función `summary`, pero esta vez aplicada a todo el *Dataframe* (`summary(Pima.tr)`) y describa qué es lo que pasa.

Resolución **ejercicio 11.1**

```
# Si es necesario
library(MASS)
data(Pima.tr)
quantile(Pima.tr$glu, 0.25)
summary(Pima.tr$bmi)
summary(Pima.tr$bp)
mean(Pima.tr$age[Pima.tr$type=="Yes"])
median(Pima.tr$glu[Pima.tr$type=="No"])
```

- ♦ **Ejercicio 11.2:** utilizando la base de datos `Cushings` de la librería `MASS`, cree una tabla de frecuencias para la variable `Type`.

Resolución ejercicio 11.2

```
# Si es necesario
library(MASS)
data("Cushings")
x <- table(Cushings$Type)

data_freq <-
  data.frame(
    Nombres= names(x),
    Freq =as.numeric(x),
    Freq_Rela = as.numeric(prop.table(x)),
    Freq_Acum = as.numeric(cumsum(x)),
    Freq_Rela_Acum = as.numeric (cumsum(prop.table(x))))
print(data_freq)
questionr::freq(Cushings$Type)
```

- ♦ **Ejercicio 11.3:** utilizando la base de datos `birthwt` de la librería `MASS`, cree una tabla de frecuencias para la variable `race` y construya un gráfico de barras utilizando `ggplot2`.

Resolución ejercicio 11.3

```
# Si es necesario
library(MASS)
data("birthwt")
library(ggplot2)
birthwt$race <- factor(birthwt$race,
  labels = c("Grupo étnico White",
             "Grupo étnico Black",
             "Grupo étnico Otro"))

x <- table(birthwt$race)
```

```
data_freq <-
  data.frame(
    Nombres= names(x),
    Freq =as.numeric(x),
    Freq_Rela = as.numeric(prop.table(x)),
    Freq_Acum = as.numeric(cumsum(x)),
    Freq_Rela_Acum = as.numeric (cumsum(prop.table(x))))
print(data_freq)
birthwt|>
  ggplot(aes(x=race, fill = race))+
  geom_bar()+
  theme_linedraw()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
# La última parte del código cambia los ángulos de los ejes
# a 45 grados
```

- ♦ **Ejercicio 11.4:** utilice la base datos del [ejemplo 11.5](#) y genere estadística descriptiva que incluya: media, mediana, desviación estándar, varianza, Q1, Q2, Q3, valor mínimo y máximo para todas las variables numéricas de la base de datos dividido por hombres y mujeres.

Resolución [ejercicio 11.4](#)

Identifique cuáles son las variables numéricas. Puede utilizar la función `str()` o `summary()` para ello.

Se muestra la resolución solo para una de las variables:

```
# De ser necesario
df <- readRDS(file = "Bases/Base Descriptivos.rds")
tapply(df$Edad, INDEX = df$Genero, FUN = summary)
tapply(df$Edad, INDEX = df$Genero, FUN = var)
tapply(df$Edad, INDEX = df$Genero, FUN = sd)
```


Puede optimizar el código para todas las variables:

```
# 2:4, 14:17,19 son las variable numéricas de la base
lapply(df[,c(2:4, 14:17,19)], FUN = tapply, df$Genero, summary)
```

- ♦ **Ejercicio 11.5:** utilizando la base de datos `birthwt` de la librería `birthwt`, cree gráfico de mosaico con las proporciones de las variables `low` y `ht`. Realice una interpretación de su gráfico.

Resolución ejercicio 11.5

```
colores <- c("#fd7f6f", "#7eb0d5", "#b2e061",
             "#bd7ebe", "#ffb55a", "#ffee65",
             "#beb9db", "#fdcce5", "#8bd3c7")

table(birthwt$low, birthwt$ht)|>
  mosaicplot(col=colores,
             main="Gráfico de mosaicos para bajo peso e HTA")
```

- ♦ **Ejercicio 11.6:** utilizando la base de datos `Pima.tr2` de la librería `MASS`, realice cortes de la variable `BMI` para clasificar a las pacientes, según los criterios de la OMS descritos en la siguiente tabla:

CLASIFICACIÓN	IMC (KG/M2)	RIESGO ASOCIADO A LA SALUD
Normo peso	18.5 - 24.9	Promedio
Sobrepeso o preobeso	25 - 29.9	AUMENTADO
Obesidad grado I o moderada	30 - 34.9	AUMENTO MODERADO
Obesidad grado II o severa	35 - 39.9	AUMENTO SEVERO
Obesidad grado III o mórbida	≥ 40	AUMENTO MUY SEVERO

Incluya la nueva variable en la base datos y cree un gráfico de barras mediante el paquete `base` y otra utilizando `ggplot`.

Resolución ejercicio 11.6

```
# Si es necesario
library(MASS)
data("Pima.tr2")
library(ggplot2)
Pima.tr2$bmi_gdo ← cut(Pima.tr2$bmi,
                      breaks = c(0,25,30,35,40,Inf),
                      labels = c("Normo peso", "Sobrepeso", "Obesidad I",
                                "Obsedidad II", "Obesidad III"))
tab3 ← table(Pima.tr2$bmi_gdo)
barplot(tab3, col=colores,
        xlab="Grado de obesidad",
        ylab="Frecuencias",
        main="Grado de obesidad según la OMS")

Pima.tr2|>
  ggplot(aes(x=bmi_gdo, fill = bmi_gdo))+
  geom_bar()+
  theme_bw()+
  labs(title = "Gráfico de barras para la
         clasificación de BMI según la OMS",
        x= "Conteos",
        y= "Frecuencias")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

- ♦ **Ejercicio 11.7:** cree una gráfica de barras de las variable que creó en el ejercicio anterior y una gráfica de mosaico que agrupe la nueva variable y el diagnóstico de diabetes en los pacientes.

Resolución ejercicio 11.7

```
tab3 ← table(Pima.tr2$bmi_gdo)
barplot(tab3, col=colores,
        xlab="Grado de obesidad",
```

```

      ylab="Frecuencias",
      main="Grado de obesidad según la OMS")
tab4 ← table(Pima.tr2$bmi_gdo, Pima.tr2$type)
mosaicplot(tab4, col=colores, main="")

```

- ♦ **Ejercicio 11.8:** el siguiente código crea una variable edades que contiene datos aleatorios para las edades de un grupo de pacientes.

```

set.seed(123) # Para reproducibilidad, sembrando una semilla
edades ← sample(18:80, 200, replace = TRUE)

```

Utilizando esa variable, cree una variable ordinal que contenga los siguientes niveles: [18,30), [30,40), [40,50), [50,60), [60,70) y [70,80).

Cambie las etiquetas y realice lo siguiente:

- Cree una tabla de frecuencias.
- Cree un gráfico de barras en el que el eje de las “y” sean porcentajes.
- Cree un gráfico de sectores y utilice una paleta de colores que le permita combinar los colores de su gráfico con R base y luego cree otro gráfico con ggplot. Puede apoyarse de internet.

Resolución ejercicio 11.8

```

edades_gdo ← cut(edades, breaks = c(18,30,40,
                                     50,60,70, Inf),
                 labels = c ("Gpo Edad 1",
                             "Gpo Edad 2",
                             "Gpo Edad 3",
                             "Gpo Edad 4",
                             "Gpo Edad 5",
                             "Gpo Edad 6"))
tab5 ← table(edades_gdo)
tab6 ← prop.table(tab5)
barplot(tab6*100, col = col
res, #Objeto creado anteriormente con colores
      ylab="%",

```

```

        xlab="Grupo de edades",
        main="Gráfico de barras para un grupo de edads")
pie(tab6, main = "Grupos de edades", # Título del gráfico
    col = colores, # Colores
    density = 50, # Densidad de las líneas de sombreado en el gráfico
    angle = 45 # Ángulo de las líneas de sombreado
)

# Añadir una leyenda al gráfico para facilitar la interpretación
legend(
  "topright", # Posición de la leyenda dentro del gráfico
  c("Gpo Edad 1", "Gpo Edad 2", "Gpo Edad 3",
    "Gpo Edad 4", "Gpo Edad 5", "Gpo Edad 6"), # Etiquetas
  fill = colores) # Colores
Tabla_conteos <- data.frame(
  Grupo= c("Gpo Edad 1", "Gpo Edad 2", "
           Gpo Edad 3", "Gpo Edad 4",
           "Gpo Edad 5", "Gpo Edad 6"),
  Conteo= c(36, 32, 42, 33, 32, 24)
)

Tabla_conteos|>
  ggplot(mapping = aes(x="", y=Conteo, fill = Grupo))+
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0)+
  labs(title = "Gráfico de sectores para la edad",
       x = "Conteos",
       y = "Grupo de edad")+
  theme_bw()

```

- ♦ **Ejercicio 11.9:** en un estudio sobre la prevalencia de una enfermedad crónica en una población, se recopiló la edad de 100 pacientes.
 - a) Genere un vector de 100 edades aleatorias entre 18 y 80 años. Utilice la función `sample`.
 - b) Utilice la función `cut` para categorizar estas edades en los siguientes grupos: 18-30, 31-45, 46-60 y más de 60 años. Asegúrese de cambiar las etiquetas.
 - c) Cree una tabla de frecuencias para mostrar cuántos pacientes caen en cada grupo de edad.
 - d) Cree un gráfico de barras con barras diferentes para cada categoría, cambie los títulos de los ejes y el título del gráfico.

Resolución ejercicio 11.9

```
set.seed(123)
edades2 <- sample(x=18:80, size = 100, replace = T)
edades2_gdo <- cut(edades2, breaks = c(17,30,45,60,Inf),
                  labels = c("18-30", "31-45", "46-60",
                           "más de 60 años"))
tab7 <- table(edades2_gdo)
tab7
barplot(tab7, col = colores,
        ylab="Frecuencias",
        xlab="Grupo de edades",
        main="Gráfico de barras para un grupo de edades 2")
```

- ♦ **Ejercicio 11.10:** se tiene un conjunto de datos que incluye el tipo de tratamiento recibido por pacientes con una enfermedad específica: Tratamiento A, Tratamiento B o Tratamiento C.
 - a) Cree un vector que represente el tipo de tratamiento recibido por 150 pacientes, asegúrese de que los tratamientos estén distribuidos de manera no uniforme.
 - b) Utilice la función `table` para obtener la frecuencia de cada tipo de tratamiento.

- c) Emplee `barplot` para crear un gráfico de barras que muestre visualmente estas frecuencias. Cambie los títulos de los ejes y el título del gráfico.

Resolución ejercicio 11.10

```
set.seed(123)
tratamientos ← sample(x=c("Tratamiento A",
                           "Tratamiento B",
                           "Tratamiento C"), size = 150,
                      replace = T)
table(tratamientos)
barplot(table(tratamientos), col=colores,
        xlab="Tratamientos",
        ylab="Frecuencias",
        main="Tratamientos de una enfermedad")
```

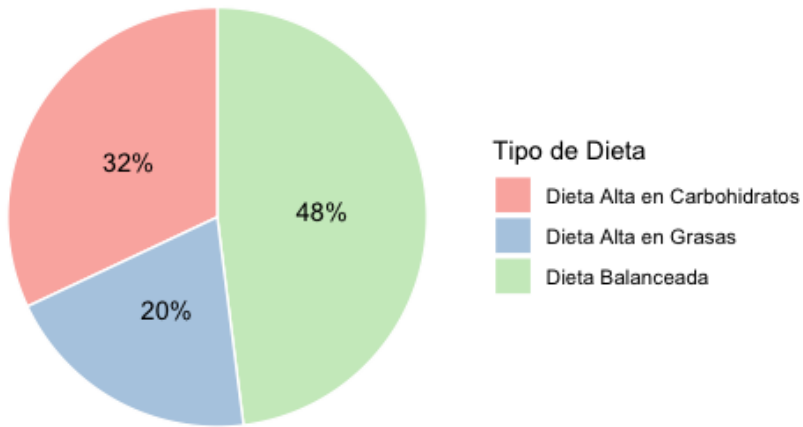
- ♦ **Ejercicio 11.11:** en un estudio sobre hábitos de vida saludables, se encuestó a personas sobre su nivel de actividad física: bajo, moderado o alto.
- Genere un vector que simule las respuestas de 200 personas sobre su nivel de actividad física.
 - Calcule la frecuencia de cada nivel de actividad utilizando `table`.
 - Use `prop.table` para convertir estas frecuencias en proporciones.
 - Cree un gráfico de pastel (`pie`) para representar estas proporciones.

Resolución ejercicio 11.11

```
set.seed(123)
actividad ← sample(x=c("bajo", "moderado", "alto"), size = 200,
                  replace = T)
table(actividad)
barplot(table(actividad), col=colores,
        xlab="Actividad física",
        ylab="Frecuencias",
        main="Hábitos saludables")
```

- ♦ **Ejercicio 11.12:** observe el gráfico de pastel proporcionado. Responda las siguientes preguntas: a) ¿qué tipo de dieta es la más común entre los pacientes?, b) ¿cuál es la proporción de pacientes que siguen una dieta balanceada?, c) discuta cómo los resultados podrían influir en las recomendaciones nutricionales para pacientes con diabetes tipo 2.

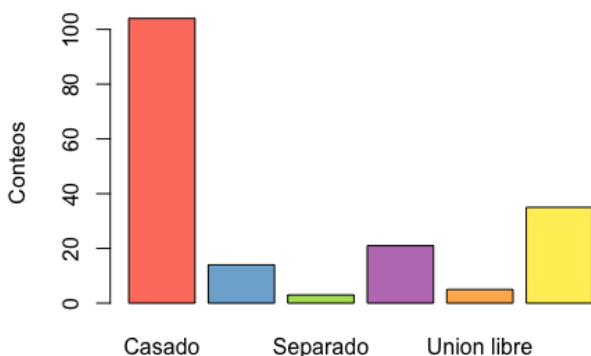
Distribución de Tipos de Dieta en Pacientes con Diabetes Tipo 2



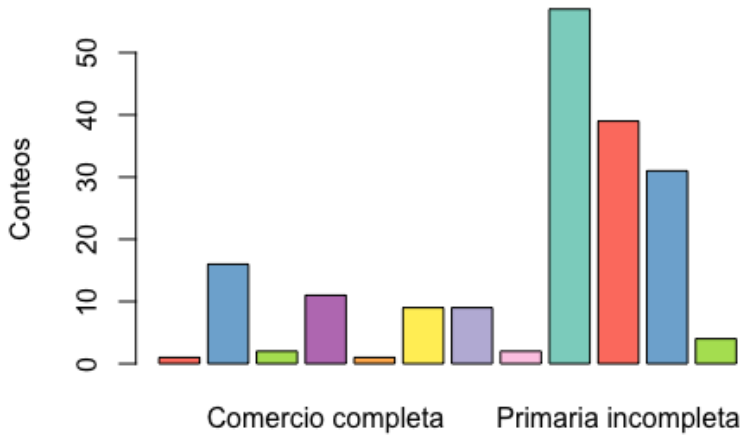
- ♦ **Ejercicio 11.13:** utilice la base de datos [Base Descriptivo Cualitativos](#) para realizar una tabla de frecuencias y un gráfico de barras con las siguientes variables:
- CIVIL
 - ESCOLARIDAD
 - GDO_ESCOLARIDAD
 - OCUPACION
 - GDO_IMC

Resolución ejercicio 11.3

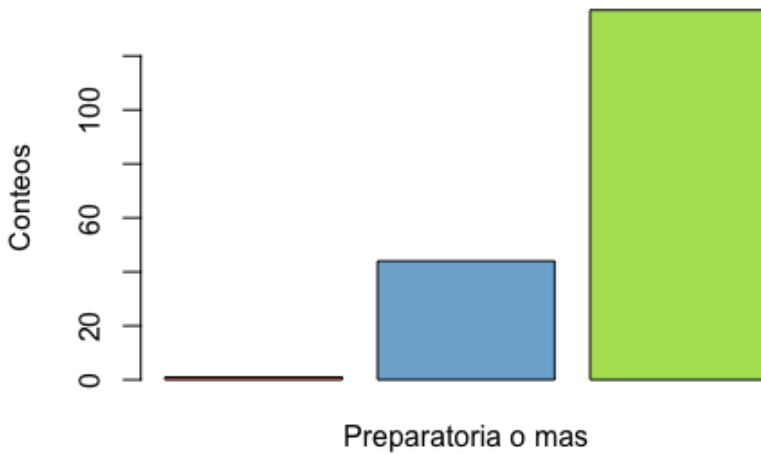
	Nombres	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum
1	Casado	104	0.57142857	04	0.5714286
2	Divorciado	4	0.07692308	118	0.6483516
3	Separado	3	0.01648352	121	0.6648352
4	Soltero	21	0.11538462	142	0.7802198
5	Union libre	5	0.02747253	147	0.8076923
6	Viudo	35	0.19230769	182	1.0000000



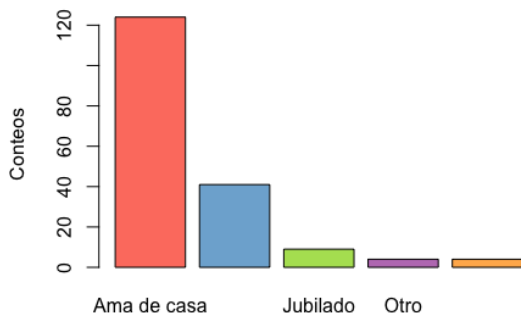
	Nombres	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum
1		1	0.005494505	1	0.005494505
2	Carrera completa	16	0.087912088	17	0.093406593
3	Carrera incompleta	2	0.010989011	19	0.104395604
4	Comercio completa	11	0.060439560	30	0.164835165
5	Estudios de Posgrado	1	0.005494505	31	0.170329670
6	Nula	9	0.049450549	40	0.219780220
7	Preparatoria completa	9	0.049450549	49	0.269230769
8	Preparatoria incompleta	2	0.010989011	51	0.280219780
9	Primaria completa	57	0.313186813	108	0.593406593
10	Primaria incompleta	39	0.214285714	147	0.807692308
11	Secundaria completa	31	0.170329670	178	0.978021978
12	Secundaria incompleta	4	0.021978022	182	1.000000000



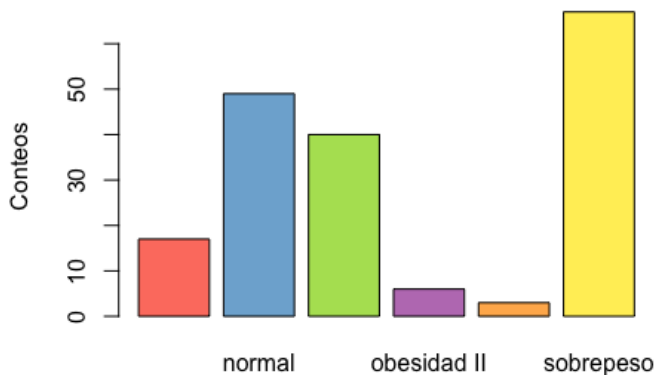
Nombres	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum1
1	1	0.005494505	1	0.005494505
2 Preparatoria o mas	44	0.241758242	45	0.247252747
3 Secundaria o menos	137	0.752747253	182	1.000000000



	Nombres	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum1
1	Ama de casa	124	0.68131868	124	0.6813187
2	Empleado	41	0.22527473	165	0.9065934
3	Jubilado	9	0.04945055	174	0.9560440
4	Otro	4	0.02197802	178	0.9780220
5	Pensionado	4	0.02197802	182	1.0000000



	Nombres	Freq	Freq_Rela	Freq_Acum	Freq_Rela_Acum1
1		17	0.09340659	17	0.09340659
2	normal	49	0.26923077	66	0.36263736
3	obesidad I	40	0.21978022	106	0.58241758
4	obesidad II	6	0.03296703	112	0.61538462
5	Obesidad III	3	0.01648352	115	0.63186813
6	sobrepeso	67	0.36813187	182	1.00000000



- ♦ **Ejercicio 11.14:** usted pertenece a un grupo de investigadores que tienen la tarea de analizar el peso al nacer (variable `bwt`), utilizando medidas de tendencia central y dispersión, y crear gráficos que permitan visualizar la distribución de esta variable y su relación con otras variables. Todo ello utilizando la base de datos `birthwt`.

Le encomiendan seguir y realizar las siguientes instrucciones:

1. Cargue y explore la base de datos:
 - Cargue la base de datos `birthwt` del paquete `MASS`.
 - Explore las primeras filas y la estructura de la base de datos para familiarizarse con las variables.
2. Medidas de tendencia central y dispersión:
 - Calcule la media, mediana, moda, varianza, desviación estándar, rango intercuartílico (IQR) y rango del peso al nacer (`bwt`).
3. Visualice los datos con `ggplot`:
 - Cree un histograma para visualizar la distribución del peso al nacer.
 - Cree un boxplot para visualizar la distribución del peso al nacer y compare entre diferentes niveles de la variable `smoke` (fumar durante el embarazo).
4. Interprete sus resultados.

Resolución [ejercicio 11.14](#)

```
[1] "Exploración de la base de datos"
```

	low	age	lwt	race	smoke	ptl	ht	ui	ftv	bwt
85	0	19	82	2	0	0	0	1	0	2523
86	0	33	155	3	0	0	0	0	3	2551
87	0	20	105	1	1	0	0	0	1	2557
88	0	21	108	1	1	0	0	1	2	2594
89	0	18	107	1	1	0	0	1	0	2600
91	0	21	124	3	0	0	0	0	0	2622

```
data.frame': 189 obs. of 10 variables:
```

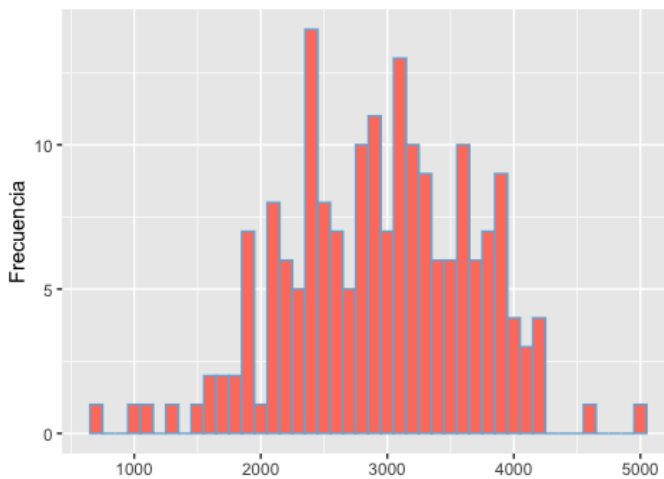
```
$ low : int 0 0 0 0 0 0 0 0 0 0 ...
$ age : int 19 33 20 21 18 21 22 17 29 26 ...
```

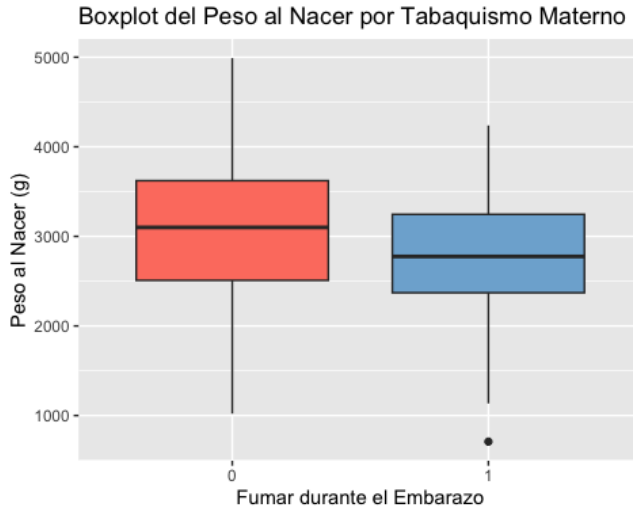
```
$ lwt : int 182 155 105 108 107 124 118 103 123 113 ...
$ race : int 2 3 1 1 1 3 1 3 1 1 ...
$ smoke: int 0 0 1 1 1 0 0 0 1 1 ...
$ ptl : int 0 0 0 0 0 0 0 0 0 0 ...
$ ht : int 0 0 0 0 0 0 0 0 0 0 ...
$ ui : int 1 0 0 1 1 0 0 0 0 0 ...
$ ftv : int 0 3 1 2 0 0 1 1 1 0 ...
$ bwt : int 2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

[1] "Medidas de tendencia central y dispersión"

	Medida	Valor
1	Media	2944.5873
2	Mediana	2977.0000
3	Moda	3062.0000
4	Varianza	531753.4883
5	Desviación Estándar	729.2143
6	Rango Intercuartílico	1073.0000
7	Rango	4281.0000

Histograma del Peso al Nacer





Referencias

- Barnier J, Briatte F y Larmarange J (2023). *questionr: Functions to Make Surveys Processing Easier*. R package version 0.7.
- Bear, A. y Knobe, J. (2017). Normality: Part descriptive, part prescriptive. *Cognition*, 167, 25–37. <https://doi.org/10.1016/j.cognition.2016.10.024>
- Brownstein, N. C., Adolfsson, A. y Ackerman, M. (2019). *Descriptive statistics and visualization of data from the R datasets package with implications for clusterability*. *Data in brief*, 25, 104004. <https://doi.org/10.1016/j.dib.2019.104004>
- Firke, S. (2023). *janitor: Simple Tools for Examining and Cleaning Dirty Data*. R package version 2.2.0. <https://CRAN.R-project.org/package=janitor>
- Hebbali, A. (2020). *descriptr: Generate Descriptive Statistics*. R package version 0.5.2. <https://CRAN.R-project.org/package=descriptr>
- Indrayan, A. y Malhotra, R. K. (2017). Confidence Intervals, Principles of Tests of Significance, and Sample Size. En *Medical Biostatistics* (4ª ed., pp. 255-304). Chapman & Hall/CRC Biostatistics Series.

- Kim, H., Sefcik, J. S. y Bradway, C. (2017). Characteristics of Qualitative Descriptive Studies: A Systematic Review. *Research in nursing & health*, 40(1), 23-42. <https://doi.org/10.1002/nur.21768>
- Martínez, M. Á., Toledo, E., Donat-Vargas, C. y Lopez-Fidalgo, J. (2020). Procedimientos descriptivos. En M. Á. Martínez González, A. Sánchez Villegas, E. Toledo Atucha y J. Faulin Fajardo (Eds.), *Bioestadística amigable* (4ª ed., pp. 13-63).
- McHugh, M. L. (2003). Descriptive statistics, Part II: Most commonly used descriptive statistics. *Journal for specialists in pediatric nursing: JSPN*, 8(3), 111-116. <https://doi.org/10.1111/j.1088-145x.2003.00111.x>
- Shahbaba, B. (2011). Introduction. En *Biostatistics with R: An Introduction to Statistics Through Biological Data* (pp. 1-16). Springer. <https://doi.org/10.1007/978-1-4614-1302-8>
- Spiestersbach, A., Röhrig, B., du Prel, J. B., Gerhold-Ay, A. y Blettner, M. (2009). Descriptive statistics: the specification of statistical measures and their presentation in tables and graphs. Part 7 of a series on evaluation of scientific publications. *Deutsches Arzteblatt international*, 106(36), 578-583. <https://doi.org/10.3238/arztebl.2009.0578>
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics with S. Fourth Edition*. Springer.
- Wickham, H., Çetinkaya-Rundel, M. y Grolemund, G. (2018). *R for Data Science: Import, Tidy, Transform, Visualize, and Model*. <https://r4ds.hadley.nz/>
- Wickham, H., Navarro, D. y Lin Perderse, T. (2016). *ggplot2: Elegant Graphics for Data Analysis (3e)* (3.a ed.). <https://ggplot2-book.org/>
- Wissing, D. R. y Timm, D. (2012). Statistics for the nonstatistician: Part I. *Southern medical journal*, 105(3), 126-130. <https://doi.org/10.1097/SMJ.0b013e3182498ad5>

Distribuciones de probabilidad

Alberto Beltrán Ramírez

Edsaúl Emilio Pérez Guerrero

Imagine que puede acceder y conocer las estaturas de todos los estudiantes de posgrado de una determinada universidad. La totalidad de estos alumnos es de 12500 estudiantes. Estos datos son tabulados y graficados, y además se encuentra una función matemática que describe las estaturas y permite identificar los parámetros que las definen. Ahora, suponga que usted recibe una serie de estaturas de unos estudiantes desconocidos. Con los datos generados a partir de la distribución de las estaturas, fácilmente podría evaluar la probabilidad de que estos estudiantes fueran de posgrado de dicha universidad. En este caso, los datos de los 12500 estudiantes sirvieron para generar una distribución que posteriormente nos permitió conocer las probabilidades de que las estaturas de estos nuevos alumnos pertenezcan o no a la universidad.

Una distribución de probabilidad es una función matemática que proporciona los posibles valores de una variable aleatoria y la probabilidad de que estos valores ocurran. Por ello pudimos determinar la probabilidad de que los alumnos pertenecieran a la universidad. Las distribuciones pueden ser continuas o discretas; las primeras hacen referencia a distribuciones de probabilidad de variables aleatorias que pueden tomar cualquier valor dentro de un rango continuo, mientras que las discretas solo pueden tomar valores específicos y contables.

En este capítulo, realizaremos una descripción de las principales distribuciones de probabilidad. En esta breve revisión, identificaremos las principales características de cada distribución, así como los parámetros que las definen.

Es necesario que para los ejemplos de este capítulo considere que un evento puede tener la probabilidad de ocurrencia de 0 a 1.

Tome en cuenta que existen muchas más distribuciones, por lo que lo invitamos a revisar información complementaria en caso de ser necesario.

Distribución uniforme discreta

La distribución uniforme discreta es una distribución de probabilidad en la que un conjunto finito de valores tiene la misma probabilidad de ocurrencia. En términos más simples, si hay n posibles resultados, cada uno tiene una probabilidad de $\frac{1}{n}$.

Por ejemplo, considere lanzar un dado de seis caras. Tiene la misma probabilidad de obtener los seis números en su siguiente tirada, es decir, obtener precisamente uno de 1, 2, 3, 4, 5 o 6, lo que equivale a una probabilidad de $1/6$; de ahí un ejemplo de distribución uniforme discreta.

Cálculo

$$P(X = k) = \frac{1}{n} \text{ para } k = 1, 2, \dots, n$$

Usos en bioestadística y principales características

En una distribución uniforme discreta se debe tener en cuenta que:

- ♦ Todos los eventos son igualmente probables.
- ♦ La suma de las probabilidades de todos los posibles resultados es 1.
- ♦ No hay preferencia o sesgo hacia ningún resultado específico.

Distribución uniforme en R

Para implementar y trabajar con la distribución uniforme discreta en R, utilizamos las siguientes funciones:

- ♦ `dunif()`: función de densidad de probabilidad.
- ♦ `punif()`: función de distribución acumulativa.

- ◇ `qunif()`: función cuantílica.
- ◇ `runif()`: generador de números aleatorios (en el caso de distribuciones continuas, pero discretas podemos utilizar `sample`).
- ◇ **Ejemplo 12.1:** suponga que un investigador está estudiando la distribución de tipos de sangre entre un grupo pequeño de pacientes y quiere asignar a cada paciente un tipo de sangre de manera aleatoria. Si hay 4 tipos de sangre (A, B, AB y O) y cada tipo de sangre es igualmente probable, podemos modelar esto con una distribución uniforme discreta. En R podemos emplear el siguiente código para esta distribución:

```
# Parámetros
valores <- c("A", "B", "AB", "O")
k <- length(valores)

# Probabilidad de cada tipo de sangre
probabilidad <- rep(1/k, k)
names(probabilidad) <- valores

# Mostrar las probabilidades
probabilidad
  A      B      AB      O
0.25 0.25 0.25 0.25

# Generar una muestra aleatoria de tipos de sangre para 20 pacientes
set.seed(123)
muestra <- sample(valores, size = 20, replace = TRUE)
table(muestra)
muestra
  A      AB      B      O
 3      8      7      2
```

Ahora ya podemos hacer estimaciones, por ejemplo:

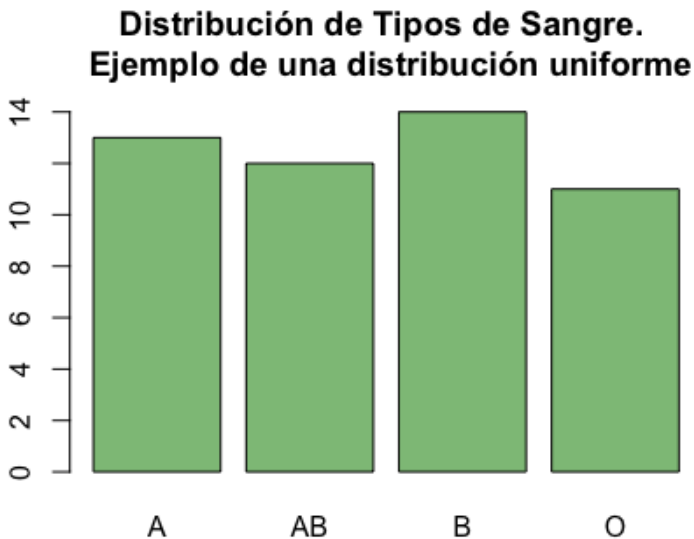
Podemos calcular la probabilidad de que un paciente tenga el tipo de sangre “A” en un grupo de 5 tipos de sangre igualmente probables:

```
# Ejercicio 1
valores_5 <- c("A", "B", "AB", "O", "C")
k_5 <- length(valores_5)
probabilidad_5 <- rep(1/k_5, k_5)
names(probabilidad_5) <- valores_5

# Probabilidad de tipo de sangre "A"
probabilidad_5["A"]
A
0.2
```

También podemos generar una muestra de 50 pacientes con tipos de sangre distribuidos uniformemente y visualizar los resultados en un gráfico de barras:

```
set.seed(456) # Sembrando la semilla
muestra_50 <- sample(valores, size = 50, replace = TRUE)
barplot(table(muestra_50), col = "#8EC187", main = "Distribución de Tipos de Sangre. \n Ejemplo de una distribución uniforme")
```



Distribución de Bernoulli

La distribución de Bernoulli es una distribución de probabilidad discreta que toma solo dos valores: 1 (éxito) con probabilidad p y 0 (fracaso) con probabilidad $1 - p$. Un ejemplo de esto es el resultado de un volado, donde el resultado es cara (éxito) o cruz (fracaso) y la probabilidad de que salga cara es un número entre 0 y 1.

Cálculo

$$P(X = k) = p^k (1 - p)^{1-k} \quad \text{para } k = 0, 1$$

Donde: X es una variable aleatoria que sigue una distribución de Bernoulli, k puede tomar los valores 0 o 1, p es la probabilidad de éxito (cuando $(k = 1)$ y $(1 - p)$ es la probabilidad de fracaso (cuando $(k = 0)$).

Usos en Bioestadística

En el campo de las ciencias de la salud, la distribución de Bernoulli es muy utilizada en situaciones donde los resultados pueden clasificarse en dos categorías exclusivas: éxito o fracaso, sí o no, presencia o ausencia de una condición, etcétera.

Distribución de Bernoulli en R

Para trabajar con una distribución de Bernoulli en R utilice las siguientes funciones:

- ♦ `dbinom()`
- ♦ `pbinom()`
- ♦ `qbinom()`
- ♦ `rbinom()`

- ♦ **Ejemplo 12.2:** calcule la probabilidad de éxito y fracaso para un tratamiento con una probabilidad de éxito de 0.85:

```
# Ejercicio 1
```

```
p_tratamiento <- 0.85
prob_exitito <- p_tratamiento

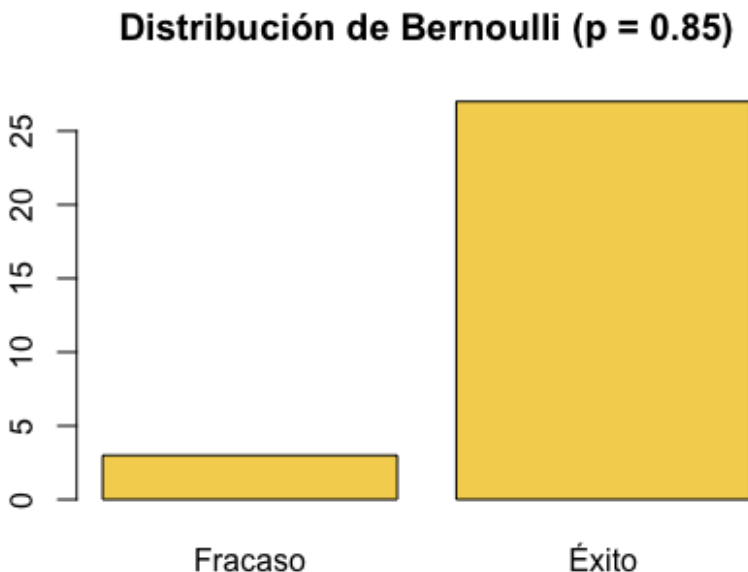
prob_fracaso <- 1 - p_tratamiento

prob_exitito
[1] 0.85
prob_fracaso
[1] 0.15
```

- ◇ **Ejemplo 12.3:** genere una muestra de 30 observaciones de una distribución de Bernoulli con $p = 0.85$ y visualice los resultados en un gráfico de barras:

```
set.seed(456) # Sembrando una semilla
muestra_bernoulli_30 <- rbinom(n = 30, size = 1, prob = p_tratamiento)

barplot(table(muestra_bernoulli_30), col = "#F4D35E", main = "Distribución de Bernoulli (p = 0.85)", names.arg = c("Fracaso", "Éxito"))
```



Distribución binomial

La distribución binomial es la suma de variables aleatorias independientes de Bernoulli, todas las cuales tienen la misma probabilidad de éxito. La cantidad se llama número de intentos y probabilidad de éxito.

Cálculo

$$P(X = k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \quad \text{para } k = 0, 1, 2, \dots, n$$

Donde:

- ♦ x es la variable aleatoria que representa el número de éxitos en ensayos de Bernoulli.
- ♦ n es el número total de ensayos.
- ♦ k es el número de éxitos.
- ♦ p es la probabilidad de éxito en un solo ensayo.
- ♦ $1-p$ es la probabilidad de fracaso en un solo ensayo.
- ♦ n/k es el coeficiente binomial calculado como .

Usos en Bioestadística y principales características

- ♦ Valores posibles: $k = 0, 1, 2, \dots, n$
- ♦ Media: $\mu = np$
- ♦ Varianza: $\sigma^2 = np(1-p)$

La distribución binomial nos serviría para responder ¿cuál es la probabilidad de que exactamente 12 pacientes respondan positivamente, si se sabe que la probabilidad de que un paciente responda positivamente al tratamiento es del 60% en un ensayo clínico de 20 pacientes?

Distribución bionomial en R

Para trabajar con una distribución de Bernuilli en R utilice las siguientes funciones:

- ♦ `dbinom()`
- ♦ `pbinom()`
- ♦ `qbinom()`
- ♦ `rbinom()`

Note cómo son las mismas que Bernoulli.

- ♦ **Ejemplo 12.4:** ¿cuál es la probabilidad de que exactamente 12 pacientes respondan positivamente, si se sabe que la probabilidad de que un paciente responda positivamente al tratamiento es del 60% en un ensayo clínico de 20 pacientes?

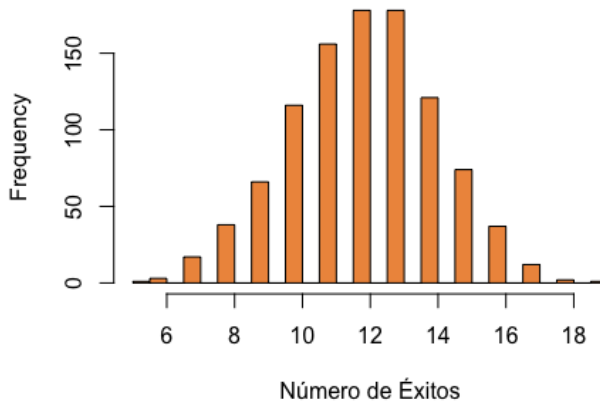
```
# Parámetros
n ← 20 # número de ensayos
p ← 0.6 # probabilidad de éxito

# Probabilidad de observar exactamente 12 éxitos
prob_12_exitos ← dbinom(12, size = n, prob = p)
prob_12_exitos
[1] 0.1797058

# Probabilidad acumulada de observar hasta 12 éxitos
prob_acumulada ← pbinom(12, size = n, prob = p)
prob_acumulada
[1] 0.5841071

# Generar una muestra de tamaño 1000 de una distribución binomial
set.seed(123)
muestra_binomial ← rbinom(n = 1000, size = n, prob = p)
hist(muestra_binomial, breaks = n, col = '#EE964B', main = 'Histograma de
Éxitos del Tratamiento', xlab = 'Número de Éxitos')
```

Histograma de Éxitos del Tratamiento

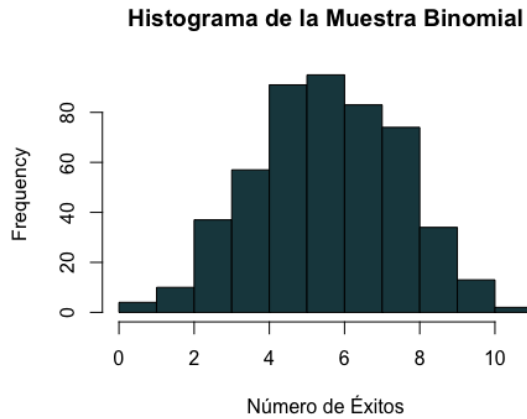


- ♦ **Ejemplo 12.5:** calcule la probabilidad de observar exactamente 15 éxitos en 30 ensayos con una probabilidad de éxito del 70%:

```
# Ejercicio 1
n_ensayos <- 30
p_exito <- 0.7
prob_15_exitos <- dbinom(15, size = n_ensayos, prob = p_exito)
prob_15_exitos
[1] 0.01056697
```

- ♦ **Ejemplo 12.6:** genere una muestra de 500 observaciones de una distribución binomial con $n = 15$ y $p = 0.4$. Visualice los resultados en un histograma:

```
n_muestra <- 500
n_ensayos_2 <- 15
p_exito_2 <- 0.4
set.seed(456)
muestra_binomial_2 <- rbinom(n = n_mue
tra, size = n_ensayos_2, prob = p_exito_2)
hist(muestra_binomial_2, breaks = n_ensayos_2, col = '#1C464D', main =
'Histograma de la Muestra Binomial', xlab = 'Número de Éxitos')
```



Distribución Poisson

La distribución de Poisson describe la probabilidad de encontrar exactamente x eventos en un periodo determinado, si los eventos ocurren de forma independiente a un ritmo constante.

La distribución de Poisson describe la probabilidad de que ocurran un número fijo de eventos en un intervalo fijo de tiempo o espacio, siempre que estos eventos ocurran con una tasa promedio constante y de manera independiente unos de otros.

Cálculo

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{para } k = 1, 2, \dots$$

Donde:

- ♦ x es la variable aleatoria que representa el número de eventos en un intervalo de tiempo o espacio.
- ♦ k es el número de eventos.
- ♦ λ es la tasa promedio de ocurrencia de eventos por intervalo de tiempo o espacio. e es la base del logaritmo natural (aproximadamente 2.71828).

Distribución de Poisson en R

- ◇ `dpois()`
 - ◇ `ppois()`
 - ◇ `qpois()`
 - ◇ `rpois()`
- ◇ **Ejemplo 12.7:** un ejemplo típico en ciencias de la salud es el número de llamadas que recibe un centro de emergencias médicas en una hora. Si se sabe que, en promedio, se reciben 5 llamadas por hora, podemos modelar esto con una distribución de Poisson con $\lambda = 5$ utilizando el siguiente código:

```
# Parámetro
lambda ← 5 # tasa promedio de eventos (llamadas por hora)

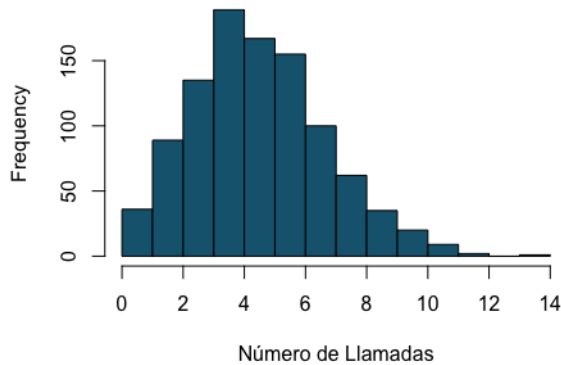
# Probabilidad de recibir exactamente 3 llamadas en una hora
prob_3_llamadas ← dpois(3, lambda)
prob_3_llamadas
[1] 0.1403739

# Probabilidad acumulada de recibir hasta 3 llamadas en una hora
prob_acumulada_3_llamadas ← ppois(3, lambda)

prob_acumulada_3_llamadas
[1] 0.2650259

# Generar una muestra de tamaño 1000 de una distribución de Poisson
set.seed(123)
muestra_poisson ← rpois(n = 1000, lambda = lambda)
hist(muestra_poisson, breaks = max(muestra_poisson), col = '#19647E', main =
'Histograma de Llamadas por Hora', xlab = 'Número de Llamadas')
```

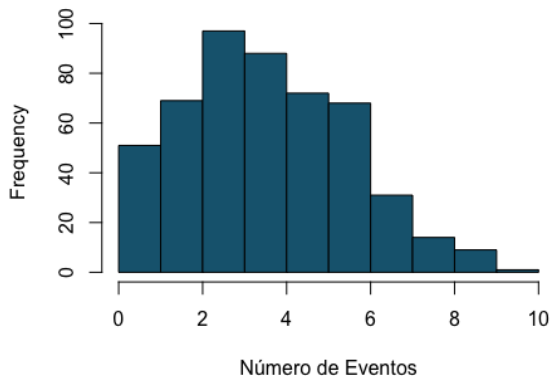
Histograma de Llamadas por Hora



Genere una muestra de 500 observaciones de una distribución de Poisson con $\lambda = 4$ y visualice los resultados en un histograma:

```
lambda_4 <- 4
set.seed(456)
muestra_poisson_2 <- rpois(n = 500, lambda = lambda_4)
hist(muestra_poisson_2, breaks = max(muestra_poisson_2), col =
'#19647E',
main = 'Histograma de la Muestra Poisson', xlab = 'Número de Eventos')
```

Histograma de la Muestra Poisson



Distribución normal

La distribución normal o de Gauss es una de las distribuciones de probabilidad más utilizada e importante en Estadística. Es esencial para muchos procedimientos estadísticos debido a sus propiedades matemáticas. Algunas de sus características principales, es el uso de la media, mediana, moda y desviación estándar. La distribución normal se define por los valores de la media (μ) y de la desviación estándar (σ). Para cada valor de μ y de σ hay una distribución.

Para determinar la probabilidad de una distribución normal, utilice la función `pnorm()` y la función `qnorm()`, para los valores.

Otros aspectos relacionados con la distribución normal serán revisados en el [capítulo 13](#).

Grados de libertad

Antes de abordar las últimas distribuciones de este capítulo, debemos explicar qué es un grado de libertad, parámetro indispensable para entender las siguientes líneas. Los grados de libertad son el número de valores independientes que pueden variar en una muestra sin violar ninguna restricción impuesta. En otras palabras, son la cantidad de datos que pueden ser seleccionados con la misma probabilidad. Para algunas distribuciones los grados de libertad son un parámetro indispensable, ya que las definen.

Suponga usted que tiene 6 sobrinos, a los que a cada uno les regala un marcador de diferente color para evitar confusiones. Usted tiene marcadores de color negro, rosa, azul, amarillo, rojo y verde. Sus sobrinos escogerán en diversos momentos el marcador que más les guste. El sobrino 1 teniendo 6 marcadores para escoger, se quedará con el verde; el sobrino 2, con 5 posibles elecciones, tomará el rosa; el sobrino 3, con 4 opciones para elegir, optará por el negro; el sobrino 4 contará con 3 marcadores de donde seleccionar y lo hará por el azul; el sobrino 5 escogió el amarillo entre las últimas dos opciones que quedaban, lo que hace que el sobrino 6 y último no tenga elección y se quede con el último marcador que nadie tomó, es decir, el rojo. En el momento en que

el sobrino 5 realizó su elección, supimos automáticamente el marcador que le tocaría al sobrino. En este ejemplo, los grados de libertad que se tienen son 5, que fueron los que tuvieron la probabilidad de ser elegidos.

Distribución chi-cuadrada (χ^2)

La distribución χ^2 es una distribución común en ciencias de la salud, especialmente en pruebas de hipótesis y en la estimación de la varianza de una población. Es la suma de los cuadrados de (k) variables aleatorias normales estándar independientes, donde (k) es el número de grados de libertad.

Cálculo

Dada una variable aleatoria y que sigue una distribución normal, con media μ y varianza σ^2 cada valor puede transformarse en la variable normal estándar z por medio de la siguiente fórmula:

$$z = \frac{y_i - \mu}{\sigma}$$

Cada valor de z puede elevarse al cuadrado para obtener z^2 . Cuando se estudia la distribución muestra de z^2 , se observa que sigue una distribución χ^2 con 1 grado de libertad. Es decir:

$$X^2_{(1)} = \left(\frac{y_i - \mu}{\sigma} \right)^2 = z^2$$

Si todos los datos aleatorios de una muestra se elevan al cuadrado, se obtiene:

$$X^2_{(n)} = z^2_1 + z^2_2 + \dots + z^2_n$$

Es la suma de los valores z^2 resultantes tendrá una distribución X^2 , con n grados de libertad. Después de realizar una integral, la fórmula matemática de la distribución X^2 es la siguiente:

$$f(u) = \frac{1}{(\frac{k}{2} - 1)!} \frac{1}{2^{k/2}} u^{(k/2) - 1} e^{- (u/2)}, u > 0$$

$$f(u) = \frac{1}{(\frac{k}{2} - 1)!} \frac{1}{2^{k/2}} u^{(k/2) - 1} e^{- (u/2)}, u > 0$$

Donde:

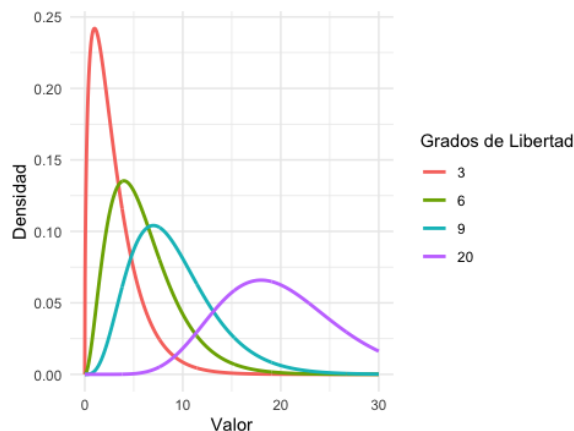
- ♦ e es el número irracional 2.71828
- ♦ k es el numero de grados de libertad
- ♦ u es x

Características de la distribución

- ♦ Es una distribución que contiene solo valores positivos.
- ♦ No sigue una forma de campana.
- ♦ Está formada por cada uno de los valores de Z de datos aleatorios elevada al cuadrado.
- ♦ Se describe por una función exponencial.
- ♦ Tiene n grados de libertad.

En la [figura 12.1](#), podemos observar una distribución χ^2 con 3, 6, 9 y 20 grados de libertad. La distribución X^2 en realidad es una familia de distribuciones.

FIGURA 12.1. DISTRIBUCIÓN CHI-CUADRADA



Usos en Bioestadística

La distribución χ^2 se utiliza para evaluar la variabilidad observada en relación con la variabilidad esperada bajo un supuesto modelo. Es particularmente útil para:

1. **Pruebas de independencia:** (ver [capítulo 19](#))
 - » Evaluar si dos variables categóricas son independientes en una tabla de contingencia.
2. **Pruebas de bondad de ajuste:**
 - » Determinar si una distribución de datos observados se ajusta a una distribución teórica esperada.

Distribución χ^2 en R

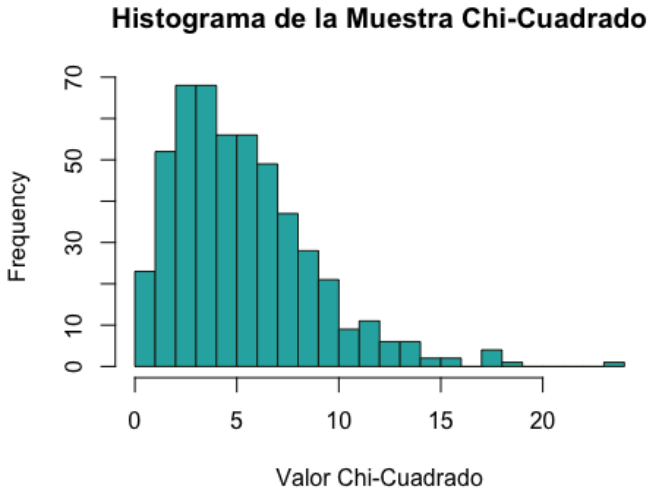
- ◇ `dchisq()`
- ◇ `pchisq()`
- ◇ `qchisq()`
- ◇ `rchisq()`
- ◇ **Ejemplo 12.8:** calcule la probabilidad de observar un valor específico de 20 en una distribución X^2 con 8 grados de libertad:

```
k_8 ← 8
prob_20 ← dchisq(20, df = k_8)
prob_20
[1] 0.003783327
```

- ◇ **Ejemplo 12.9:** genere una muestra de 500 observaciones de una distribución chi-cuadrado con 5 grados de libertad y visualice los resultados en un histograma:

```
k_5 ← 5
set.seed(456)
muestra_chi2_2 ← rchisq(n = 500, df = k_5)
```

```
hist(muestra_chi2_2, breaks = 30, col = '#28AFB0', main = 'Histograma de
la Muestra Chi-Cuadrado', xlab = 'Valor Chi-Cuadrado', border = '#1F271B')
```



Distribución t de Student

La distribución t de Student es una distribución muy parecida a la distribución normal de probabilidad continua, que surge cuando se estima la media de una población normalmente distribuida en situaciones donde el tamaño de la muestra es pequeño y la desviación estándar de la población es desconocida. Esta distribución es una de las más utilizadas en las pruebas de hipótesis (ver [capítulo 16](#)). Algunas de sus características son:

- ♦ Tiene media de 0.
- ♦ Es simétrica con respecto a la media.
- ♦ Varianza mayor que 1, aunque tiene a 1 cuando la n aumenta.
- ♦ Puede tomar valores de $-\infty$ a ∞ .
- ♦ Es una familia de distribuciones. Hay una distribución para cada valor de n .
- ♦ Es menos espigada y mas alargada que la distribución normal.
- ♦ Se aproxima a la distribución normal a medida que se aproxima al infinito.

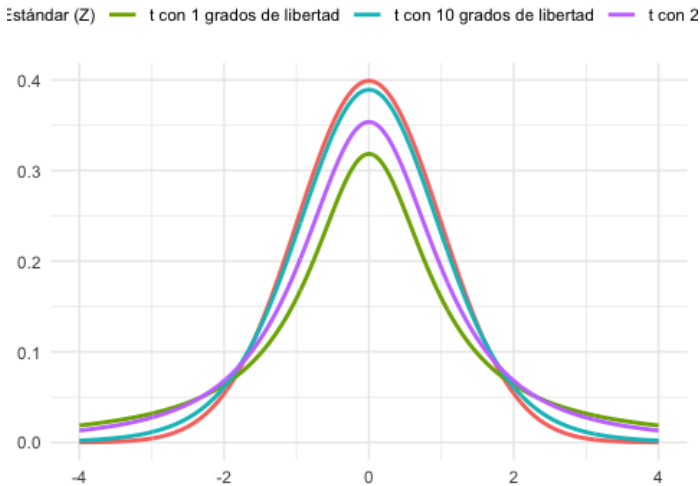
Cálculo

$$f(t;v) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})} \left(1 + \frac{t^2}{v}\right)^{-\frac{v+1}{2}}$$

Donde: t es la variable aleatoria que sigue la distribución t de Student, v es el número de grados de libertad y Γ es la función gamma.

La [figura 12.2](#) muestra la comparación de una distribución normal comparada con distribuciones t con 1, 2 y 10 grados de libertad.

FIGURA 12.2. COMPARACIÓN DE DISTRIBUCIONES T Y Z



Usos en Bioestadística

La distribución t de Student es utilizada para:

1. **Realizar pruebas de hipótesis sobre la media de una población** (ver [capítulo 15](#))
2. **Construir intervalos de confianza para la media:**
 - » Proporciona intervalos de confianza más precisos en comparación con la distribución normal cuando se trabaja con muestras pequeñas.
3. **Comparación de medias entre dos grupos** (ver [capítulo 16](#))

- » Se usa en la prueba t de Student para muestras independientes y en la prueba t pareada.

Principales características

- ♦ Valores posibles: $t \in (-\infty, \infty)$
- ♦ Grados de libertad: $k = n - 1$
- ♦ Media: 0 (para $k > 1$)
- ♦ Varianza: $\frac{k}{k-2}$ (para $k > 2$)

Distribución de Student en R

Para implementar y trabajar con la distribución t de Student en R, utilice las siguientes funciones:

- ♦ `dt()`
- ♦ `pt()`
- ♦ `qt()`
- ♦ `rt()`

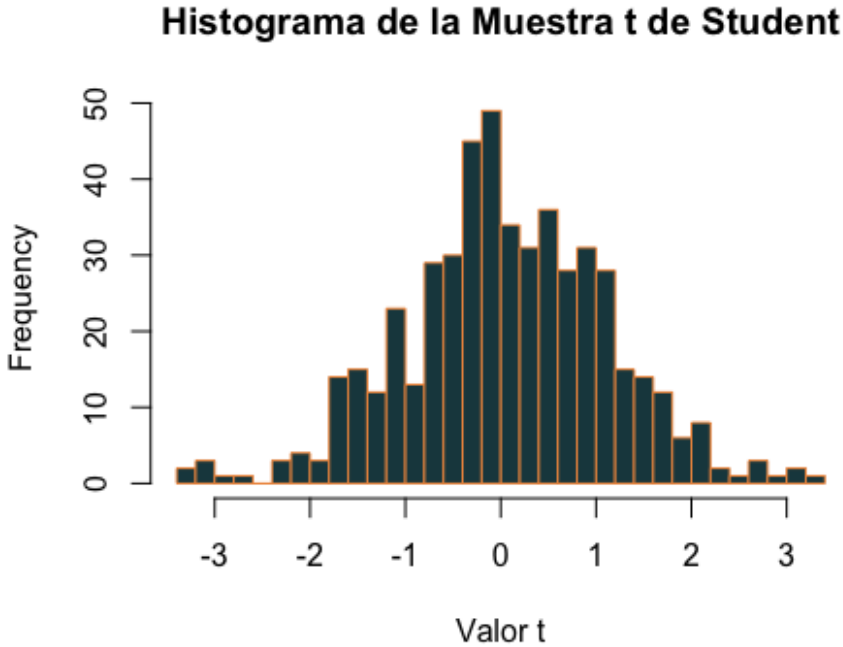
- ♦ **Ejemplo 12.10:** calcule la probabilidad de observar un valor específico de 1.5 en una distribución t de Student con 15 grados de libertad:

```
df_15 <- 15
prob_1_5 <- dt(1.5, df = df_15)
prob_1_5
[1] 0.1282609
```

- ♦ **Ejemplo 12.11:** genere una muestra de 500 observaciones de una distribución t de Student con 20 grados de libertad y represéntela en un histograma:

```
df_20 <- 20
```

```
set.seed(456)
muestra_t_2 <- rt(n = 500, df = df_20)
hist(muestra_t_2, breaks = 30, col = '#1C464D', main = 'Histograma de la
Muestra t de Student', xlab = 'Valor t', border = '#EE964B')
```



Distribución F

La distribución F, conocida también como Fisher-Snedecor, es utilizada en el análisis de varianzas, para evaluar si dos poblaciones tienen o no la misma varianza. Presenta las siguientes características:

- ♦ Se encuentra definida por dos parámetros de grados de libertad: d_1 (numerador) y d_2 (denominador). Estos parámetros determinan la forma de la distribución.

- ♦ La distribución F es asimétrica y sesgada a la derecha. La asimetría disminuye a medida que los grados de libertad aumentan, y la distribución se aproxima a una distribución normal.
- ♦ Solo valores positivos $F \geq 0$.

Cálculo

$$f(x|d_1, d_2) = \frac{\sqrt{\left(\frac{d_1 x}{d_1 x + d_2}\right)^{d_1} \left(1 - \frac{d_1 x}{d_1 x + d_2}\right)^{d_2}}}{xB\left(\frac{d_1}{2} + \frac{d_2}{2}\right)}$$

Usos en Bioestadística

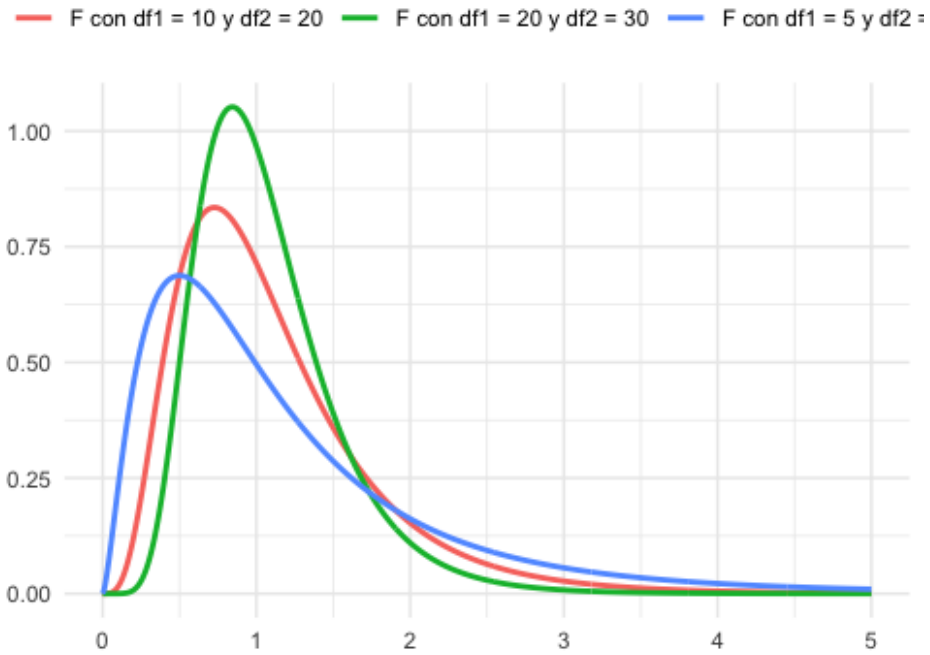
La distribución F es utilizada para:

1. **Comparar varianzas entre dos poblaciones:** evaluar si las varianzas de dos muestras independientes son significativamente diferentes.
2. **Análisis de varianza (ANOVA):** (ver [capítulo 17](#)) determinar si hay diferencias significativas entre las medias de tres o más grupos.
3. **Evaluar la significancia global de un modelo de regresión:** (ver [capítulo 22](#)) verificar si al menos una de las variables independientes en un modelo de regresión tiene un efecto significativo en la variable dependiente.

Utilice la función $\text{pf}(q, \text{df1}, \text{df2})$ para estimar valores de probabilidad y $\text{pf}(p, \text{df1}, \text{df2})$ para estimar los valores de F.

La [figura 12.3](#) muestra la comparación de una distribución F con diferentes grados de libertad.

FIGURA 12.3. DISTRIBUCIÓN F CON DIFERENTES GRADOS DE LIBERTAD



Distribución F en R

Para implementar y trabajar con la distribución t de Student en R, utilice las siguientes funciones:

- ◇ `df()`
- ◇ `pf()`
- ◇ `qf()`
- ◇ `rf()`

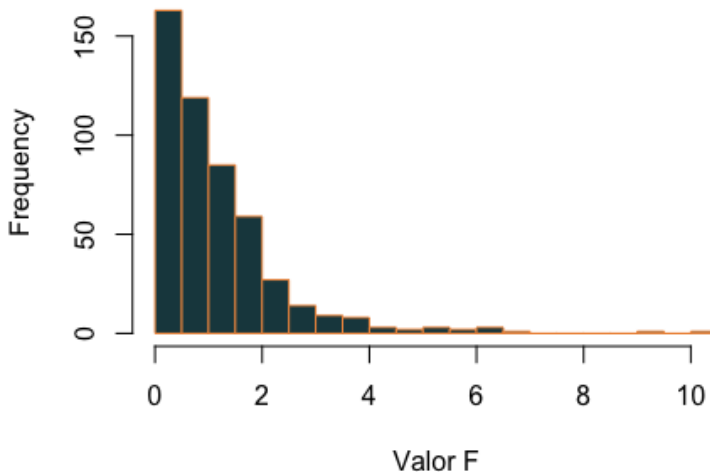
- ◇ **Ejemplo 12.12:** calcule la probabilidad de observar un valor específico de 3 en una distribución F con 10 grados de libertad en el numerador y 15 en el denominador:

```
df1_10 <- 10
df2_15 <- 15
prob_3 <- df(3, df1 = df1_10, df2 = df2_15)
prob_3
[1] 0.03530959
```

- ♦ **Ejemplo 12.13:** genere una muestra de 500 observaciones de una distribución F con 3 grados de libertad en el numerador y 12 en el denominador. Muestre sus resultados en un histograma:

```
df1_3 <- 3
df2_12 <- 12
set.seed(456)
muestra_f_2 <- rf(n = 500, df1 = df1_3, df2 = df2_12)
hist(muestra_f_2, breaks = 30, col = '#1C464D', main = 'Histograma de
la
Muestra F', xlab = 'Valor F', border = '#EE964B')
```

Histograma de la Muestra F



Ejercicios

- ♦ **Ejercicio 12.1:** imagine que está analizando el tipo de sangre de un grupo de 30 pacientes, sabiendo que hay 4 tipos de sangre posibles (A, B, AB, O).
 - a) Calcule la probabilidad de que un paciente tenga el tipo de sangre “AB”.
 - b) Genere una muestra aleatoria de 30 pacientes y visualice los resultados en un gráfico de barras.

Resolución ejercicio 12.1

```
valores <- c("A", "B", "AB", "O")
k <- length(valores)
probabilidad <- rep(1/k, k)
names(probabilidad) <- valores
probabilidad["AB"]

set.seed(123)
muestra <- sample(valores, size = 30, replace = TRUE)
barplot(table(muestra), col = '#1C2319', main = 'Distribución de Tipos
de Sangre')
```

- ♦ **Ejercicio 12.2:** un nuevo tratamiento tiene una probabilidad de éxito del 75%.
 - a) Calcule la probabilidad de fracaso del tratamiento.
 - b) Genere una muestra de 50 tratamientos y visualice los resultados en un gráfico de barras.

Resolución ejercicio 12.2

```
p <- 0.75
fracaso <- 1 - p
fracaso

set.seed(456)
muestra_bernoulli <- rbinom(n = 50, size = 1, prob = p)
barplot(table(muestra_bernoulli), col = '#1F271B', main = 'Resultados
del Tratamiento', names.arg = c("Fracaso", "Éxito"))
```

- ♦ **Ejercicio 12.3:** en un ensayo clínico, 15 pacientes reciben un tratamiento con una probabilidad de éxito del 60%.
- Calcule la probabilidad de observar exactamente 10 éxitos.
 - Genere una muestra de 1000 ensayos y visualice los resultados en un histograma.

Resolución ejercicio 12.3

```
n ← 15
p ← 0.6
prob_10 ← dbinom(10, size = n, prob = p)
prob_10

set.seed(789)
muestra_binomial ← rbinom(n = 1000, size = n, prob = p)
hist(muestra_binomial, breaks = n, col = '#1C464D', main = 'Resultados
del Ensayo Clínico', xlab = 'Número de Éxitos')
```

- ♦ **Ejercicio 12.4:** el número promedio de pacientes que llegan a una sala de emergencias es de 4 por hora.
- Calcule la probabilidad de que lleguen exactamente 5 pacientes en una hora.
 - Genere una muestra de 500 horas y visualice los resultados en un histograma.

Resolución ejercicio 12.4

```
lambda ← 4
prob_5 ← dpois(5, lambda)
prob_5

set.seed(321)
muestra_poisson ← rpois(n = 500, lambda = lambda)
hist(muestra_poisson, breaks = max(muestra_poisson), col = '#19647E',
main = 'Llegadas por Hora', xlab = 'Número de Llegadas')
```

- ♦ **Ejercicio 12.5:** usted está evaluando la independencia entre dos variables categóricas con una tabla de contingencia 3×3 . Los grados de libertad son 4.
 - a) Calcule la probabilidad de observar un valor específico de 10.
 - b) Genere una muestra de 1000 observaciones y visualice los resultados en un histograma.

Resolución ejercicio 12.5

```
df <- 4 # para una distribución de 3X3
prob_10 <- dchisq(10, df = df)
prob_10
set.seed(654)
muestra_chi2 <- rchisq(n = 1000, df = df)
hist(muestra_chi2, breaks = 30, col = '#28AFB0', main = 'Distribución
Chi-Cuadrado', xlab = 'Valor Chi-Cuadrado')
```

- ♦ **Ejercicio 12.6:** un investigador quiere comparar las medias de dos grupos con 15 grados de libertad.
 - a) Calcule la probabilidad de observar un valor específico de 1.2.
 - b) Genere una muestra de 500 observaciones y visualice los resultados en un histograma.

Resolución ejercicio 12.6

```
df <- 15
prob_1_2 <- dt(1.2, df = df)
prob_1_2

set.seed(987)
muestra_t <- rt(n = 500, df = df)
hist(muestra_t, breaks = 30, col = '#8EC187', main = 'Distribución t de
Student', xlab = 'Valor t')
```


- ♦ **Ejercicio 12.7:** en un análisis de varianza (ANOVA) con 5 grados de libertad en el numerador y 10 en el denominador.
- Calcule la probabilidad de observar un valor específico de 3.
 - Genere una muestra de 1000 observaciones y visualice los resultados en un histograma.

Resolución ejercicio 12.7

```
df1 ← 5
df2 ← 10
prob_3 ← df(3, df1 = df1, df2 = df2)
prob_3
set.seed(111)
muestra_f ← rf(n = 1000, df1 = df1, df2 = df2)
hist(muestra_f, breaks = 30, col = '#F4D35E', main = 'Distribución F',
xlab = 'Valor F', border = '#F0A05B')
```

Referencias

- Dai, H., Wang, W., Xu, Q., Xiong, Y. y Wei, D. Q. (2019). Estimation of Probability Distribution and Its Application in Bayesian Classification and Maximum Likelihood Regression. *Interdisciplinary sciences, computational life sciences*, 11(3), 559-574. <https://doi.org/10.1007/s12539-019-00343-w>
- Hooft, R. W., Straver, L. H. y Spek, A. L. (2009). Probability plots based on Student's t-distribution. *Acta crystallographica. Section A, Foundations of crystallography*, 65(Pt 4), 319-321. <https://doi.org/10.1107/S0108767309009908>
- Indrayan, A. y Malhotra, R. K. (2017). Confidence Intervals, Principles of Tests of Significance, and Sample Size. En *Medical Biostatistics* (4ª ed., pp. 255-304). Chapman & Hall/CRC Biostatistics Series.
- Kim, H. Y. (2019). Statistical notes for clinical researchers: the independent samples t-test. *Restorative dentistry & endodontics*, 44(3), e26. <https://doi.org/10.5395/rde.2019.44.e26>
- McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica*, 23(2), 143-149. <https://doi.org/10.11613/bm.2013.018>

- Moscarelli, M. (2023). Data Distributions. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 55-76). Springer Cham/Taylor & Francis Group, LLC.
- Riffenburgh, R. H. (2005). *Statistics in Medicine* (2nd ed., pp. 15-35). Elsevier.
- Schober, P. y Vetter, T. R. (2019). Chi-square Tests in Medical Research. *Anesthesia and analgesia*, 129(5), 1193. <https://doi.org/10.1213/ANE.0000000000004410>
- Shahbaba, B. (2011). Random Variables and Probability Distributions. En *Biostatistics with R: An Introduction to Statistics Through Biological Data* (pp. 109-149). Springer. <https://doi.org/10.1007/978-1-4614-1302-8>
- Viti, A., Terzi, A. y Bertolaccini, L. (2015). A practical overview on probability distributions. *Journal of thoracic disease*, 7(3), E7–E10. <https://doi.org/10.3978/j.issn.2072-1439.2015.01.37>
- Wadhwa, R. R. y Marappa-Ganeshan, R. (2023). *T Test*. StatPearls Publishing.
- Wampler, J. E. (1990). Analysis of the probability distribution of small random samples by nonlinear fitting of integrated probabilities. *Analytical biochemistry*, 186(2), 209-218. [https://doi.org/10.1016/0003-2697\(90\)90068-k](https://doi.org/10.1016/0003-2697(90)90068-k)
- Wilson, E. B. y Hilferty, M. M. (1931). The Distribution of Chi-Square. *Proceedings of the National Academy of Sciences of the United States of America*, 17(12), 684-688. <https://doi.org/10.1073/pnas.17.12.684>

Distribución normal

Edsaúl Emilio Pérez Guerrero

En este capítulo se tocan los aspectos teóricos básicos sobre la distribución normal y cómo evaluar e interpretar si un conjunto de datos sigue una distribución normal o no.

Conocer si mis datos siguen una distribución normal es importante por las siguientes razones:

- ♦ **Presentación de mis resultados:** la presentación de mis resultados muchas veces depende de la distribución de mis datos. Si mis datos siguen una distribución normal, se suele utilizar medias y desviación estándar para presentar los datos, ya que estos describen mejor la distribución de mis datos. Por el contrario, si mis datos no se apegan a una distribución normal, se sugiere presentarlos utilizando medianas y una medida de dispersión como rangos o el IQR, ya que de esta manera podremos dar a conocer mejor las características de estos datos.
- ♦ **Selección de pruebas estadísticas:** muchas pruebas estadísticas, conocidas como pruebas paramétricas, necesitan que los datos sigan el comportamiento de una distribución normal para poder ser válidas. Pruebas estadísticas, como la *t* de Student, el ANOVA y las pruebas de correlación de Pearson, asumen que los datos siguen una distribución normal. Si los datos no siguen una distribución normal, pueden requerirse pruebas no paramétricas, como la prueba de Mann-Whitney, la prueba de Kruskal-Wallis o la correlación de Spearman, que no hacen esta suposición y son más apropiadas para datos no normales.

- ♦ **Interpretación:** la normalidad de los datos afecta la interpretación de los intervalos de confianza y los valores . En datos normalmente distribuidos, los intervalos de confianza y las inferencias estadísticas son más precisas y confiables.
- ♦ **Modelos estadísticos:** algunos modelos estadísticos, como la regresión lineal, requieren que los errores (residuos) estimados en el modelo sigan una distribución normal para confirmar la validez de los modelos.

Comenzaremos este capítulo explicando algunos de los aspectos teóricos de la distribución normal y, posteriormente, revisaremos los gráficos y las pruebas estadísticas que nos permiten la evaluación de la misma.

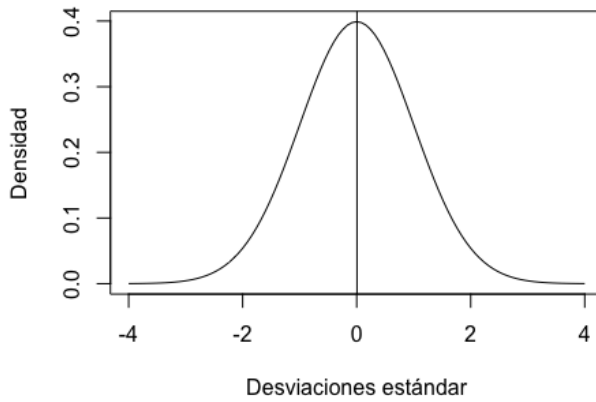
Características de la distribución normal

La distribución normal es probablemente el tipo de distribución de probabilidad más conocido. En esta, cuando los datos son graficados, presenta forma de campana simétrica y con cierta altura (ni muy alta ni muy ancha). Para esta distribución, las desviaciones estándar son sumamente importantes, ya que permiten establecer valores de referencia, y con ello poder conocer dónde se encuentran el 99%, 95% y 68% de los datos.

Algunas de las características más importantes de la distribución normal son:

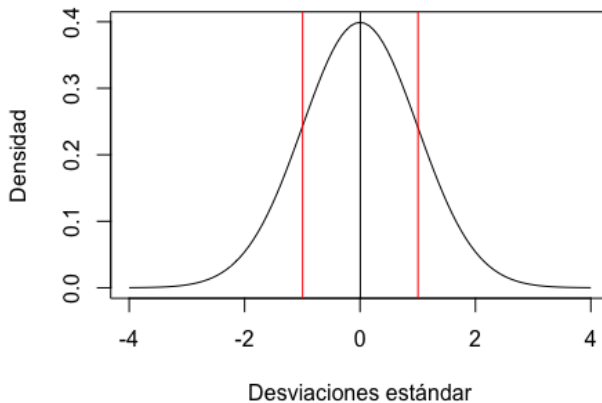
1. El área total que se encuentra por debajo de la curva es igual a 1.
2. Su distribución es simétrica y la media divide en dos partes iguales.
3. Cuando la variable es discreta, la media, la mediana y la moda son iguales.
4. Existe una distribución diferente por cada valor de μ y de s .
5. Puede tomar valores de ∞ a $-\infty$.

La [figura 13.1](#) muestra una distribución normal típica.

FIGURA 13.1. DISTRIBUCIÓN NORMAL

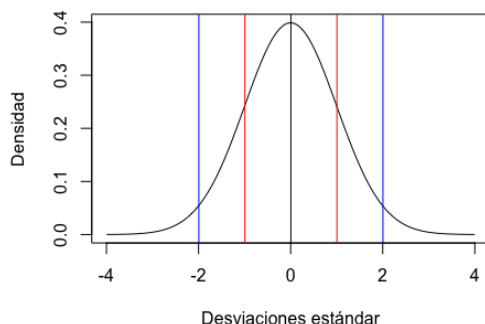
En la [figura 13.2](#), la línea vertical negra representa el valor de media para el conjunto de datos graficado. Podemos notar que en una distribución normal, la media se encuentra en el centro. Además, en esta distribución la media, la mediana y la moda en el mejor de los casos son el mismo valor.

Una característica de la distribución normal es que si restamos y sumamos una desviación estándar a la media, tendremos el 68% de los datos. En la [figura 13.3](#), las líneas rojas representan el valor de restar o sumar una desviación estándar a la media del conjunto de datos.

FIGURA 13.2. DISTRIBUCIÓN NORMAL +/- 1 DESVIACIÓN ESTÁNDAR

Ahora, si restamos o sumamos a la media dos desviaciones estándar tendremos el 95% de los datos, que en el caso de la [figura 13.3](#) están representados con líneas azules.

FIGURA 13.3. DISTRIBUCIÓN NORMAL ± 2 DESVIACIÓN ESTÁNDAR

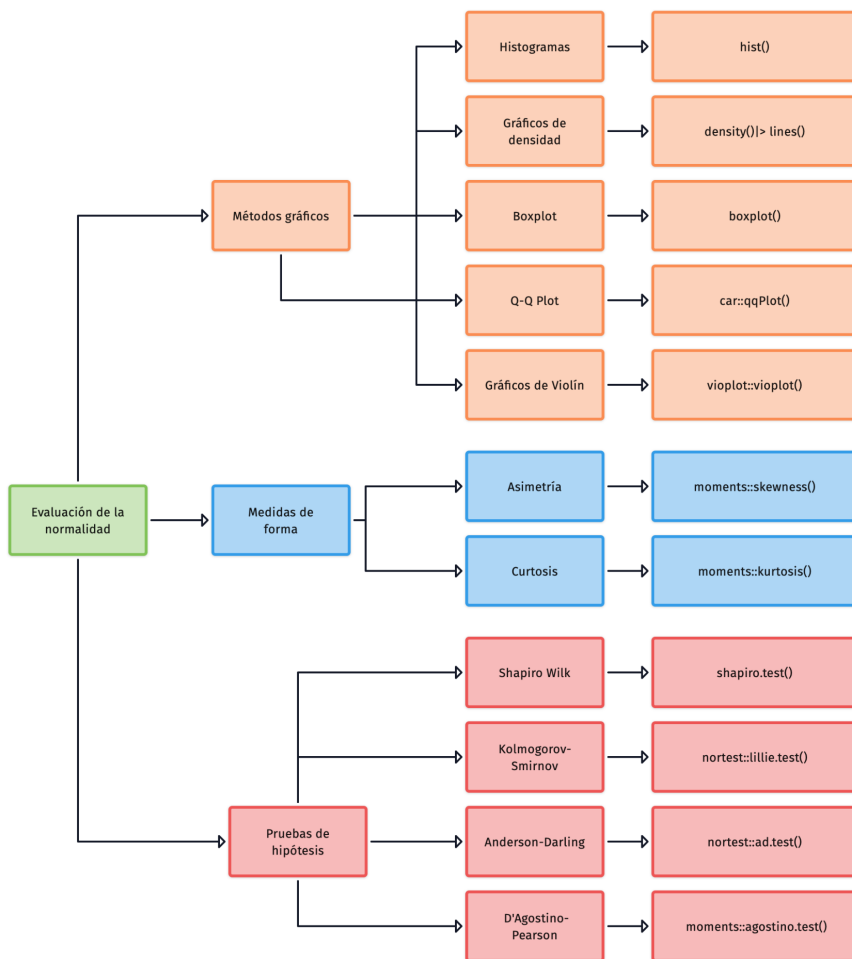


Evaluación de la normalidad

Para evaluar la normalidad de los datos existen diversas alternativas, como lo son: gráficos, medidas de forma y pruebas de hipótesis. En la medida de lo posible utilice todas. Entienda sus datos.

La [figura 13.4](#) muestra los criterios que los autores de este libro consideran indispensables para evaluar la normalidad. En esta figura, al final de la derecha se muestra la función utilizada en R para cada criterio.

FIGURA 13.4. EVALUACIÓN DE NORMALIDAD



Aunque la opción más rápida para evaluar la normalidad es utilizar una prueba de hipótesis (en el [capítulo 14](#) y en capítulos posteriores se describe con más detalle a qué nos referimos con una prueba de hipótesis), mi recomendación es que se utilicen todas las herramientas disponibles para tomar la decisión de si un conjunto de datos sigue o no una distribución normal. Recuerde entender sus datos y que en R las gráficas son gratis.

En las siguientes secciones vamos a desglosar y explicar cada uno de los puntos de la [figura 13.4](#).

Evaluación gráfica

La evaluación gráfica de la normalidad nos permite identificar cómo los datos de una variable aleatoria numérica se distribuyen. Las gráficas que revisaremos a continuación son una guía para conocer si los datos tienen un comportamiento normal o no. Sin embargo, mi recomendación es que nunca se vean de forma aislada y siempre se tomen en cuenta otros factores, como las pruebas de hipótesis, la forma de la campana, medidas de tendencia central y dispersión. Aquí solo será descrita la evaluación gráfica para histogramas, gráficos de densidad (en conjunto con histogramas), boxplot y gráficos Q-Q. Dado que los gráficos de violín son gráficas compuestas por gráficos de densidad y boxplot, la interpretación de los mismos se puede inferir de los otros gráficos.

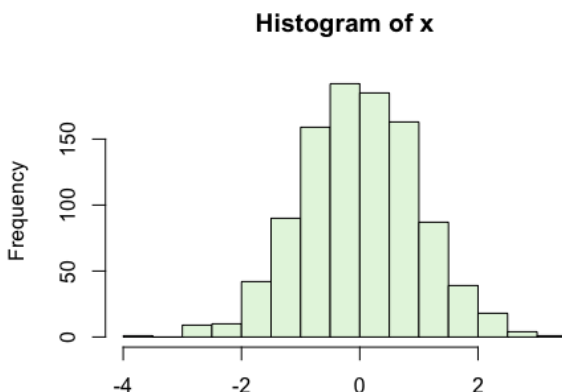
Histogramas

Aunque la perspectiva de un histograma puede cambiar según el número de intervalos de clase que lo conformen, en ciertas ocasiones puede ser una guía sobre el comportamiento de los datos. Para ejemplificarlo, primero vamos a crear un objeto con datos aleatorios de una distribución normal y después lo vamos a graficar utilizando la función `hist`.

```
x ← rnorm(n=1000, mean = 0, sd= 1)
```

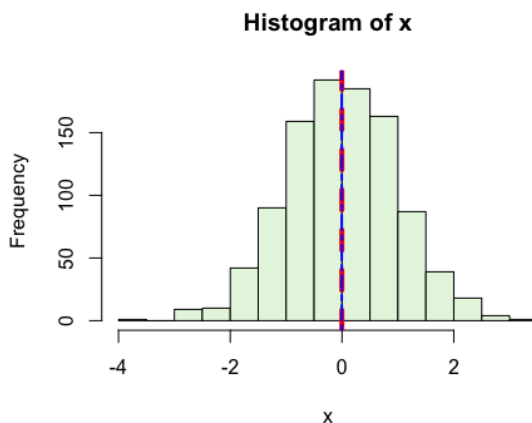
El código anterior creó un objeto (`x`) con 1000 datos aleatorios de una distribución normal con media de 0 y desviación estándar de 1. Si lo graficamos mediante un histograma obtenemos:

```
hist(x, col = "#e5f5e0")
```

Podemos ver que el histograma tiene una distribución simétrica, en la que la mayoría de los datos se concentran en el centro. La forma del histograma anterior es muy similar a la de la distribución normal. Al gráfico anterior le podemos agregar una línea que muestra dónde se encuentra la media y dónde se encuentra la mediana. Recuerde que la media y la mediana, en el caso de la distribución normal, deben ser iguales o tener un valor muy similar y encontrarse en el centro de la distribución.

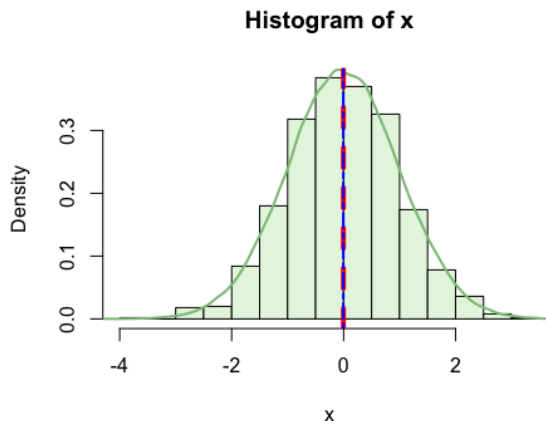
```
hist(x, col="#e5f5e0")
abline(v=mean(x), col="red", lty = 2, lwd = 4) # Línea vertical para la media
abline(v=median(x), col="blue", lty= 6, lwd = 2) # Línea vertical para mediana
```



La función `abline` permite añadir líneas a una gráfica ya creada. En el código anterior utilizamos el argumento `v` para indicarle que la línea que queremos añadir es vertical (de ahí la `v`), y para indicarle el sitio donde esta debe de ser añadida, utilizamos el valor de la media o de la mediana, aunque también puede utilizar un número como 1, 2, etc. Cambiamos el color, el tipo y el tamaño de la línea con los argumentos `col`, `lty` y `lwd`, respectivamente. Puede encontrar un poco más sobre los tipos de líneas [aquí](#), [aquí](#) y más contenido sobre la función `abline` [aquí](#). Tome en cuenta que la función `abline` no puede añadir líneas si antes no creamos una gráfica, y que para poder borrar una línea debemos de correr de nuevo la gráfica base. No olvide que las gráficas se construyen en forma de capas.

A la gráfica anterior podríamos agregarle una gráfica de densidad para hacer todavía más visible la forma que tiene la distribución:

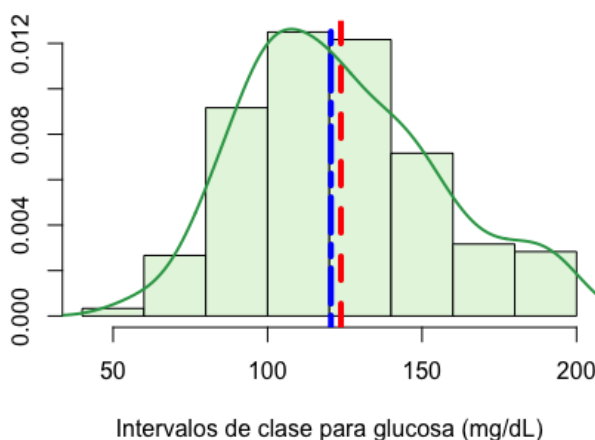
```
hist(x, col="#e5f5e0",
     prob=T, # Graficar densidades en lugar de frecuencias
)
# Agregar una linea de densidad para un conjunto de datos normales
lines(density(rnorm(100000)),
      lwd=2,
      col="#8EC187")
abline(v=mean(x), col="red", lty = 2, lwd = 4) # Línea vertical para la media
abline(v=median(x), col="blue", lty= 6, lwd = 2) # Línea vertical para mediana
```



Como podemos notar, la evaluación de la normalidad utilizando histogramas requiere de diversas características extras que nos permitan evaluarla adecuadamente. Las figuras anteriores se crearon utilizando un objeto con datos aleatorios normales. Por ello, la forma de la gráfica es perfecta. La [figura 13.5](#) muestra un caso de datos que no se apegan a una distribución normal perfecta.

FIGURA 13.5. HISTOGRAMA DE LAS CONCENTRACIONES DE GLUCOSA PARA UN GRUPO DE PACIENTES

Concentraciones de glucosa de un grupo de pacien



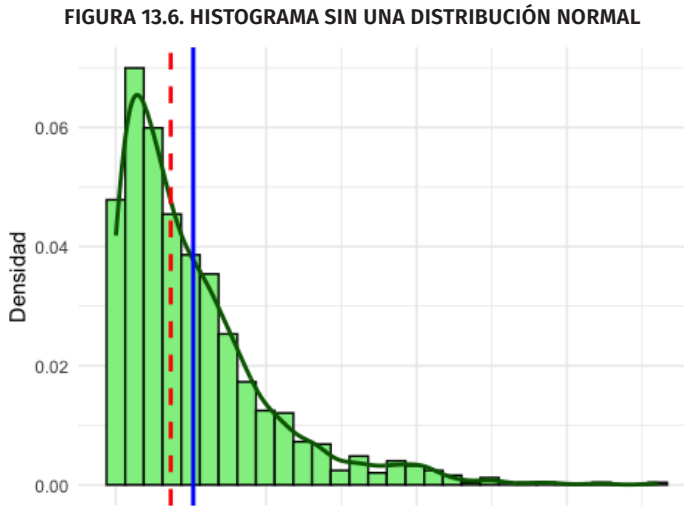
La [figura 13.5](#) muestra características que no son del todo compatibles con la distribución normal. Por ejemplo, la media y la mediana no coinciden (aunque en la realidad esto difícilmente ocurre). En una distribución normal, la media y la mediana deberían coincidir o estar muy cercanas. Por otro lado, los datos más representativos para evaluar la normalidad con la gráfica anterior se presentan en los extremos. La barra del extremo izquierdo es mucho más pequeña que la barra de la derecha. Esto indica que hay muy poca cantidad de datos cercanos a 50 mg/dL de glucosa (lo cual es lógico) comparado con la barra de la extrema derecha. En una distribución normal, ambas barras deberían ser de la misma altura. Además, las barras de la parte extrema derecha indican una cantidad importante de datos a concentraciones superiores a los 150 mg/dL. Por último, si observamos con más detenimiento, podemos ver que tanto

el histograma como la línea de densidad no son simétricas, sino que están más cargadas hacia la derecha (cola larga a la derecha). Esto indica que la dispersión es mayor en los valores más grandes que la media, comparado con los valores más pequeños que la media.

En resumen, en un histograma como el de la [figura 13.5](#) debemos de:

- ♦ Evaluar las alturas de las barras (sobre todo en los extremos).
- ♦ Verificar que la media y la mediana coincidan.
- ♦ Evaluar la simetría de las barras y de la línea de densidad.

La [figura 13.6](#) muestra otro histograma (más evidente) sobre unos datos que no siguen una distribución normal.

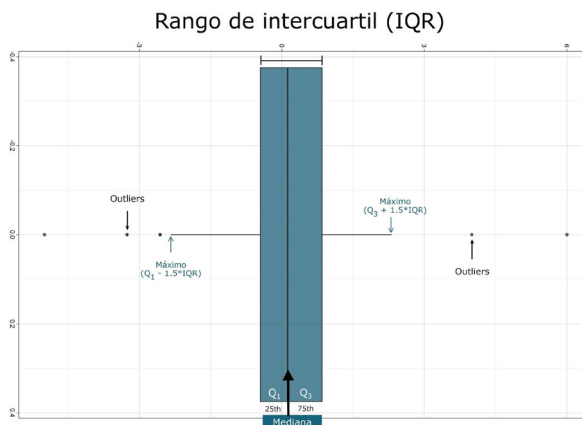


Boxplot

Los boxplot son una de las gráficas más útiles para evaluar la normalidad, ya que proporciona datos sobre la dispersión, asimetría y la presencia de valores atípicos. Aunque ya se había utilizado con anterioridad este gráfico, no había sido explicado en su totalidad. Vamos, pues, a explicar las características que conforman a un boxplot (ver [figura 13.7](#)).

- ◇ Caja (rectángulo):
 - » Percentil 25 (Q_1): el borde inferior de la caja representa el primer cuartil, que es el valor por debajo del cual se encuentra el 25% de los datos.
 - » Mediana (percentil 50, Q_2): la línea dentro de la caja indica la mediana de los datos, es decir, el valor por debajo del cual se encuentra el 50% de los datos.
 - » Percentil 75 (Q_3): el borde superior de la caja representa el tercer cuartil, que es el valor por debajo del cual se encuentra el 75% de los datos.
- ◇ Bigotes (*whiskers*):
 - » Los bigotes se extienden desde los bordes de la caja hasta el valor más grande y más pequeño dentro de 1.5 veces el rango intercuartílico (IQR). El IQR se calcula como $Q_3 - Q_1$.
 - » Bigote inferior: se extiende desde el percentil 25 hasta el valor mínimo dentro de 1.5 veces el IQR.
 - » Bigote superior: se extiende desde el percentil 75 hasta el valor máximo dentro de 1.5 veces el IQR.
- ◇ Valores atípicos (*outliers*):
 - » Los puntos que se encuentran fuera de los bigotes son considerados *outliers*. Estos valores son más pequeños que $Q_1 - 1.5 * \text{IQR}$ o más grandes que $Q_3 + 1.5 * \text{IQR}$. En la medida de lo posible no oculte los valores atípicos, ya sea cambiando los límites de los ejes o alargando los bigotes.
- ◇ Los *outliers* se muestran típicamente como puntos individuales.

FIGURA 13.7. EJEMPLO DE BOXPLOT Y SUS COMPONENTES

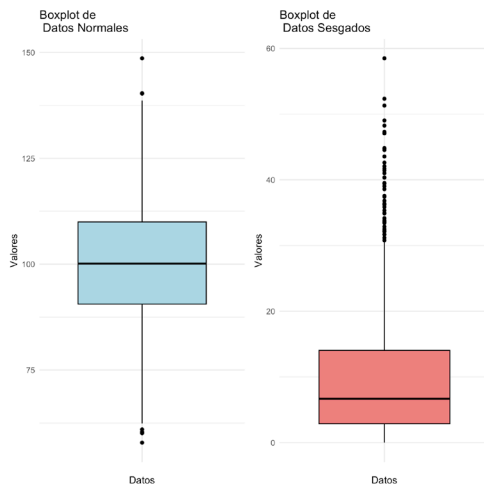


Cuando se evalúa la normalidad mediante un boxplot se debe prestar atención en lo siguiente:

- ❖ La simetría de la caja. En una distribución normal la caja debe de ser simétrica, es decir, los cuartiles deben de estar a la misma distancia de la mediana.
- ❖ Los bigotes deben ser aproximadamente iguales en longitud en una distribución normal.
- ❖ Debe de haber pocos o ningún outlier.

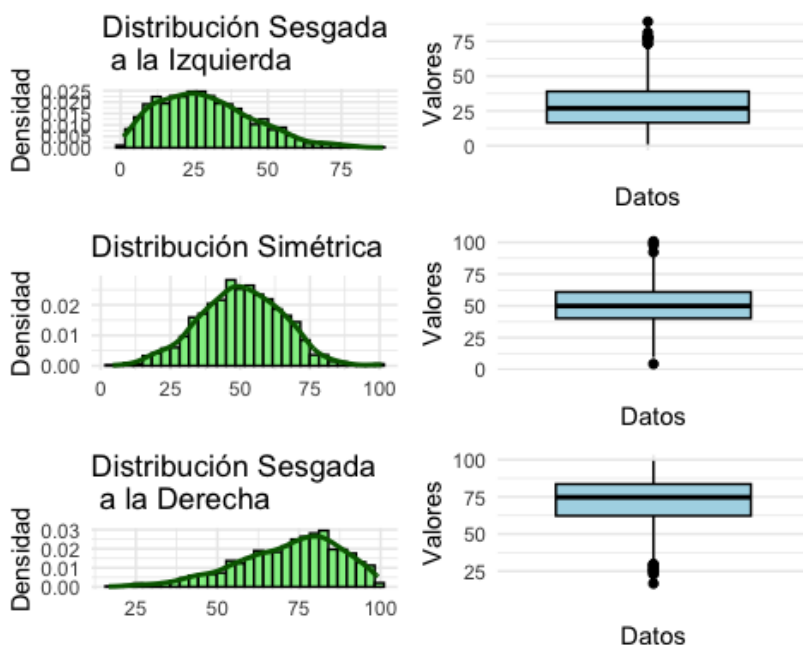
La [figura 13.8](#) muestra dos boxplot, uno para un conjunto de datos con una distribución normal (izquierda) y sin una distribución normal. Note cómo en la figura de la derecha la caja no es simétrica, hay una cantidad importante de *outliers* y uno de los bigotes es más amplio que otro. Este boxplot muestra que, a valores más grandes, por encima del Q3 la dispersión aumenta considerablemente. Los datos están más compactados en los valores pequeños.

FIGURA 13.8. BOXPLOT PARA EVALUAR LA NORMALIDAD



La [figura 13.9](#) muestra la falta de simetría (asimetría) de un grupo de datos y cómo esta modifica la forma de un histograma (parte izquierda de la gráfica) y los boxplot (parte derecha). Visualice la dispersión y luego compárela en uno y otro gráfico, identifique dónde se concentran la mayoría de los datos.

FIGURA 13.9. COMPARACIÓN DE LA ASIMETRÍA EN HISTOGRAMAS Y BOXPLOT



Gráficos Q-Q

Un gráfico Q-Q (*Quantile-Quantile*) es una herramienta gráfica utilizada para comparar la distribución de un conjunto de datos con una distribución teórica, como la distribución normal. En un gráfico Q-Q, los cuantiles de los datos observados se comparan con los cuantiles de la distribución teórica. En pocas palabras, comparamos los datos teóricos de una distribución normal con nuestros valores.

El gráfico Q-Q se compone de la siguiente forma:

- ♦ **Eje de las “x”:** muestra los cuantiles teóricos de la distribución normal, aunque podría ser cualquier tipo de distribución
- ♦ **Eje de las y:** muestra los cuantiles de los datos observados.
- ♦ **Línea de referencia:** un gráfico Q-Q tiene una recta (usualmente diagonal) que representa dónde deberían caer los puntos, si los datos siguen exactamente la distribución teórica.

En un gráfico Q-Q se debe de evaluar lo siguiente:

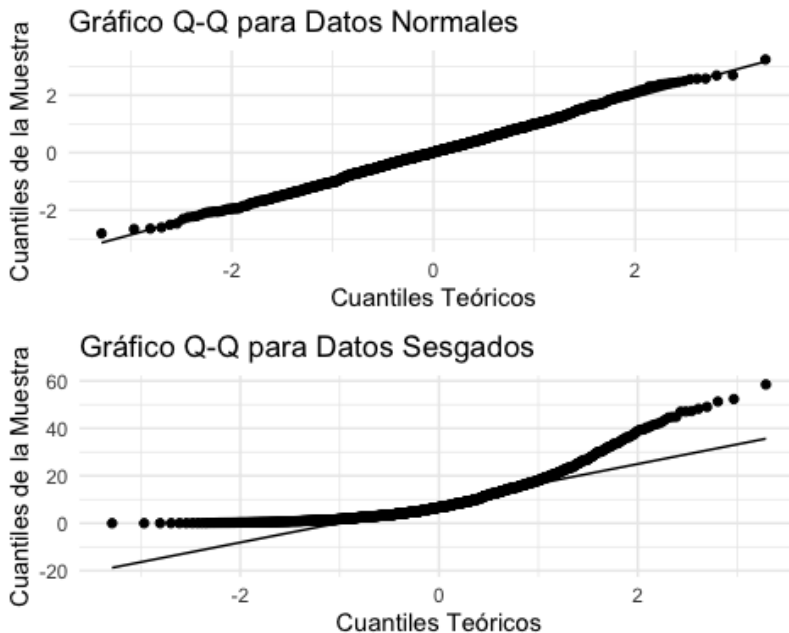
♦ **Puntos cerca de la línea de referencia:**

- » Si los puntos están cerca de la línea de referencia, sugiere que los datos siguen la distribución teórica.
- » Puntos desviados de la línea de referencia:
 - Si los puntos se desvían significativamente de la línea, indica que los datos no siguen la distribución teórica (normal en este caso).
 - Entre más alejado este un punto de la línea, más diferente es de un valor teórico.

♦ **Cola izquierda y derecha:** la forma en que los puntos se desvían en las colas del gráfico puede indicar sesgo o la presencia de *outliers*.

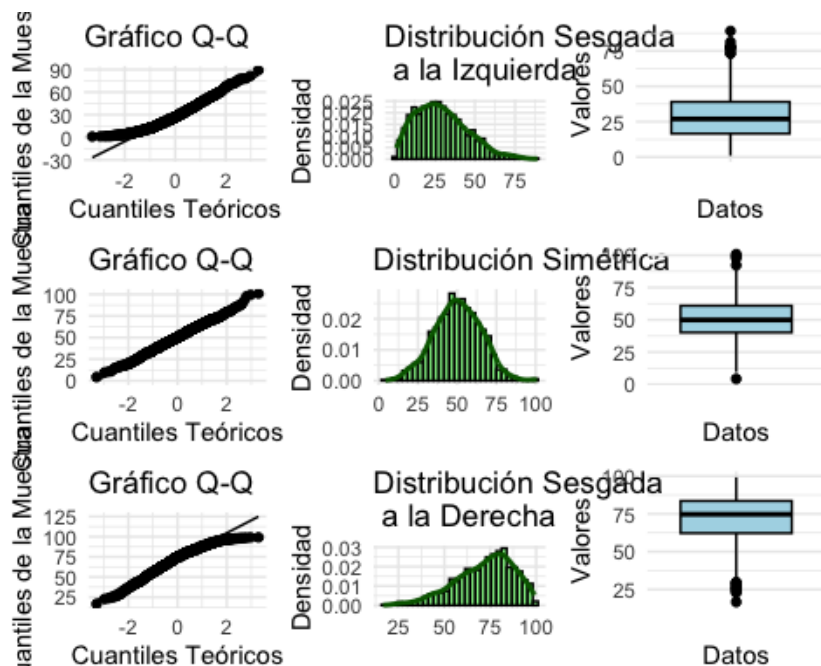
La [figura 13.10](#) muestra la comparación de un gráfico Q-Q para una serie de datos con una distribución normal (izquierda) y un gráfico Q-Q para datos que no tienen características de normalidad (derecha).

FIGURA 13.10. EVALUACIÓN DE LA NORMALIDAD UTILIZANDO GRÁFICOS Q-Q



Lo que observamos en una gráfica Q-Q tiene mucha relación con lo que observamos en un gráfico de densidad, en un histograma y en un boxplot. Esta relación se muestra en la [figura 13.11](#).

FIGURA 13.11. COMPARACIÓN DE LA ASIMETRÍA EN GRÁFICOS Q-Q, HISTOGRAMAS Y BOXPLOT

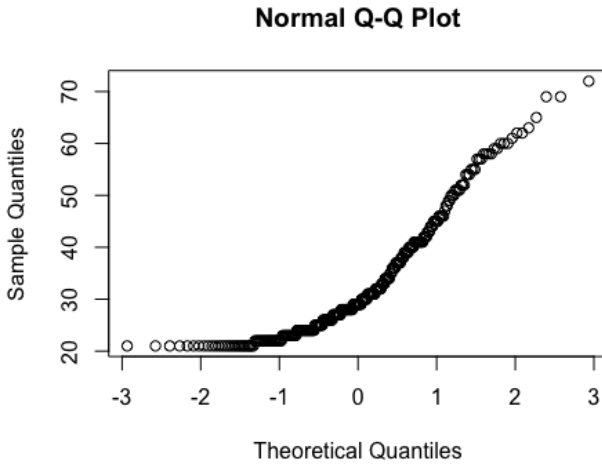


Construcción de gráficos Q-Q en R

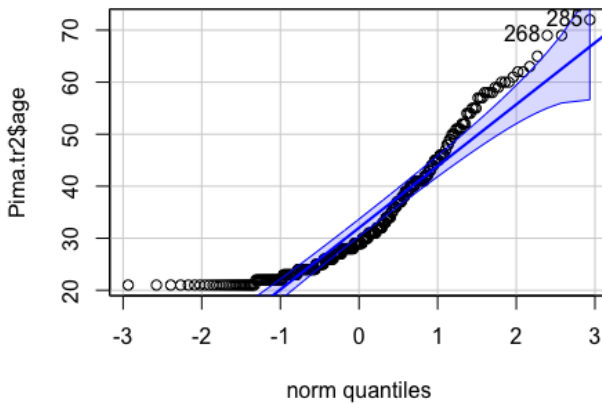
Para construir gráficos Q-Q en R se puede emplear el paquete base mediante la función `qqnorm()` que permite la comparación de nuestros datos con una distribución normal. También, se puede emplear la librería `car` que contiene la función `qqplot()`, la cual permite la construcción de un gráfico Q-Q a diferencia del primero muestra la línea de tendencia y los intervalos de confianza (que se describen en “[Intervalos de confianza](#)”). Además, `ggplot` también permite la construcción de estos gráficos. En este libro recomendamos utilizar la librería `car`.

- ♦ **Ejemplo 13.1:** construya gráficos Q-Q de la variable `edad` de `Pima.tr2` utilizando el paquete base, `car` y `ggplot`.

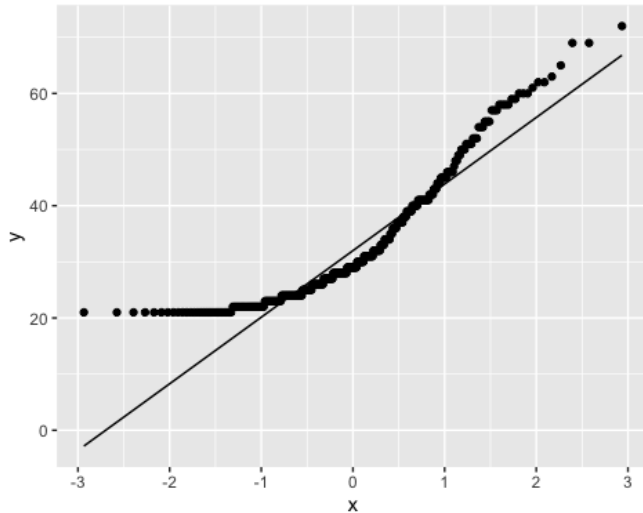
```
# Paquete base  
qqnorm(Pima.tr2$age)
```



```
install.packages("car") #Si es necesario  
car::qqPlot(Pima.tr2$age)
```



```
[1] 285 268
ggplot(Pima.tr2, aes(sample = age)) +
  stat_qq() +
  stat_qq_line()
```



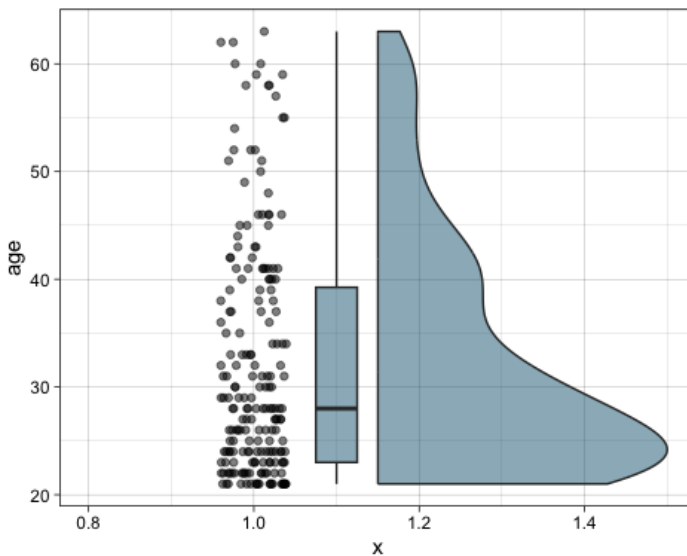
Rainclouds

Un gráfico Raincloud (Raincloud plot) es una combinación de varias técnicas de visualización de datos en una sola figura, lo que proporciona una visión completa de la distribución de los datos. Estos gráficos combinan gráficos de puntos, *bxoplot* y gráficos de densidad.

Si bien no es un gráfico diseñado para evaluar normalidad, puede darnos una idea de cómo es la distribución de mis datos. En R las librerías *ggRain* contiene una geometría para *ggplot2* que permite la creación de estos gráficos.

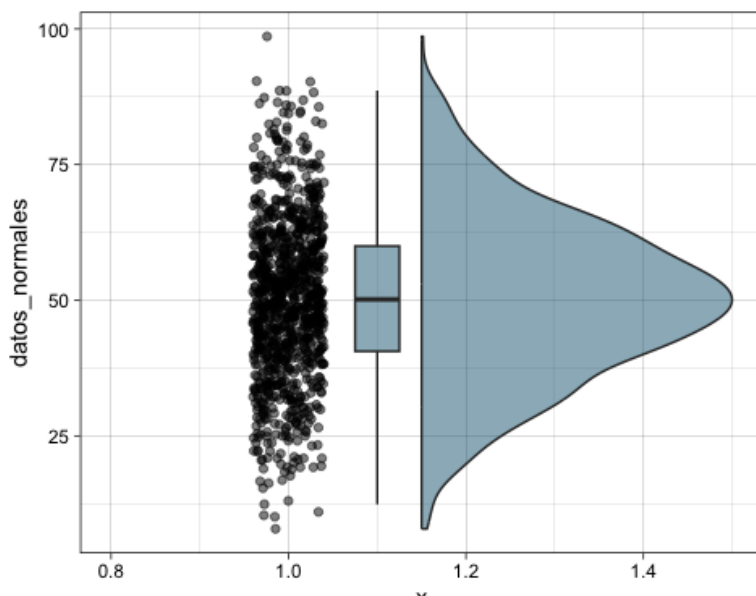
- ♦ **Ejemplo 13.2** Vamos a crear un Raincloud utilizando las edades de las mujeres de Pima utilizando el *Dataframe* `Pima.tr2` y luego uno con datos ficticios que se ajustan a una distribución normal:

```
install.packages("gggrain") # Si es necesario
library(MASS) # Si es necesario
data("Pima.tr2") # Si es necesario
library(gggrain)
Pima.tr|>
  ggplot(mapping = aes(x = 1, y = age))+
  geom_rain(fill = "#19647E", alpha=0.5)+
  theme_linedraw()
```



```
# Generar los datos
set.seed(123)
datos_normales <- rnorm(1000, mean = 50, sd = 15) # Simétrica
# Graficar
ggplot(mapping = aes(x = 1, y = datos_normales))+
  geom_rain(fill = "#19647E", alpha=0.5, )+
  theme_linedraw()
```

FIGURA 13.12. DATOS CON UNA DISTRIBUCIÓN NORMAL EN UNA GRÁFICA DE RAINCLOUDS



Medidas de forma para evaluar la normalidad

No todos los gráficos de densidad tienen la misma forma, la distribución de los datos debe ajustarse a ciertos criterios de algunos parámetros de altura (concentración de los datos) y de simetría. Vimos en la sección anterior cómo algunas de las gráficas se inclinaban un poco hacia la izquierda o hacia la derecha y que esta falta de simetría era precisamente lo que permitía intuir que los datos no seguían una distribución normal. Además, ¿qué pasaría si una gráfica de densidad tuviera una altura fuera de lo “normal” y fuera mucho más alta o mucho más baja que los datos mostrados en la [figura 13.12](#)? Mejor dicho, ¿qué pasaría si los datos se concentraran mucho en el centro? O por el contrario, ¿qué pasaría si los datos no se concentraran en el centro?

Existen dos medidas que nos permiten evaluar la simetría y la altura (concentración de datos) de los gráficos de densidad. La **asimetría** se refiere a la medida de la simetría de la distribución de los datos. Específicamente, la asimetría indica si los datos están distribuidos de manera uniforme alrededor de la media o si están

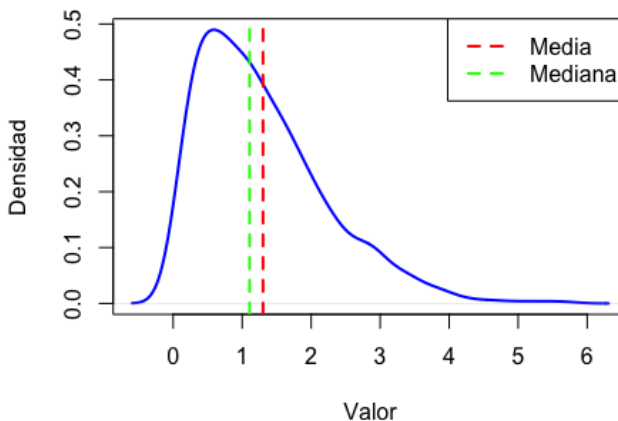
inclinados hacia un lado. Por su parte, la **kurtosis** es una medida estadística que describe la forma de las colas de una distribución de datos en relación con una distribución normal. Específicamente, la kurtosis indica la concentración de los datos en el centro de la distribución, por tal razón suele confundirse con el “pico” de la distribución (Bhattacharya *et al.*, 2022). Revisaremos a detalle estas dos medidas.

Coeficiente de asimetría

Existen dos tipos principales de asimetría:

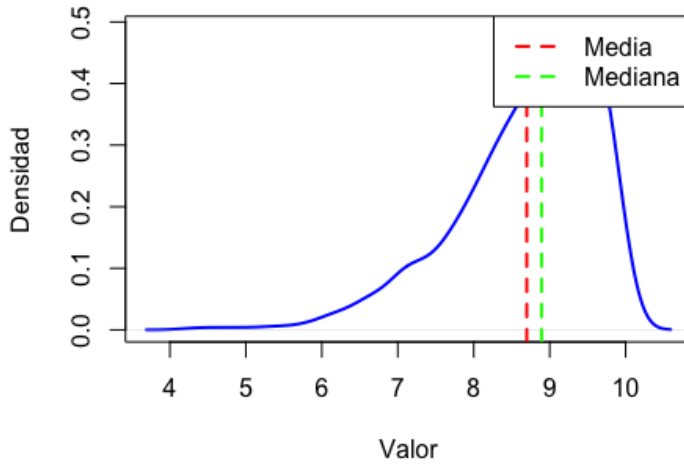
- ♦ **Asimetría positiva (o sesgo a la derecha):** en una distribución con asimetría positiva, la cola de la distribución es más larga en el lado derecho de la media. Esto significa que hay más valores extremos altos, que la dispersión aumenta hacia la derecha de la media y es más compacta hacia la izquierda (hay más datos en la parte izquierda). Además, en la asimetría positiva, la media es más grande que la mediana. La [figura 13.13](#) muestra un ejemplo de asimetría positiva.

FIGURA 13.13. ASIMETRÍA POSITIVA



- ♦ **Asimetría negativa (o sesgo a la izquierda):** en una distribución con asimetría negativa, la cola de la distribución es más larga en el lado izquierdo de la media. Esto significa que hay más valores extremos bajos y que la dispersión es menor en el lado izquierdo de la gráfica. Por lo tanto, la media será menor que la mediana. La [figura 13.14](#) muestra un ejemplo de asimetría positiva.

FIGURA 13.14. ASIMETRÍA NEGATIVA



El coeficiente de asimetría se puede realizar mediante diversas formas, sin embargo, la más común es mediante el momento estandarizado de la distribución (Upton y Cook, 2014) utilizando la siguiente fórmula:

$$\text{Momento estandarizado} = \frac{\sum_{i=1}^N (x_i - \bar{x})^k / N}{\left(\sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}} \right)^k}$$

Donde:

- ♦ N es el número de observaciones.
- ♦ X_i es cada valor de la muestra.
- ♦ $\bar{x}$ es la media de la muestra.
- ♦ k es el orden del momento (3 para asimetría y 4 para kurtosis).

La interpretación de los valores de asimetría es sencilla:

- ♦ Un valor de asimetría cercano a 0 indica una distribución simétrica.
- ♦ Un valor de asimetría positivo indica asimetría positiva.
- ♦ Un valor de asimetría negativo indica asimetría negativa.

Una distribución difícilmente tendrá un valor exactamente de 0, y por tanto juzgar si un grupo de datos sigue una distribución normal o no utilizando criterio un coeficiente de 0. Debemos de tener flexibilidad. Por ejemplo, los datos de la [figura 13.14](#) tienen un coeficiente de asimetría de -1.10, mientras que los datos de la [figura 13.12](#) tienen un valor de -0.02. Los autores de este libro no hemos identificado un punto de corte bien consensuado sobre cuánto es una “buena” o “mala” asimetría. Además, dependerá mucho del uso que se le quiera dar a los datos. Sin embargo, recomendamos prestar atención cuando hay coeficientes mayores a 1 o menores a -1.

Asimetría en R

Para evaluar asimetría en R utilice la función `skewness()` de la librería `moments`.

- ♦ **Ejemplo 13.3:** genere una serie de datos aleatorios de una distribución normal para evaluar su asimetría. Después evalúe la asimetría de la edad de las mujeres de la base de datos `Pima.tr2`:

```
# Generar los datos
set.seed(123)
datos_normales <- rnorm(1000)
moments::skewness(datos_normales)
[1] 0.06529391
moments::skewness(Pima.tr2$age)
[1] 1.122823
```

Kurtosis

Esta medida se utiliza para identificar la presencia de valores atípicos y la concentración de los datos en el centro de la distribución (Westfall, 2014). Se estima con la siguiente fórmula:

$$Kurtosis = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \sum_{i=1}^N \left(\frac{X_i - \bar{X}}{s} \right)^4 - \frac{3(N-1)^2}{(N-2)(N-3)}$$

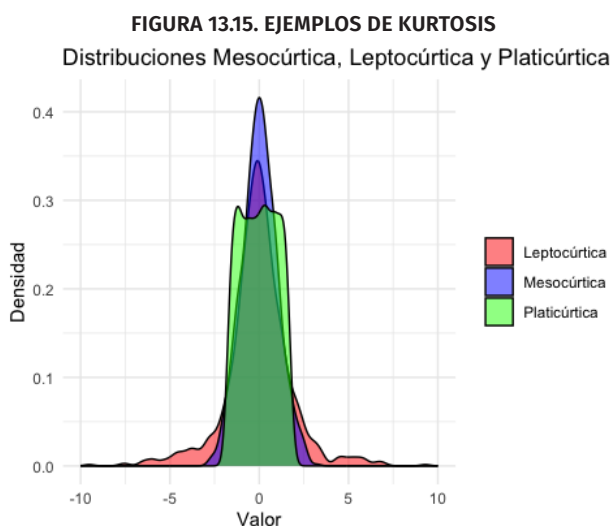
Donde: N es el número de observaciones, X_i es cada valor de la muestra, $\bar{X}$ es la media de la muestra y s es la desviación estándar de la muestra.

La kurtosis permite identificar si los datos una distribución es mesocúrtica, leptocúrtica o platocúrtica.

- ♦ **Distribución mesocúrtica:** es la distribución “ideal”, es decir, indica la forma estándar de la campana de una distribución normal, como la de [figura 13.12](#). Esta distribución tiene un valor de kurtosis cercano a 3.
- ♦ **Distribución leptocúrtica:** una distribución con kurtosis mayor a 3. Las colas son más largas y delgadas, y hay una mayor concentración de valores en el centro. Esto indica una mayor presencia de valores atípicos extremos.
- ♦ **Distribución platocúrtica:** una distribución con kurtosis menor a 3. Las colas son más cortas y gruesas, y los datos están más dispersos alrededor de la media. Esto indica una menor presencia de valores atípicos extremos.

Al igual que la asimetría, es difícil establecer cuando se tiene una buena o mala kurtosis, y de nuevo no se debe ser estricto y utilizar el valor de 3 como único criterio.

La [figura 13.15](#) muestra un ejemplo sobre una distribución mesocúrtica, una distribución leptocúrtica y una distribución platocúrtica.



Kurtosis en R

Para evaluar la kurtosis en R utilice la función `kurtosis()` de la librería `moments`.

- ♦ **Ejemplo 13.4:** genere una serie de datos aleatorios de una distribución normal para evaluar su kurtosis.

```
# Generar los datos
set.seed(123)
datos_normales <- rnorm(1000)
moments::kurtosis(datos_normales)
[1] 2.925747
```

Ejemplo 13.5: ahora evalúe la kurtosis de la edad de las mujeres Pima utilizando la base de datos `Pima.tr2`.

```
moments::kurtosis(Pima.tr$age)
[1] 3.266984
```

Pruebas de hipótesis para evaluar la normalidad

En el [capítulo 14](#) se revisará a detalle qué es una prueba de hipótesis y cómo interpretarlas correctamente. Por ahora recuerde que las pruebas de hipótesis son un método estadístico que permite hacer inferencias sobre una población. Pues bien, existen pruebas de hipótesis que nos permiten concluir si un conjunto de datos se apega a una distribución normal o no. La [tabla 13.1](#) muestra la prueba de hipótesis, una breve descripción y la función que se utiliza en R para evaluarla.

TABLA 13.1. PRINCIPALES PRUEBAS DE HIPÓTESIS PARA EVALUAR LA NORMALIDAD

PRUEBA DE HIPÓTESIS	FUNCIÓN EN R	DESCRIPCIÓN
Shapiro-Wilk	<code>shapiro.test()</code>	Prueba para evaluar la normalidad de los datos, adecuada para pequeñas muestras.

PRUEBA DE HIPÓTESIS	FUNCIÓN EN R	DESCRIPCIÓN
Kolmogorov-Smirnov	<code>nortest::lillie.test()</code>	Prueba que compara la distribución de una muestra con una distribución de referencia, sensible a las diferencias en la cola de la distribución.
Anderson-Darling	<code>nortest::ad.test()</code>	Prueba que pone más énfasis en las colas de la distribución, una mejora del test de Kolmogorov-Smirnov.
D'Agostino-Pearson	<code>moments::agostino.test()</code>	Prueba que evalúa la normalidad combinando las pruebas de asimetría y kurtosis.

Prueba de Shapiro-Wilk

La prueba de Shapiro-Wilk es particularmente adecuada para muestras pequeñas debido a su alta sensibilidad y poder estadístico en detectar desviaciones de la normalidad en tamaños de muestra pequeños. Originalmente, cuando fue propuesta en 1965 fue definida hasta para 50 datos y ha demostrado ser muy eficiente para evaluar normalidad dentro del rango de 3 a 50 muestras (Shapiro et al., 1972). Si bien la prueba se ha extendido a tamaños de muestra más grandes, como hasta 2000 (Shapiro et al., 1972), su diseño original y su enfoque permanecen en tamaños de muestra más pequeños.

Se prefiere la prueba de Shapiro-Wilk para evaluar la normalidad en muestras pequeñas debido a su mayor especificidad en comparación con otras pruebas de normalidad, como las pruebas de Kolmogorov-Smirnov y Anderson-Darling. Las investigaciones han demostrado que la prueba de Shapiro-Wilk da mejores resultados que otras pruebas en general, tanto para distribuciones normales como para distribuciones no normales (Korkmaz y Demir, 2023).

Shapiro-Wilk en R.

Para realizar la prueba de Shapiro-Wilk en R utilice la prueba `shapiro.test()` del paquete `base`.

- ♦ **Ejemplo 13.6:** evalúe la normalidad de la edad de las mujeres de Pima utilizando el *Dataframe* `Pima.tr2`.

```
shapiro.test(Pima.tr2$age)
```

Shapiro-Wilk normality test

data: Pima.tr2\$age

W = 0.87315, p-value = 5.025e-15

Por ahora, solo identifique p -value. Cuando p -value es menor a 0.05 considere que sus datos no se apegan a las características de una distribución normal, lo cual ha sido consistente con las gráficas que hemos realizado con esta variable.

Los autores de este libro creemos firmemente que evaluar una prueba de hipótesis solo con los valores de p es una evaluación incompleta y que no refleja la totalidad del panorama. Pero por ahora, y solo por ahora, utilice el valor de p .

Kolmogorov-Smirnov

Es una prueba estadística propuesta por Kolmogorov y después desarrollada por Smirnov en 1993. Es una prueba estadística no paramétrica que se utiliza para comparar una muestra con una distribución de referencia (prueba de ajuste) o para comparar dos muestras entre sí (prueba de igualdad de distribuciones). En nuestro caso y para fines de este capítulo es de nuestro interés comparar nuestros datos con una distribución normal (distribución de referencia).

Kolmogorov-Smirnov en R

- ♦ **Ejemplo 13.7:** genere una serie de datos aleatorios de una distribución normal y compruebe si siguen una distribución normal y después estime la normalidad de la edad de la mujeres de Pima del *Dataframe* `Pima.tr2`. Utilice la función `ks.test()` del paquete `base`:

```
# Generar una muestra aleatoria
```

```
set.seed(123)
```

```
muestra ← rnorm(50)
```

```
# Realizar la prueba KS para ver si la muestra proviene de una distribución normal
```

```
ks.test(muestra, "pnorm", mean = mean(muestra), sd = sd(muestra))
```

Exact one-sample Kolmogorov-Smirnov test

```
data: muestra
D = 0.083232, p-value = 0.8506
alternative hypothesis: two-sided
```

En el código anterior le indicamos a la función que queremos que compare nuestros datos del objeto `muestra` con una distribución normal “pnorm” con media y desviación estándar obtenida de nuestra muestra.

Con los resultados anteriores no se ha podido demostrar que los datos de muestra sean diferentes de una distribución normal.

Ahora vamos a estimar la normalidad de la edad de las mujeres de Pima:

```
ks.test(Pima.tr2$age, "pnorm", mean = mean(Pima.tr2$age,
                                           sd = sd(Pima.tr2$age)))
```

```
Warning in ks.test.default(Pima.tr2$
ge, "pnorm", mean = mean(Pima.tr2$age, :
ties should not be present for the one-sample Kolmogorov-Smirnov test
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data: Pima.tr2$age
D = 0.56866, p-value < 2.2e-16
alternative hypothesis: two-sided
```

La prueba de Kolmogorov-Smirnov es una prueba estadística ampliamente utilizada para evaluar la bondad del ajuste de una muestra a una distribución de probabilidad de referencia. Sin embargo, existen algunas limitaciones asociadas con su uso. Uno de los principales problemas de la prueba KS es que supone que se conocen los parámetros de la distribución de referencia. En la práctica, estos parámetros a menudo se estiman a partir de los datos, lo que genera posibles imprecisiones en los resultados de las pruebas. Además, la prueba KS es sensible a

las desviaciones en las colas de la distribución, lo que puede llevar a que la prueba sea demasiado conservadora al rechazar la hipótesis nula de ajuste.

Kolmogorov-Smirnov corrección Lilliefors

Para abordar algunas de estas limitaciones, la corrección de Lilliefors se utiliza junto con la prueba KS. La corrección de Lilliefors está diseñada específicamente para evaluar la normalidad de un conjunto de datos y es una adaptación de la prueba KS que tiene en cuenta la estimación de parámetros a partir de los datos. Se recomienda para tamaños de muestra mayores o iguales a 50 al comprobar cuantitativamente la normalidad de un conjunto de datos. La corrección de Lilliefors extiende la prueba KS a situaciones donde se desconocen la media y la varianza de la población, lo que la convierte en una herramienta valiosa en el análisis estadístico.

Kolmogorov-Smirnov Lilliefors en R

Para realizar la prueba de Kolmogorov-Smirnov con corrección de Lilliefors en R utilice la librería `nortest` y la función `lillie.test()`.

- ♦ **Ejemplo 13.8:** utilice los datos del ejemplo [figura 13.7](#) para estimar si el conjunto de datos se ajustan o no a una distribución normal mediante la prueba de Kolmogorov-Smirnov con corrección de Lilliefors, y luego repita esta operación para la edad de las mujeres de Pima:

```
# Si es necesario
install.packages("nortest")
nortest::lillie.test(muestra)

Lilliefors (Kolmogorov-Smirnov) normality test

data:  muestra
D = 0.083232, p-value = 0.5233
nortest::lillie.test(Pima.tr2$age)

Lilliefors (Kolmogorov-Smirnov) normality test
data:  Pima.tr2$age
D = 0.15852, p-value < 2.2e-16
```

Como podrá notar, la decisión comparada con el [ejemplo 13.7](#) es la misma; en ambos casos se demostró que el objeto muestra sigue una distribución normal y que la edad de las mujeres de Pima no se apega a una distribución normal. Sin embargo, el estadístico D que se utiliza para estimar el valor de p , en el caso de la corrección Lilliefors es menor que la prueba sin corrección.

Anderson-Darling

La prueba de Anderson-Darling es otra prueba estadística de hipótesis que se utiliza para evaluar si una muestra determinada de datos proviene de una distribución de probabilidad específica. Esta prueba es particularmente útil para probar la desviación de datos experimentales de distribuciones teóricas (como la distribución normal).

Se considera que la prueba de Anderson-Darling tiene alta sensibilidad, ya que otorga más “peso” a las colas (extremos) a la distribución que estamos evaluando. De hecho, este test es mucho más estricto y sensible en comparación con otras pruebas, como la prueba de Kolmogorov-Smirnov. Si bien tanto Kolmogorov-Smirnov como Anderson-Darling se utilizan comúnmente para evaluaciones de bondad de ajuste (que también los datos se apegan a un modelo o distribución), se prefiere la prueba de Anderson-Darling cuando es necesario garantizar que los datos se ajusten a una distribución específica, lo que la convierte en una opción más estricta en tales casos.

En términos de tamaño de muestra, la prueba de Anderson-Darling ha sido estudiada por su eficiencia con tamaños de muestra limitados, mostrando su aplicabilidad incluso en situaciones donde la disponibilidad de datos es restringida. Por lo que se podría decir que la prueba de Anderson-Darling es especialmente poderosa cuando se prueba la normalidad en escenarios con tamaños de muestra bajos.

Anderson-Darling en R

Utilice la función `ad.test()` de la librería `nortest`.

- ♦ **Ejemplo 13.9:** utilice los datos del ejemplo anterior ([figura 13.8](#)) para estimar si los datos se ajustan a una distribución normal o no mediante la prueba de Anderson-Darling:

```
nortest::ad.test(muestra)

Anderson-Darling normality test

data:  muestra
A = 0.20995, p-value = 0.8529
nortest::ad.test(Pima.tr2$age)

Anderson-Darling normality test

data:  Pima.tr2$age
A = 12.114, p-value < 2.2e-16
```

Si notamos, el resultado sigue siendo el mismo. La muestra tiene una distribución que no es diferente a una normal y `Pima.tr2$age` no es compatible con una distribución normal.

Prueba de D'Agostino-Pearson

La prueba de D'Agostino-Pearson, también conocida como prueba K2, es una prueba muy utilizada y muy sensible y eficaz para evaluar la normalidad en muestras de datos. Se recomienda especialmente cuando se trata de muestras de datos que tienen más de 50 valores. Además, la prueba es especialmente útil cuando el investigador sospecha que la distribución es simétrica con coeficientes de curtosis moderados o altos, especialmente para tamaños de muestra pequeños.

Al comparar la prueba de D'Agostino-Pearson con la prueba de Kolmogorov-Smirnov, la primera se prefiere por su sensibilidad y su uso generalizado en la evaluación de la normalidad. Si bien ambas pruebas son herramientas valiosas, la prueba de D'Agostino-Pearson se recomienda especialmente por su eficacia para detectar desviaciones de la normalidad, cuando los datos muestran una curtosis de moderada a alta y en muestras con más de 50 valores.

Prueba de D'Agostino-Pearson en R

En R utilice la función `agostino.test()` de la librería `moments`.

- ♦ **Ejemplo 13.10:** utilice los datos del [figura 13.8](#) para estimar si los datos se ajustan a una distribución normal o no mediante la rueba de D'Agostino-Pearson:

```

moments::agostino.test(muestra)

D'Agostino skewness test

data:  muestra

skew = 0.16770, z = 0.53726, p-value = 0.5911
alternative hypothesis: data have a skewness
moments::agostino.test(Pima.tr2$age)

D'Agostino skewness test

data:  Pima.tr2$age
skew = 1.1228, z = 6.7067, p-value = 1.991e-11
alternative hypothesis: data have a skewness

```

La conclusión sigue siendo la misma.

Conclusiones

Como hemos notado, todas las pruebas, medidas y gráficas aquí propuestas nos ayudan a entender nuestros datos y a evaluar normalidad de una manera más integral. Volvemos a hacer énfasis en que es altamente recomendable evaluar la normalidad con todas las estrategias aquí revisadas. En cuanto a las pruebas de hipótesis, todos tienen ventajas y desventajas, así como situaciones en las que son especialmente útiles, por lo que utilizar uno u otra dependerá de muchas situaciones. Sin embargo, recomendamos que en el caso de utilizar la prueba de Kolmogorov-Smirnov, siempre utilice una corrección.

Ejercicios

♦ **Ejercicio 13.1:** la base de datos [Base_Normalidad.csv](#) contiene datos de un grupo de pacientes con diabetes mellitus y un grupo controles. Contiene las siguientes variables:

1. ID: Identificador único para cada paciente o registro.
2. PacientesVsCl: clasificación del paciente respecto a si es control o tiene diabetes.
3. ERC (enfermedad renal crónica): indica si el paciente tiene diabetes solo, ERC o ambas.
4. Edad: edad del paciente en años.
5. Leptina: niveles de leptina en la sangre (ng/mL).
6. Adiponectina: niveles de adiponectina en la sangre ($\mu\text{g/mL}$).
7. Marcador1, Marcador2, Marcador3, Marcador4: diferentes biomarcadores clínicos.
8. Proteinuria_g: niveles de proteinuria en gramos.
9. Peso: peso del paciente en kilogramos.
10. Talla: altura del paciente en centímetros.
11. BMI (índice de masa corporal): calculado como peso en kilogramos dividido por el cuadrado de la altura en metros.
12. Porcentaje_Grasa: porcentaje de grasa corporal.
13. Depuracion, eGFR (tasa de filtración glomerular estimada): indicadores de la función renal.
14. Dolor, Dolor6M, Dolor12M: niveles de dolor reportados en diferentes tiempos.
15. Leptin6M, Leptin12M: niveles de leptina en 6 y 12 meses.
16. Adiponectin3M, Adiponectin6M, Adiponectin12M: niveles de adiponectina en 3, 6 y 12 meses.
17. Biomarcador1, Biomarcador2, Biomarcador3: otros biomarcadores específicos.
18. Glucosa_Ayuno: niveles de glucosa en ayuno (mg/dL).
19. HbA1c: hemoglobina glicosilada, indica el control de glucosa en los últimos 2-3 meses (%).
20. Insulina_Ayuno: niveles de insulina en ayuno ($\mu\text{U/mL}$).

- 21. C-Péptido:** nivel de C-Péptido, que ayuda a evaluar la producción propia de insulina (ng/mL).

Utilizando este *Dataframe* realice los siguiente:

1. Filtre la base de datos para seleccionar solo a los pacientes con diabetes.
2. Estime la media, mediana, desviación estándar e IQR para cada una de las variables numéricas.
3. Realice un histograma combinado con un gráfico de densidad y además añada la media y la mediana a este gráfico. Incluya todas las variables numéricas.
4. Realice un boxplot utilizando `ggplot2` para todas las variables numéricas.
5. Realice un Rainclouds para todas las variables numéricas.
6. Realice pruebas de hipótesis para todas las variables numéricas, incluya shapiro wilk, Lilliefors y Anderson-Darling.

Resolución ejercicio 13.1

Descargue y guarde la base en su carpeta de bases. Puede utilizar el menú de Rstudio para importar la abse de datos o utilizar el siguiente código:

```
df <- read.csv("Bases/Base_Normalidad.csv")
```

Filtrar la base de datos:

```
df <- df[df$PacientesVsCl=="DM",]
```

También podemos emplear el siguiente código:

```
df <- subset(df, df$PacientesVsCl=="DM")
```

Se muestra el ejemplo de código para una sola variable.

Para estimar los descriptivos utilice el siguiente código:

```
mean(df$Edad)
median(df$Edad)
```

```
sd(df$Edad)
IQR(df$Edad)
```

Para graficar el histograma:

```
hist(df$Edad, col="#e5f5e0",
      prob=T, # Graficar densidades en lugar de frecuencias
      )
# Agregar una linea de densidad para un conjunto de datos normales
lines(density(df$Edad),
      lwd=2,
      col="#8EC187")
abline(v=mean(df$Edad), col="red", lty = 2, lwd = 4) # Linea vertical
para la media
abline(v=median(df$Edad), col="blue", lty= 6, lwd = 2) # Linea verti-
cal para
mediana
```

Para el boxplot:

```
df|>
  ggplot(mapping = aes(y = Edad))+
  geom_boxplot(fill="#8EC187", alpha=0.8)
```

Para el Rainclouds:

```
library(ggrain) # Si es necesario
df|>
  ggplot(mapping = aes(x = 1, y = Edad))+
  geom_rain(fill = "#19647E", alpha=0.5 )+
  theme_linedraw()
```

Para las pruebas de hipótesis:

```
shapiro.test(df$Edad)
nortest::lillie.test(df$Edad)
nortest::ad.test(df$Edad)
```

Los datos se ajustan a una distribución normal.

La evaluación del resto de las variables se resuelve de la misma forma que el ejemplo anterior.

Referencias

- Ahmad, M. I., Sinclair, C. D. y Spurr, B. D. (1988). Assessment of flood frequency models using empirical distribution function statistics. *Water Resources Research*, 24(8), 1323-1328. <https://doi.org/10.1029/WR024i008p01323>
- Albassam, M., Khan, N. y Aslam, M. (2021). Neutrosophic d'Agostino test of normality: An application to water data. *Journal of Mathematics*, 2021, 1-5. <https://doi.org/10.1155/2021/5582102>
- Allen M, Poggiali D, Whitaker K, Marshall TR, van Berthe, K. A., Abdramane, B. y Reichenbach, S. (2015). Gumbel Weibull distribution function for Sahel precipitation modeling and predicting: Case of Mali. *African Journal of Environmental Science and Technology*, 9(5), 405-412. <https://doi.org/10.5897/ajest2014.1826>
- Bhattacharya, S., Bera, O. P., Saleem, S. M., Hossain, M. M., Varshney, D. S., Kaur, R., Rana, R. K., Tripathi, S., Gokdemir, O., Bacorro, M., Mehta, K. y Singh, A. (2022). Dietary salt consumption pattern as an antecedent risk factor for hypertension: Status, vision, and future recommendations. *Clinical nutrition ESPEN*, 47, 422-430. <https://doi.org/10.1016/j.clnesp.2021.12.006>
- Breton, M. D., DeVore, M. D. y Brown, D. E. (2008). A tool for systematically comparing the power of tests for normality. *Journal of Statistical Computation and Simulation*, 78(7), 623-638. <https://doi.org/10.1080/00949650701377984>
- Breton, M. D., DeVore, M. D. y Brown, D. E. (2008). A tool for systematically comparing the power of tests for normality. *Journal of Statistical Computation and Simulation*, 78(7), 623-638. <https://doi.org/10.1080/00949650701377984>

- Burdenski, T. (2000). Evaluating univariate, bivariate, and multivariate normality using graphical and statistical procedures. *Multiple Linear Regression Viewpoints*, 26(2), 15-28.
- Clauset, A., Shalizi, C. R. y Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4), 661-703. <https://doi.org/10.1137/070710111>
- Contreras--Cristán, A., Gutiérrez--Peña, E. y Walker, S. G. (2017). On the asymptotic power of a goodness-of-fit test based on a cumulative Kullback–Leibler discrepancy. *Statistics & Probability Letters*, 120, 118-125. <https://doi.org/10.1016/j.spl.2016.09.023>
- Coronel-Brizio, H. F., Montoya, A. R. H., Sánchez, H. R. O. y Scalas, E. (2012). *Analysis of short-term price trends in daily stock-market index data*. <https://doi.org/10.48550/arxiv.1211.3060>
- Cumulants, Skewness, Kurtosis and Related Tests. R package version 0.14.1. <https://CRAN.R-project.org/package=moments>
- Dalgaard, P. (2008). Random variables and probability distributions. En *Introductory Statistics with R* (2ª ed., pp. 120-149). Springer.
- Formenti, M., Spadafora, L. y Terraneo, M. (2019). The efficiency of the Anderson–Darling test with a limited sample size: An application to backtesting counterparty credit risk internal models. *Journal of Risk*, 21(6), 69-100. <https://doi.org/10.21314/jor.2019.415>
- Fox, J. y Weisberg, S. (2019). *An R Companion to Applied Regression, Third edition*. Sage. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>
- Gross, J. y Ligges, U. (2015). *Nortest: Tests for Normality*. R package version 1.0-4. <https://CRAN.R-project.org/package=nortest>
- Habibzadeh, F. (2024). Data distribution: normal or abnormal? *Journal of Korean Medical Science*, 39(3). <https://doi.org/10.3346/jkms.2024.39.e35>
- Indrayan, A. y Malhotra, R. K. (2017). Some quantitative aspects of medicine. En *Medical Biostatistics* (4ª ed., pp. 159-199). Chapman & Hall/CRC Biostatistics Series.
- Jäntschi, L. y Bolboacă, S. D. (2018). Computation of probability associated with Anderson–Darling statistic. *Mathematics*, 6(6), 88. <https://doi.org/10.3390/math6060088>
- Jatoi, M. A., Kamboh, S. A., Abro, O. A., Rajput, S. A. y Zardari, L. A. (2024). Investigation the stochastic behaviour of the traffic flow: A case study of a section of a road. *VFAST Transactions on Mathematics*, 12(1), 189-201. <https://doi.org/10.21015/vtm.v12i1.1784>
- Korkmaz, S. y Demir, Y. (2023). Investigation of some univariate normality tests in terms of type-I errors and test power. *Journal of Scientific Reports-A*, (052), 376-395. <https://doi.org/10.59313/jsr-a.1222979>

- Krithikadatta, J. (2014). Normal distribution. *Journal of Conservative Dentistry*, 17(1), 96-97. <https://doi.org/10.4103/0972-0707.124171>. PMID: 24554873; PMCID: PMC3915399
- Kwak, S. G. y Park, S. H. (2019). Normality test in clinical research. *Journal of Rheumatic Diseases*, 26, 5-11. <https://doi.org/10.4078/jrd.2019.26.1.5>
- Le Boedec, K. (2016). Sensitivity and specificity of normality tests and consequences on reference interval accuracy at small sample size: A computer-simulation study. *Veterinary Clinical Pathology*, 45, 648-656. <https://doi.org/10.1111/vcp.12390>
- Limpert, E. y Stahel, W. A. (2011). Problems with using the normal distribution—and ways to improve quality and efficiency of data analysis. *PLoS ONE*, 6(7), e21403. <https://doi.org/10.1371/journal.pone.0021403>
- Maltenfort, M. G. (2015). Understanding a normal distribution of data. *Journal of Spinal Disorders & Techniques*, 28(10), 377-378. <https://doi.org/10.1097/BSD.0000000000000337>. PMID: 26523909
- Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C. y Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of Cardiac Anaesthesia*, 22(1), 67-72. https://doi.org/10.4103/aca.ACA_157_18
- Pearson, K. (1905). Das Fehlergesetz und seine Verallgemeinerungen durch Fechner und Pearson. A rejoinder. *Biometrika*, 4, 169-212.
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.
- Royston, J. P. (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 31(2), 115-124. <https://doi.org/10.2307/2347973>
- Royston, P. (1992). Approximating the Shapiro-Wilk W-test for non-normality. *Statistics and Computing*, 2, 117-119. <https://doi.org/10.1007/BF01891203>
- Ruan, S., Jaggi, C., Xue, J., Fadili, J. M. y Bloyet, D. (2000). Brain tissue classification of magnetic resonance images using partial volume modeling. *IEEE Transactions on Medical Imaging*, 19(12), 1179-1187. <https://doi.org/10.1109/42.897810>
- Shapiro, S. S. y Francia, R. S. (1972). An approximate analysis of variance test for normality. *Journal of the American Statistical Association*, 67(337), 215-216. <https://doi.org/10.1080/01621459.1972.10481232>

- Shin, H., Jung, Y., Jeong, C. y Heo, J. H. (2011). Assessment of modified Anderson–Darling test statistics for the generalized extreme value and generalized logistic distributions. *Stochastic Environmental Research and Risk Assessment*, 26(1), 105-114. <https://doi.org/10.1007/s00477-011-0463-y>
- Takase, H., Sugiura, T., Kimura, G., Ohte, N. y Dohi, Y. (2015). Dietary Sodium Consumption Predicts Future Blood Pressure and Incident Hypertension in the Japanese Normotensive General Population. *Journal of the American Heart Association*, 4(8), e001959. <https://doi.org/10.1161/JAHA.115.001959>
- Uzielli, M. y Phoon, K. (2006). *Some observations on assessment of Gaussianity for correlated profiles*. GeoCongress 2006. [https://doi.org/10.1061/40803\(187\)106](https://doi.org/10.1061/40803(187)106)
- Westfall, P. (2014). Kurtosis as peakedness, 1905-2014. R.I.P. *The American Statistician*, 68(3), 191-195. <https://doi.org/10.1080/00031305.2014.917055>
- Yastrebov, K., Åneman, A., Slama, M., Кохно, Б. Н., Luchansky, V., Orde, S. y Polovnikov, E. (2019). The stop-flow arm equilibrium pressure in preoperative patients: Stressed volume and correlations with echocardiography. *Acta Anaesthesiologica Scandinavica*, 63(5), 594-600. <https://doi.org/10.1111/aas.13318>

Introducción a las pruebas de hipótesis

Edsaúl Emilio Pérez Guerrero

El campo de las ciencias de la salud y en otras disciplinas a menudo requiere que los investigadores hagan inferencias sobre las poblaciones basándose en datos de muestra. Las pruebas de hipótesis son una de las técnicas que permiten trasladar las decisiones tomadas en una muestra hacia la población.

En este capítulo se revisarán los conceptos básicos para la realización de las pruebas de hipótesis. Comenzaremos explicando cómo construir una hipótesis nula y una alterna, posteriormente pasaremos a explicar la relación de la estadística y el azar. Además, explicaremos conceptos, como prueba de hipótesis, poder estadístico, tipos de errores entre otros. Hacia el final del capítulo enlistaremos los pasos que consideramos necesarios para la realización de una prueba de hipótesis y los concluiremos con algunos ejercicios.

Para explicar todos estos conceptos utilizaremos el siguiente ejemplo: imagine que pertenece a un grupo de investigadores cuyo interés es comparar si la eficacia de un nuevo fármaco para tratar el cáncer de pulmón de células no pequeñas es superior a la del tratamiento estándar. Para ello, diseña un ensayo clínico con dos brazos, uno para el tratamiento nuevo y otro para el tratamiento estándar. En este ensayo clínico, la principal variable de desenlace es la tasa de supervivencia a los 5 años. Su objetivo, entonces, es probar que la tasa de supervivencia a los 5 años con el tratamiento nuevo es mejor que la del tratamiento estándar.

Al término de su estudio, usted observa una tasa de supervivencia en el grupo de pacientes con el tratamiento nuevo del 19%, mientras que con el tratamiento estándar la tasa de supervivencia es del 15%. La pregunta aquí es: ¿la

diferencia del 4% entre las tasas de supervivencia es suficiente para concluir estadísticamente que el tratamiento nuevo tiene una mejor eficacia comparado con el tratamiento estándar? Vamos a tratar de responder esta pregunta.

Intervalos de confianza

Antes de comenzar a hablar sobre las pruebas de hipótesis, conviene hacer un paréntesis y hablar de los intervalos de confianza, ya que su concepto se abordará en este capítulo y a lo largo del libro.

Imagine el siguiente escenario: el grupo de investigadores A, después de aplicar una intervención nutricional en un grupo de 80 pacientes, reportó un índice de masa corporal de 25.5 kg/m^2 . Estos investigadores también reportaron que el verdadero valor del índice de masa corporal se encontraba entre 24.0 kg/m^2 y 26.8 kg/m^2 . Por otro lado, el grupo de investigadores B también reportó un índice de masa corporal de 25.5 kg/m^2 , pero concluyó que el verdadero valor del índice de masa corporal de sus 80 pacientes se encontraba entre 23.0 kg/m^2 y 28.8 kg/m^2 . Evidentemente, los investigadores del grupo A reportan resultados con una mayor precisión y, en cierta medida, sus resultados se podrían considerar más confiables. En este sentido, ambos grupos de investigadores reportan sus resultados utilizando **intervalos de confianza**, una herramienta estadística muy importante en ciencias de la salud.

Es común que en Estadística se utilicen estimaciones puntuales basadas en el cálculo de un parámetro poblacional. Como su nombre lo indica, este es un valor único. Por otro lado, los IC son un rango de valores, derivados de los datos muestrales, que se utilizan para estimar el parámetro poblacional. Está asociado a un nivel de confianza, típicamente, y por costumbre, del 95%. En otras palabras, los IC nos ayudan a conocer cuál es el error asociado a una estimación puntual. En lugar de dar una estimación única, un intervalo de confianza da un rango de valores posibles, reflejando la incertidumbre inherente al muestreo. Por lo tanto, son los IC una medida básica en estadística (al menos así lo creemos los autores) porque proporcionan una medida de la precisión y de la variabilidad del estimador puntual. En lugar de dar una estimación única, un intervalo de confianza da un rango de valores posibles, reflejando la

incertidumbre inherente al muestreo. Esperamos que a lo largo de este libro logre comprender la importancia de los IC.

Hay diversas formas de estimar los IC, de hecho, hay un IC para cada parámetro. Por ejemplo, podemos estimar:

- ♦ **Intervalo de confianza para la media**
- ♦ Cuando la desviación estándar poblacional (σ) es conocida.
- ♦ Cuando la desviación estándar poblacional (σ) es desconocida y se utiliza la desviación estándar muestral (s).
- ♦ **Intervalo de confianza para la proporción**
- ♦ **Intervalo de confianza para la diferencia de medias**
- ♦ Muestras independientes.
- ♦ Muestras pareadas.
- ♦ **Intervalos de confianza utilizando la distribución**

A lo largo del libro estimaremos estos y otros IC utilizando R.

Con estos conceptos básicos en mente ya podemos comprender con mayor claridad las pruebas de hipótesis e IC

Hipótesis estadística

En el ensayo del inicio del capítulo, usted y su grupo de investigación partieron de una hipótesis, ya que antes de la realización del estudio se consideró que el nuevo tratamiento debería tener mejor eficacia que el tratamiento estándar. Sin embargo, esto debe ser probado. En Estadística, una hipótesis se define como una suposición que realizamos acerca de una población y que después trasladaremos a nuestra muestra. Las hipótesis estadísticas se deben establecer de tal forma que puedan ser evaluadas por medio de técnicas estadísticas, lo que las hace muy diferentes a una hipótesis de trabajo.

Para la construcción de una hipótesis debemos de establecer dos enunciados, uno dirigido a la igualdad y la nulidad y otro dirigido a buscar diferencias, asociaciones, correlaciones, entre otros. El primero de estos enunciados es la hipótesis nula, mientras que el segundo es la hipótesis alterna. Vamos a explicar más a fondo cada una de ellas.

Hipótesis nula

Es nuestro principal actor. Todo gira en torno a ella. Se trata de una afirmación inicial que deberá ser evaluada mediante una prueba de hipótesis. Generalmente, la hipótesis nula representa la ausencia de un efecto o diferencia, es decir, está dirigida a la nulidad y a la igualdad. Presenta las siguientes características:

- ♦ Se expresa como H_0 .
- ♦ Es la hipótesis que se rechaza o no se rechaza, pero nunca se acepta (ver más adelante).
- ♦ Generalmente representa el *statu quo*, la nulidad o igualdad.

Es importante que tome en cuenta que la H_0 nunca se acepta, solo se rechaza o no se rechaza. No se puede comprobar definitivamente que una hipótesis nula es verdadera. Solo podemos decidir no rechazarla. Las pruebas estadísticas de hipótesis se basan en probabilidad. No hay probabilidades de 0, todo suceso tiene oportunidad de ocurrir, por lo tanto no podemos asegurar que el evento no ocurra.

Para el ejemplo del ensayo clínico del nuevo fármaco para tratar el cáncer de pulmón de células no pequeñas, la H_0 es: “La tasa de supervivencia de los pacientes con el tratamiento nuevo es igual o menor que la del tratamiento estándar.” Al ser la tasa de supervivencia una medida estadística, no es necesario hacer modificaciones a nuestra H_0 . Note cómo la H_0 incluye tanto la igualdad como la posibilidad de que la tasa de supervivencia con el nuevo tratamiento sea menor. Además, recuerde que la siempre es lo contrario a lo que queremos probar.

Hipótesis alterna

La hipótesis alterna es la contraparte de la hipótesis nula. De hecho, es y debe ser el complemento de la hipótesis nula. Presenta las siguientes características:

- ♦ Se expresa como H_A o H_1 .
- ♦ Representa la afirmación que un investigador desea probar.
- ♦ De forma general, incluye diferencias, asociaciones, correlaciones, entre otros.

Para nuestro ejemplo del capítulo, la H_A es: el tratamiento nuevo tiene una mejor tasa de supervivencia que el tratamiento estándar. Note cómo es lo contrario de la H_0 .

Asegúrese que en cuanto formule sus hipótesis se pueden incluir todos los pacientes. Por ejemplo, una hipótesis nula y alternativa incorrectas serían:

- ♦ H_0 : los pacientes con nuevo tratamiento tienen una menor tasa de supervivencia a los 5 años comparado con los pacientes con el tratamiento estándar.
- ♦ H_A : los pacientes con el nuevo tratamiento tienen una tasa de supervivencia mayor que los pacientes tratados con el tratamiento estándar.

Si notamos en el caso anterior, ninguna de las hipótesis incluye a los resultados en los que las tasas son iguales en ambos tratamientos.

Azar y estadística

En nuestro ejemplo del ensayo clínico, se observó que en el tratamiento nuevo los pacientes presentaron una supervivencia del 19% a los 5 años, mientras que en el tratamiento estándar la supervivencia fue del 15%. Clínicamente, esto sin duda es una mejora importante para los pacientes. Sin embargo, de entrada, no se puede asegurar que esta diferencia del 4% nos permita concluir si estadísticamente la eficacia del tratamiento nuevo es mejor comparado con el tratamiento estándar, ya que el azar podría estar interviniendo.

Uno de los principales supuestos de la estadística es que, a menos que se demuestre lo contrario, todas las variaciones observadas en los datos pueden explicarse por el azar. Por ejemplo, el azar explicaría, en primera instancia, las diferencias en el índice de masa corporal entre un grupo de pacientes que fueron intervenidos nutricionalmente y un grupo de pacientes que no recibieron la intervención. Para la estadística, el azar explica las variaciones en las estaturas de un grupo de niños, las diferentes tasas de respuesta entre dos tratamientos para tratar el cáncer, etcétera.

Entonces, ¿el nuevo tratamiento es mejor que el tratamiento estándar? De entrada, el azar me dice que no. El azar podría explicar la diferencia del 4% entre los tratamientos a menos que se pueda demostrar lo contrario. Para poder concluir si el nuevo tratamiento tiene mejor eficacia que el tratamiento estándar, podemos utilizar tres aproximaciones:

- ♦ **Estadística frecuentista**, la cual utiliza pruebas de hipótesis. Este enfoque se basa en la idea de que la probabilidad de un evento se calcula como la proporción de veces que ocurre el evento en un número muy grande de ensayos repetidos. Es decir, tomamos los datos de un gran número de eventos y los comparamos con nuestros datos. La probabilidad está predefinida y no cambia.
- ♦ **Intervalos de confianza (IC)**, que se basan en la estimación del rango que contiene el verdadero valor.
- ♦ **El enfoque bayesiano**, el cual se basa en el teorema de Bayes, que actualiza nuestras creencias sobre la probabilidad de un evento a medida que obtenemos nueva información.

El enfoque bayesiano se escapa de los fines de este libro, por lo que no será revisado aquí. Nos centraremos en los primeros dos enfoques.

Enfoque frecuentista

Para poder demostrar si las tasas de supervivencia a 5 años del nuevo tratamiento son mejores, los investigadores realizan una prueba de hipótesis. En ella plantean una hipótesis nula (hipótesis de no diferencias o de igualdad) en la que establecen que el nuevo tratamiento tiene una tasa de supervivencia menor o igual a la del tratamiento estándar. Después de realizar la prueba de hipótesis, los investigadores obtienen un valor de p de 0.35, por lo que deciden concluir que el nuevo tratamiento no tiene una mejor tasa de supervivencia que el tratamiento estándar y no rechazan la hipótesis nula previamente establecida. En este caso, el azar interviene en gran medida en los resultados.

Tomando en cuenta los principios de la probabilidad, que establecen que la probabilidad solo puede tomar valores entre 0 y 1, podríamos decir entonces con un valor de p de 0.35, que la probabilidad de observar los resultados obtenidos, o más extremos, si la hipótesis nula fuera cierta, es del 35%. Esto significa que el azar explica en gran medida las diferencias en la supervivencia, y no existen argumentos suficientes para afirmar que un factor externo (el nuevo tratamiento) está influyendo en la supervivencia de los pacientes.

Ahora imaginemos otro estudio en el que otros investigadores, después de realizar la prueba de hipótesis, reportaron un valor de p de 0.001. Este valor de p es mucho más pequeño que el reportado por el primer grupo de investigadores, y en este caso, se podría concluir que el nuevo tratamiento tiene una mejor tasa de supervivencia a 5 años comparado con el tratamiento estándar. Pero, ¿por qué en este caso se rechazó la hipótesis nula y se concluyó que el tratamiento nuevo tiene mejor eficacia que el tratamiento estándar? En este nuevo estudio, el valor de p de 0.001 indica que la probabilidad de que los resultados observados, o más extremos, se deban al azar, si la hipótesis nula fuera cierta, es del 0.1%, un porcentaje muy bajo que se considera estadísticamente significativo. En este estudio, sí se podría pensar que existe un factor externo (el nuevo tratamiento) que influye en la supervivencia de los pacientes.

Antes de la realización del estudio, los investigadores deben fijar qué consideran como aceptable y qué como no aceptable (ver más adelante “Error tipo I y tipo II”). Se suele utilizar por costumbre un valor de como el valor limítrofe para considerar un resultado estadísticamente significativo.

Esta explicación de la significancia estadística con los valores de p es muy simplista. Nunca se deben considerar solo los valores de p para tomar una decisión. Además, mejorar la tasa de supervivencia en 4% para los pacientes con cáncer de pulmón de células no pequeñas es un gran logro clínico. A veces, la significancia estadística y clínica son diferentes. Aquí solo nos centramos en la estadística.

Más adelante en el capítulo explicaremos con más detalle los valores de p ; por ahora, solo buscamos ilustrar cómo la probabilidad, la estadística y el azar pueden ayudarnos a hacer inferencias.

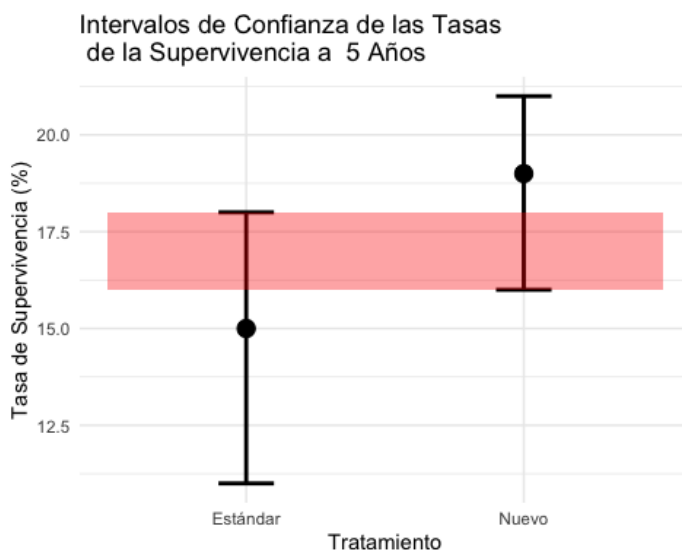
Enfoque utilizando intervalos de confianza

La definición de IC revisada en la sección “Intervalos de confianza” establece que son rangos dentro de los cuales se estima que podría estar el valor del parámetro poblacional con una determinada probabilidad. En este sentido, los IC son muy útiles para la inferencia estadística, además de ser más informativos que un simple valor puntual.

El primer grupo de investigadores reportó que, para el tratamiento estándar, la tasa de supervivencia a 5 años se encontraba con 95% de confianza entre el 11% y el 18%, con una media de 15%. Mientras que para el nuevo tratamiento, la tasa se encontraba entre el 16% y 21%, con una media de 19%. Si prestamos atención a la [figura 14.1](#), podemos notar que existe un solapamiento entre los IC. Esto quiere decir que, en algunas de las tantas veces que se realice la comparación, obtendremos valores iguales (zona roja de la figura).

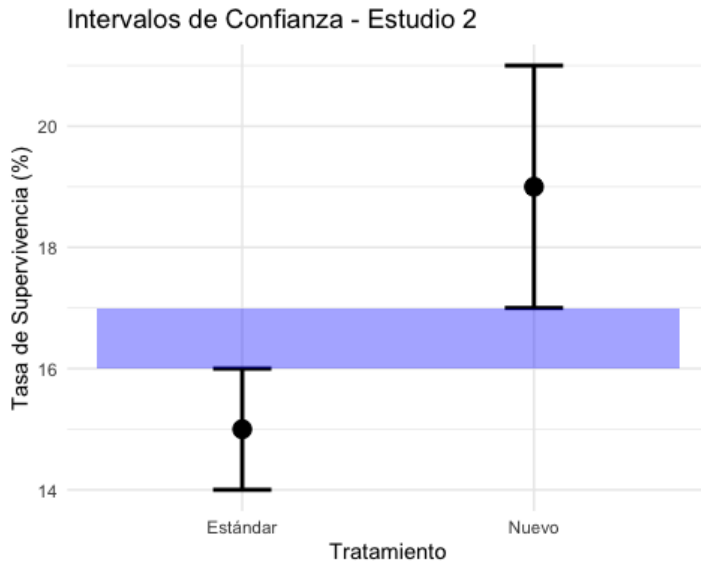
Por lo tanto, con 95% de confianza no podríamos concluir que las tasas de supervivencia sean diferentes entre el tratamiento estándar y el tratamiento nuevo. Este resultado no nos permite trasladar, mediante inferencias, las diferencias en la supervivencia a la población.

FIGURA 14.1. SOLAPAMIENTO DE LOS IC PARA LOS TRATAMIENTOS. ESTUDIO 1



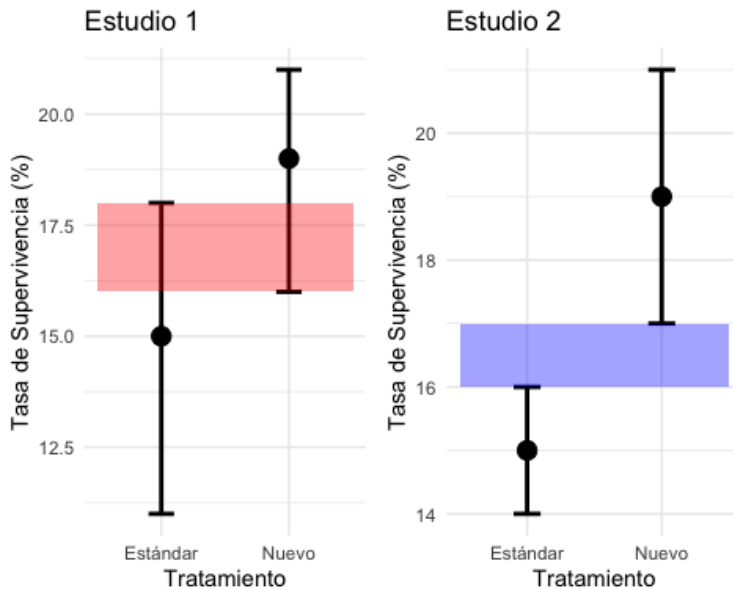
En el caso del segundo estudio, los investigadores reportaron que los intervalos de confianza para el tratamiento nuevo se encontraron entre el 17% y el 21%, mientras que para el tratamiento estándar los intervalos fueron del 14% al 16%. Como podemos observar en la [figura 14.2](#), los intervalos no se solapan (zona azul). Lo que indica que se podría concluir con un nivel de confianza del 95% que el tratamiento nuevo tiene una mejor eficacia que el tratamiento estándar.

FIGURA 14.2. NO SOLAPAMIENTOS DE LOS IC PARA LOS TRATAMIENTOS. ESTUDIO 2



La [figura 14.3](#) muestra la comparación de ambos estudios.

FIGURA 14.3. COMPARACIÓN DE LOS IC AL 95% DE LOS ESTUDIOS



En este libro utilizaremos estos dos enfoques, realizaremos tanto pruebas de hipótesis como la estimación de intervalos de confianza. Aunque los dos enfoques llevan al mismo resultado y hasta cierto punto podrías hablar de un pleonismo estadístico, utilizarlos simultáneamente es mejor.

Conceptos básicos para las pruebas de hipótesis

En la siguiente sección revisaremos algunos de los conceptos básicos relacionados con las pruebas de hipótesis y que nos ayudarán a comprender muchas de las bases de la inferencia estadística.

Nivel de confianza

Cuando estuvimos explicando el ejemplo relacionado con los tratamientos para el cáncer de pulmón de células no pequeñas, indirectamente nos referimos al nivel de confianza. En estadística y en las pruebas de hipótesis, el nivel de confianza representa la probabilidad de que el verdadero parámetro de la población se encuentre dentro de un rango específico, conocido como intervalo de confianza. Los niveles de confianza comúnmente utilizados incluyen 90%, 95% y 99%, sin embargo, cada investigador puede establecerlo según su conveniencia, sin embargo, nunca bajar más allá del 95%.

Como lo habrá notado, el nivel de confianza es el complemento del valor de que se selecciona como umbral. Para un nivel de confianza del 95%, el valor de p es 0.05, y esto tiene una relación directa con la probabilidad, que solo puede tomar valores entre 0 y 1.

Valores de p

El valor de p no es otra cosa más que un valor de probabilidad que en estadística es utilizado para determinar la significancia de una prueba de hipótesis. Específicamente, “el valor p es la probabilidad condicional de que un estadístico de prueba sea igual o más extremo al observado, dado que el conjunto de supuestos es verdadero” (Fraser, 2023).

El valor de p presenta las siguientes características:

- ♦ Puede tomar valores de 0 a 1.
- ♦ Un valor pequeño, generalmente $p < 0.50$ sugiere que es poco probable que los resultados observados sean debidos al azar, lo que lleva a rechazar la hipótesis nula. Tome este enunciado con mucha precaución, nunca utilice solo el valor p para tomar decisiones.
- ♦ Un valor p grande indica que los resultados pueden ser consistentes con la hipótesis nula.

Por costumbre, se ha utilizado un valor de $p < 0.50$ como el punto de corte para decidir si se rechaza o no la H_0 ; sin embargo, esto no es una regla estricta y cada vez hay más detractores de ello que consideran que el punto de corte debería de reimplantarse y considerar un valor más pequeño (Chen et al., 2023; Benjamin et al., 2018; Ioannidis, 2018)).

Algunas de las ventajas de utilizar el valor de p son:

- ♦ Es ampliamente utilizado en la literatura científica.
- ♦ Proporciona una medida cuantitativa de la evidencia contra H_0 .
- ♦ Hasta cierto punto, permite comparar entre diferentes resultados.

Sin embargo, también presenta algunas desventajas que describimos a continuación:

- ♦ Es dependiente del tamaño de la muestra y de la variación de los datos.
- ♦ No proporciona información sobre la magnitud del efecto.
- ♦ Puede llevar a decisiones binarias (significativo/no significativo) sin considerar el contexto.
- ♦ Frecuentemente es malinterpretado como la probabilidad de que la hipótesis nula sea verdadera o como una medida de la importancia del efecto.

Aunque existen cada vez más cuestionamientos sobre el uso de los valores p esta discusión se escapa de los fines de este libro. Al considerarse como un estándar, este libro utilizará los valores p y las pruebas de hipótesis.

Consejos para la interpretación de los valores de

Uno de los principales problemas con utilizar valores de p es su incorrecta interpretación. A continuación, se dan algunos consejos que puede utilizar para interpretar correctamente un valor de :

- ♦ La observación de un valor p menor que un punto de corte (por ejemplo, 0.05) por sí sola no respalda, y no debería respaldar, una conclusión científica binaria. Esto es más evidente y mucho más importante de seguir si el valor p está cerca del umbral, por ejemplo, 0.045 o 0.055.
- ♦ Existen otros puntos más importantes antes de “modificar” el valor p . Por ejemplo, el investigador debería replantearse si el diseño del estudio es adecuado, si la recolección de datos es apropiada, si el procesamiento de los datos es riguroso, si el modelo es adecuado y si se cumplen todos los supuestos, entre otros.
- ♦ Un valor de 0.05 no es el estándar de oro. Este valor fue acuñado por Fisher por conveniencia y no por otra razón.
- ♦ No se debe interpretar como la probabilidad de que la H_0 sea verdadera o como una medida de la importancia del efecto.
- ♦ Nunca se debe reportar un valor de p como desigualdad (por ejemplo, $p < 0.02$, $p > 0.05$), salvo cuando sean valores de .
- ♦ El valor de p no es la evidencia en favor de H_A .
- ♦ Un valor p pequeño simplemente indica que los datos son inusuales, si todos los supuestos utilizados para calcularlos (incluida la hipótesis de prueba) fueron correctos.

Error sistemático y error aleatorio

Toda medición es susceptible a errores. En el contexto clínico y de investigación, es común que las mediciones se realicen de forma incorrecta. Algunas veces estos errores se introducen de forma aleatoria y otras de forma sistemática.

Antes de definir más detalladamente un error sistemático y un error aleatorio, debemos definir otros conceptos relacionados con las mediciones: exactitud y precisión. El primero se refiere a qué tanto una medición se acerca al valor real, y el segundo hace referencia a qué tanto se asemejan varias mediciones entre sí. Lo ideal sería que todas las mediciones que realizamos sean precisas y exactas.

Vamos a retomar el ejemplo del ensayo clínico de los pacientes con cáncer de pulmón de células no pequeñas. Una de las evaluaciones clínicas que se realizan de forma regular en estos pacientes es la medición de la Lactato Deshidrogenasa (LDH), la cual, después de una reacción enzimática, se cuantifica de forma indirecta a 340 nm en un espectrofotómetro. Si la absorbancia que arroja el espectrofotómetro permite cuantificar la LDH y este valor se asemeja a la realidad, podemos decir entonces que nuestra medición es exacta. Además, si después de repetir varias veces esta medición obtenemos resultados similares o idénticos, entonces estamos realizando una medición precisa.

Relacionado con la exactitud y la precisión, hay dos tipos de errores que describiremos a continuación.

Error aleatorio

El error aleatorio se refiere a las variaciones que ocurren debido al azar en el proceso de muestreo y medición. Son errores que no se pueden controlar. En el ejemplo del ensayo clínico, un error aleatorio podría surgir de diferencias individuales entre los pacientes, variaciones en la implementación del tratamiento o cualquier otro factor no controlado que pueda afectar las tasas de supervivencia.

El error aleatorio puede reducirse aumentando el tamaño de la muestra y asegurando que las condiciones del ensayo sean lo más homogéneas posible. Pero no puede reducirse cambiando un instrumento de medición. El aumento del tamaño de la muestra permitiría que los errores en las mediciones se distribuyan aleatoriamente en ambos brazos del ensayo clínico, pero cambiar el espectrofotómetro no influiría en la reducción de este error, ya que un error aleatorio depende de varios factores y no solo de un instrumento de medición.

Error sistemático

El error sistemático, por otro lado, se refiere a desviaciones que ocurren de manera consistente y predecible debido a algún sesgo en el diseño del estudio, la recolección de datos o el análisis. Estos errores no se reducen aumentando el tamaño de la muestra y pueden llevar a conclusiones incorrectas, si no se identifican y corrigen. Imagine que el espectrofotómetro que se utilizó para evaluar la LDH presenta un problema y no puede leer correctamente a 340 ; en este instrumento, siempre que se realiza una lectura a esta longitud de onda, nos dará un resultado de 0.3 unidades

de absorbancia por encima del valor real, de tal manera que tendremos valores más altos al valor real de LDH en todos los pacientes. No podremos percatarnos del error a menos que cambiemos de instrumento de medición.

Los errores sistemáticos no necesariamente tienen que ver con instrumentos de medición. Por ejemplo, otro error sistemático en el ensayo clínico sería si el nuevo tratamiento se administra en un hospital con mejores recursos y atención más intensiva que el hospital donde se administra el tratamiento estándar. Las diferencias en las tasas de supervivencia podrían deberse a estas diferencias en lugar de a la eficacia del fármaco.

Un error sistemático es un sesgo y se debe controlar.

Error tipo I y tipo II

Otra problemática a la que nos enfrentamos a la hora de realizar una prueba de hipótesis es de rechazar la H_0 cuando no se debió rechazar o por el contrario no rechazar H_0 cuando esta debió ser rechazada. La [tabla 14.1](#) muestra las posibles combinaciones de errores que se pueden cometer cuando se realiza una prueba de hipótesis.

TABLA 14.1. ERRORES TIPO I Y II

HIPÓTESIS NULA			
		Verdadera	Falsa
Decisión	No rechazar	Acierto. 1-. Nivel de confianza	Error tipo 2.
	Rechazar	Error tipo 1.	Acierto. 1-. Potencia estadística

El error tipo I es la probabilidad de rechazar la hipótesis nula cuando esta no debería ser rechazada. En el contexto del ensayo clínico, un error tipo I sería que los investigadores concluyeran que la tasa de supervivencia del nuevo tratamiento es superior a la del antiguo tratamiento, pero en realidad estas no son diferentes. Es decir, se rechazó la H_0 cuando esta no debió haber sido rechazada. El riesgo de cometer el error tipo I se conoce como α y, por costumbre, se ha fijado en 0.05. Tome en cuenta que un α de 0.05 no es una ley y que este valor viene acuñado de trabajos del siglo pasado. Puede ser más estricto con α , pero no se recomienda ser más laxo. Además, tome en cuenta que α siempre se fija antes del estudio.

Al establecer α en 0.05, se acepta 5% de probabilidad de rechazar incorrectamente la hipótesis nula cuando en realidad no debe ser rechazada. En contextos donde se realizan múltiples comparaciones, este riesgo se incrementa, lo que puede llevar a un número considerable de falsos positivos. Cuantas más pruebas de hipótesis haga con sus datos, más posibilidades tiene de cometer un error tipo I.

Ahora suponga que sucede lo contrario. Al término del estudio se concluye que las tasas de supervivencia en los dos tratamientos son iguales; sin embargo, en realidad, la tasa de supervivencia de los pacientes con el tratamiento nuevo es superior a la del tratamiento estándar. Es decir, no se rechazó la H_0 cuando esta se debió rechazar (error tipo II). En otras palabras, el estudio está fallando en encontrar diferencias donde sí las hay. La probabilidad de cometer un error tipo II se conoce como β y se suele fijar en 0.20.

En la mayoría de los escenarios es más grave cometer un error tipo I que un error tipo II. En el ejemplo del ensayo clínico, si los investigadores cometieran un error tipo II, el tratamiento estándar seguiría en el mercado. Sin embargo, un error tipo I podría ocasionar que un fármaco no eficaz sustituya al tratamiento estándar.

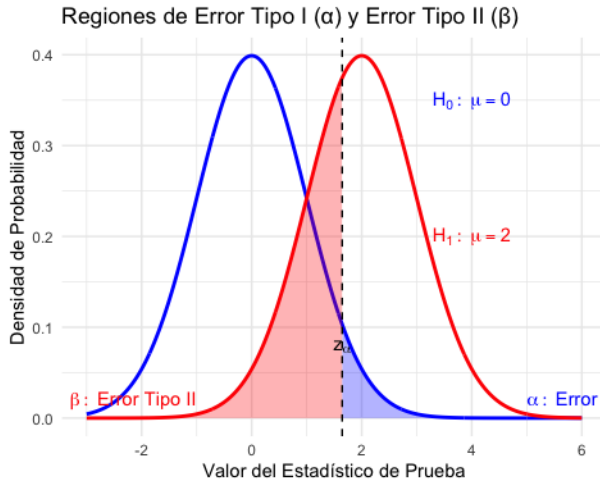
Poder estadístico

α y β tienen mucha relación con el poder estadístico. Si usted utiliza anteojos le será muy fácil comprender qué es el poder estadístico. Cuando sale de su casa con los anteojos puestos, usted tiene el poder de ver correctamente cada detalle de su exterior; ahora, si usted sale de su casa sin ellos, pierde ese poder. El poder estadístico es la capacidad que tienen las pruebas estadísticas de detectar diferencias y se estima como $1-\beta$. Cuanto más estricto sea con el valor de α y mayor sea el valor de β , menos poder tendrá una prueba.

En la [figura 14.4](#) se ilustran gráficamente α y β . En esta gráfica se representa la H_0 con la curva en color azul. La zona sombreada también en azul indica el error tipo I (α). La línea discontinua negra marca el umbral de significancia (z_α). Por otro lado, la curva en color rojo representa la distribución de la prueba bajo H_A . Mientras que la región sombreada roja indica el error tipo II (β).

En la [figura 14.4](#), el poder estadístico (ver más adelante) no está marcado explícitamente en la gráfica, pero es la región bajo la curva roja a la derecha del umbral de significancia. Representa la capacidad de la prueba para detectar una diferencia cuando realmente existe.

FIGURA 14.4. ERROR TIPO I Y II



Puntos clave sobre error tipo I y II

1. Error tipo I (α): es la probabilidad de rechazar la hipótesis nula cuando esta no se debe rechazar. Por conveniencia se acepta 5% de probabilidad de cometer este error con $\alpha = 0.05$.
2. Error tipo II (β): es la probabilidad de no rechazar la hipótesis nula cuando esta es falsa. Un valor común para β es 0.20, lo que implica un poder estadístico del 80%.
3. Poder estadístico ($1 - \beta$): es la capacidad de una prueba estadística para detectar una diferencia cuando realmente existe. Un poder del 80% es comúnmente aceptado.
4. Importancia relativa de los errores: en muchos casos, es más grave cometer un error tipo I que un error tipo II, especialmente en contextos como el de ensayos clínicos.

Pruebas de hipótesis

Hasta ahora, hemos hablado de una hipótesis nula, una hipótesis alternativa, de nivel de confianza y valores de p . Todos estos conceptos se integran dentro de la prueba de hipótesis. Estas se utilizan para tomar decisiones en una muestra

y trasladarlas a la población. Se considera un procedimiento formal que se utiliza para rechazar o no rechazar una H_0 , es decir, nos ayuda a encontrar los argumentos necesarios para poder rechazar o no rechazar la H_0 .

Para entender un poco más las pruebas de hipótesis y evitar su uso incorrecto, vamos a utilizar un ejemplo clásico fuera de las ciencias de la salud y de la estadística, pero que puede ayudar a nuestro fin.

En algunos sistemas de justicia, cuando se presume que alguien comete un delito, debe considerarse como inocente hasta que se demuestre lo contrario. Posteriormente, un juez junto con un jurado decide declarar, con base en la evidencia reunida, “culpable” o “no culpable” a la persona juzgada. Note cómo la persona no es declarada como “inocente”. Lo mismo ocurre con las pruebas de hipótesis. Se inician asumiendo que la H_0 es verdadera, después hacemos un procedimiento para reunir evidencia, y con esa evidencia rechazamos o no rechazamos la H_0 . Nunca aceptamos la H_0 como verdadera, así como nunca se declara a la persona juzgada como inocente. En las pruebas de hipótesis no se determina la probabilidad de que una H_0 sea cierta, dado que están diseñadas para evaluar si los datos observados o los más extremos son consistentes con la H_0 .

No lo olvide: “no rechazar la hipótesis nula implica que no hay evidencia suficiente para rechazar la hipótesis nula, pero no necesariamente significa que la hipótesis nula sea verdadera” (Fraser, 2023).

Para el ejemplo del ensayo clínico, la realización de la prueba de hipótesis permitiría obtener la evidencia para probar si la tasa de supervivencia de los pacientes con el nuevo tratamiento es superior a la del tratamiento estándar. Es decir, nos permitiría reunir evidencia para rechazar o no la H_0 de que la tasa de supervivencia en los pacientes con el nuevo tratamiento es menor o igual que la del tratamiento estándar.

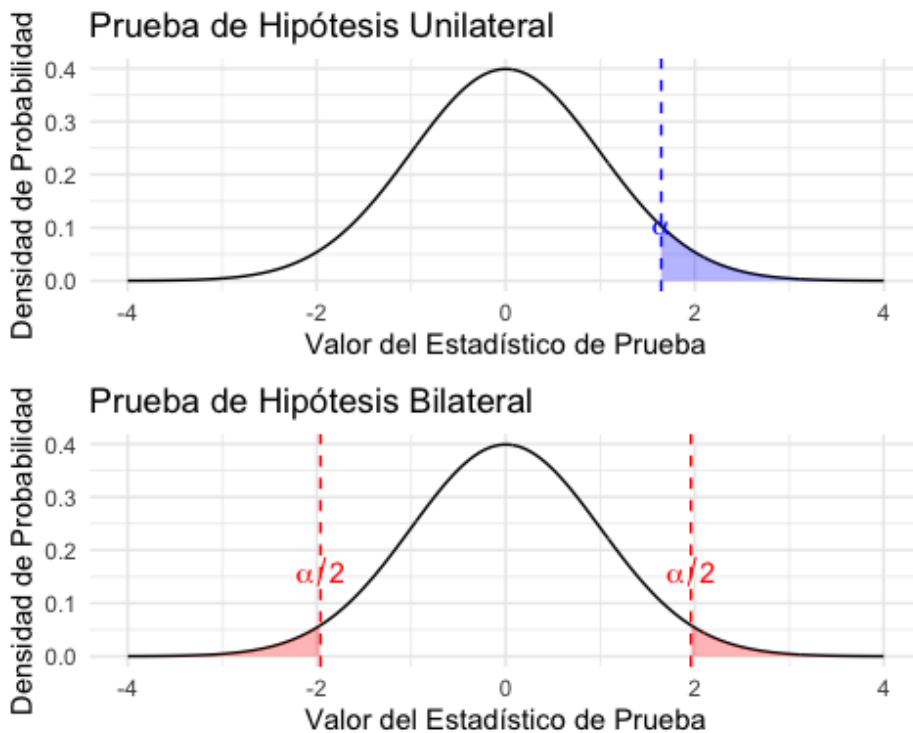
Pruebas unilaterales y bilaterales

Cuando establecimos la H_A para el ensayo clínico, se planteó con el objetivo de evaluar si el tratamiento nuevo presentaba una mejor tasa de supervivencia que el tratamiento estándar. Esta H_A se planteó de forma unilateral (a una cola), es decir, para buscar una diferencia en un solo sentido, positiva en este caso. Cuando las hipótesis se plantean para buscar una diferencia, sin importar el sentido, ya sea positiva o negativa, hablamos de una prueba de hipótesis

bilateral. En este caso, el valor α se reparte en cada uno de los extremos de la distribución. Si α es de 0.05, repartiríamos 0.025 para la diferencia positiva y 0.025 para la diferencia negativa. Por ejemplo, una H_A para el ensayo clínico sería: las tasas de sobrevida a los 5 años del tratamiento nuevo es diferente al tratamiento estándar.

Se suelen utilizar las hipótesis bilaterales en la mayoría de los casos. La [figura 14.5](#) ejemplifica una prueba de hipótesis unilateral y bilateral. En la imagen en color azul se muestra el valor de α para una hipótesis unilateral y en rojo el valor α para una hipótesis bilateral. Note cómo el valor de la primera distribución se repartió en ambos extremos de la segunda distribución.

FIGURA 14.5. HIPÓTESIS UNILATERAL Y BILATERAL



Pruebas de hipótesis paramétricas y no paramétricas

El último de los conceptos que necesitas conocer para poder emplear una prueba de hipótesis es el relacionado con las pruebas estadísticas paramétricas y no paramétricas. Las pruebas paramétricas exigen, para su correcta aplicación, el cumplimiento de ciertos parámetros y supuestos, ya que se basan en distribuciones de probabilidad conocidas que necesitan a su vez del conocimiento de parámetros, como la media, la desviación estándar, entre otros. El más importante de estos es el cumplimiento de la normalidad.

Por otro lado, las pruebas no paramétricas, también conocidas como pruebas libres de parámetros, no necesitan del cumplimiento de parámetros específicos para su aplicación. Este concepto de pruebas paramétricas y no paramétricas solo aplica para las variables numéricas. Cuando las muestras son pequeñas o cuando no se cumplen los supuestos de normalidad, deberían emplearse pruebas no paramétricas.

Pasos para la resolución de pruebas de hipótesis

Realizar pruebas de hipótesis solo por realizarlas es uno de los principales errores en estadística. Para tratar de minimizar esta situación, hemos establecido una serie de pasos que le recomendamos seguir en la medida de lo posible:

1. **Formulación de preguntas:** idealmente, antes de la realización de una prueba de hipótesis, se deberían formular y responder algunas preguntas, por ejemplo: ¿qué quiero probar? ¿Cuál es el objetivo de mi prueba? ¿Con qué datos cuento? ¿Quiero comparar medias? ¿Quiero comparar variables cualitativas? Entre otras.
2. **Formulación de hipótesis:** el siguiente paso es la formulación de hipótesis estadísticas.
3. **Estadística descriptiva y entendimiento de los datos:** después de formular sus hipótesis, el siguiente paso debe ser la realización de estadística descriptiva (incluidas gráficas). Debemos entender qué nos dicen nuestros datos. Tenga en cuenta los datos con los que se cuenta (que se incluyen en el problema), por ejemplo:

- » Medias
 - » Variación
 - » Distribución
 - » Otras medidas descriptivas En este punto, también en ocasiones se deberá probar si los datos siguen o no una distribución normal. Obtenga de sus datos toda la información posible y realice gráficas (son gratis).
4. **Selección de la prueba estadística:** tomando en cuenta mi hipótesis, la distribución de mis datos y los datos obtenidos del problema, debemos seleccionar la prueba estadística a utilizar. Puede apoyarse del [siguiente diagrama](#). Este diagrama está diseñado para iniciarse desde el centro e ir avanzando a través de preguntas con respuestas dicotómicas. Si se identifican correctamente los tipos de variables, la naturaleza de los grupos y los objetivos de la investigación, se puede seleccionar con facilidad alguna de las pruebas estadísticas clásicas descritas en el diagrama. Para utilizar correctamente el diagrama debe antes:
- » Identificar el tipo de variables a comparar o asociar.
 - » Identificar si la comparación se realizará entre grupos o si se busca estimar la relación con otra variable.
 - » Identificar si se trata de mediciones en grupos independientes o de mediciones realizadas en el mismo grupo a lo largo del tiempo.
 - » Si es el caso, identificar la cantidad de grupos que se comparan.
 - » Identificar la distribución de los datos.
5. **Evaluación de los supuestos de la prueba estadística:** cada prueba estadística requiere cumplir algunos supuestos, por lo tanto, es necesario evaluar si se cumplen estos supuestos. Puede apoyarse de las siguientes preguntas:
- » ¿Qué necesito cumplir para poder utilizar la prueba?
 - » ¿Mis datos cumplen con los supuestos necesarios para emplear mi prueba? La intención de este punto es evaluar si cumplimos con los requisitos para utilizar la prueba estadística seleccionada. Si no se cumplen, debo seleccionar otra prueba.
6. **Valores críticos y regla de decisión:** las pruebas de hipótesis requieren de una “moneda de cambio”, es decir, de un valor de una distribución conocida, por ejemplo, valores de t de la distribución t -Student, valores de F de la distribución Fisher-Snedecor. Lo primero, entonces, es determinar

a partir de qué valores de “mi moneda de cambio” voy a considerar que rechazo o no rechazo mi H_0 . Este valor está directamente relacionado con el valor de α . Por ejemplo, el valor crítico inferior para una distribución t con 50 datos es de -2.01. En este punto, se debe establecer una zona de rechazo y una zona de aceptación. En este libro, esta zona se representará de forma gráfica.

7. **Estimación del estadístico de prueba:** el siguiente paso es estimar mi moneda de cambio a partir de mis datos. Este valor posteriormente me servirá para compararlo con el de la distribución conocida (paso anterior).
8. **Decisión:** al comparar la moneda de cambio de la distribución conocida con la moneda de cambio de mis datos, ¿rechazo o no rechazo mi hipótesis nula?
9. **Intervalos de confianza:** estimar IC ayudará mucho a reforzar la decisión tomada en el paso anterior. En este libro, la mayoría de las pruebas de hipótesis irán acompañadas de la estimación de IC ya que estos pueden denotar una mayor o menor precisión de los resultados. Los intervalos amplios denotan poca potencia y poca precisión.
10. **Valor de p :** utilizando nuestra moneda de cambio vamos a estimar la probabilidad de observar los valores de mis datos o más extremos cuando la H_0 es verdadera. Para ello, estimamos la probabilidad de que mi moneda de cambio se encuentre en una determinada distribución.
11. **Conclusión:** siempre tome en cuenta todos los puntos anteriores y su nivel de confianza para concluir.

Una última recomendación para la realización de pruebas de hipótesis

Realizar pruebas de hipótesis sin un sustento adecuado y realizar estas pruebas por el solo hecho de realizarlas, es un error. En una prueba de hipótesis siempre se debe de considerar el contexto del estudio, los datos y sus características, el tamaño de la muestra, entre otros. Recuerde que la prueba de hipótesis ofrecerá un resultado binario (rechazo o no rechazo), por lo que siempre considere otras medidas como los IC o el tamaño del efecto (ver “[Tamaño del efecto](#)”).

Referencias

- Armstrong, R. A. (2014). When to use the Bonferroni correction. *Ophthalmic & physiological optics: the journal of the British College of Ophthalmic Opticians (Optometrists)*, 34(5), 502–508. <https://doi.org/10.1111/opo.12131>
- Bhattacharya, S., Bera, O. P., Saleem, S. M., Hossain, M. M., Varshney, D. S., Kaur, R., Rana, R. K., Tripathi, S., Gokdemir, O., Bacorro, M., Mehta, K. y Singh, A. (2022). Dietary salt consumption pattern as an antecedent risk factor for hypertension: Status, vision, and future recommendations. *Clinical nutrition ESPEN*, 47, 422–430. <https://doi.org/10.1016/j.clnesp.2021.12.006>
- Benjamin, D. J., Berger, J. O., Johannesson, M., Nosek, B. A., Wagenmakers, E. J., Berk, R., Bollen, K. A., Brembs, B., Brown, L., Camerer, C. et al. (2018). Redefine statistical significance. *Nature Human Behaviour*, 2, 6–10.
- Chén, O. Y., Bodelet, J. S., Saraiva, R. G., Phan, H., Di, J., Nagels, G., Schwantje, T., Cao, H., Gou, J., Reinen, J. M., Xiong, B., Zhi, B., Wang, X. y De Vos, M. (2023). The roles, challenges, and merits of the p value. *Patterns*, 4(12), 100878. <https://doi.org/10.1016/j.patter.2023.100878>
- Fraser, R. A. (2023). Inappropriate use of statistical power. *Bone Marrow Transplantation*, 58(5), 474–477. <https://doi.org/10.1038/s41409-023-01935-3>
- Goel, H., Raheja, D. y Nadar, S. K. (2024). Evidence-based medicine or statistically manipulated medicine? Are we slaves to the P-value? *Postgraduate medical journal*, 100(1185), 451–460. <https://doi.org/10.1093/postmj/qgae012>
- Grech, V. y Eldawlatly, A. A. (2023). The P value - and its historical underpinnings - pro and con. *Saudi journal of anaesthesia*, 17(3), 391–393. https://doi.org/10.4103/sja.sja_223_23
- Greenland, S., Senn, S. J., Rothman, K. J., Carlin, J. B., Poole, C., Goodman, S. N. y Altman, D. G. (2016). Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations. *European journal of epidemiology*, 31(4), 337–350. <https://doi.org/10.1007/s10654-016-0149-3>
- Indrayan, A. y Malhotra, R. K. (2017). Confidence Intervals, Principles of Tests of Significance, and Sample Size. En *Medical Biostatistics* (4ª ed., pp. 255–304). Chapman & Hall/CRC Biostatistics Series.
- Ioannidis, J. P. A. (2018). The proposal to lower p value thresholds to .005. *JAMA*, 319, 1429–1430.

- Kühberger, A., Fritz, A., Lerner, E. y Scherndl, T. (2015). The significance fallacy in inferential statistics. *BMC research notes*, 8, 84. <https://doi.org/10.1186/s13104-015-1020-4>
- Kumbhare, D., Alavinia, M. y Furlan, J. (2019). Hypothesis Testing in Superiority, Noninferiority, and Equivalence Clinical Trials: Implications in Physical Medicine and Rehabilitation. *American journal of physical medicine & rehabilitation*, 98(3), 226–230. <https://doi.org/10.1097/PHM.0000000000001023>
- Lee, D. K. (2016). Alternatives to P value: confidence interval and effect size. *Korean journal of anesthesiology*, 69(6), 555-562. <https://doi.org/10.4097/kjae.2016.69.6.555>
- Lehmann, E. L. (1993). The Fisher, Neyman-Pearson Theories of Testing Hypotheses: One Theory or Two? *Journal of the American Statistical Association*, 88(424), 1242-1249. <https://doi.org/10.1080/01621459.1993.10476404>
- Mansournia, M. A., Nazemipour, M. y Etminan, M. (2022). P-value, compatibility, and S-value. *Global Epidemiology*, 4, 100085. <https://doi.org/10.1016/j.gloepi.2022.100085>
- Martínez González, M. Á., Toledo, E., Donat-Vargas, C. y Lopez-Fidalgo, J. (2020). Intervalos de confianza y contraste de hipótesis. En M. Á. Martínez González, A. Sánchez Villegas, E. Toledo Atucha y J. Faulin Fajardo (Eds.), *Bioestadística amigable* (4ª ed., pp. 101-146).
- McCloskey, A. y Michailat, P. (2020). Critical values robust to p-hacking. *arXiv*. <https://doi.org/10.48550/arXiv.2005.04141>
- Royston, J. P. (1982). An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31(2), 115-124. <https://doi.org/10.2307/2347973>
- Royston, P. (1992). Approximating the Shapiro-Wilk W-test for non-normality. *Stat Comput* 2, 117-119. <https://doi.org/10.1007/BF01891203>
- Shrout, P. E. (1997). Should significance tests be banned? Introduction to a special section exploring the pros and cons. *Psychological Science*, 8, 1-2.
- Sidebotham, D., Dominick, F., Deng, C., Barlow, J. y Jones, P. M. (2024). Statistically significant differences versus convincing evidence of real treatment effects: an analysis of the false positive risk for single-centre trials in anaesthesia. *British journal of anaesthesia*, 132(1), 116-123. <https://doi.org/10.1016/j.bja.2023.10.036>
- Spanos, A. (2010). Is frequentist testing vulnerable to the base-rate fallacy. *Philosophy of Science*, 77, 565-583.

- Szucs, D. y Ioannidis, J. P. A. (2017). When Null Hypothesis Significance Testing Is Unsuitable for Research: A Reassessment. *Frontiers in human neuroscience*, 11, 390. <https://doi.org/10.3389/fnhum.2017.00390>
- Takase, H., Sugiura, T., Kimura, G., Ohte, N. y Dohi, Y. (2015). Dietary Sodium Consumption Predicts Future Blood Pressure and Incident Hypertension in the Japanese Normotensive General Population. *Journal of the American Heart Association*, 4(8), e001959. <https://doi.org/10.1161/JAHA.115.001959>
- Tsushima, E. (2022). Interpreting Results from Statistical Hypothesis Testing: Understanding the Appropriate P-value. *Physical therapy research*, 25(2), 49-55. <https://doi.org/10.1298/ptr.R0019>
- Wei, Z., Yang, A., Rocha, L., Miranda, M. F. y Nathoo, F. S. (2022). A Review of Bayesian Hypothesis Testing and Its Practical Implementations. *Entropy (Basel, Switzerland)*, 24(2), 161. <https://doi.org/10.3390/e24020161>
- Whitley, E. y Ball, J. (2002). Statistics review 3: hypothesis testing and P values. *Critical care*, 6(3), 222-225. <https://doi.org/10.1186/cc1493>

Pruebas de hipótesis para la comparación de una media

Miryam Rosario Guillén Medina

Ana Gabriela Flores Vargas

Imagine el siguiente escenario. Usted es un médico adscrito al área de ginecología de un hospital. Actualmente en el hospital se están realizando una serie de estudios cuyo objetivo es conocer los factores asociados a las complicaciones en los recién nacidos. Su tarea en específico es evaluar si la media de los pesos de los recién nacidos no supera los 3 000 gramos. Para esto, el encargado del hospital le entrega a usted datos de una muestra representativa de los pesos de los recién nacidos en su hospital durante el último año.

Para cumplir con su objetivo usted tiene dos opciones. Una de ellas es la realización de una prueba t de Student o una prueba z . Este capítulo se centra en la evaluación de estas dos pruebas, primero comenzaremos con la explicación de la prueba t de Student y en la segunda parte del capítulo nos enfocaremos en la prueba z . En un inicio, explicaremos las pruebas de hipótesis conforme a los pasos descritos en “[Pasos para la resolución de pruebas de hipótesis](#)”, sin embargo, conforme avancemos en el desarrollo de este capítulo iremos dejando de lado algunos de estos pasos. Le recomendamos que siempre que lo necesite regrese aquí. Tome en cuenta que este es el único capítulo en el que se explicará a detalle la prueba de hipótesis por pasos.

Prueba t de Student vs. prueba z

Las dos pruebas que se revisarán en este capítulo se utilizan en circunstancias muy similares, para evaluar una media. Sin embargo, la principal entre ellas es que la prueba z requiere de conocer la varianza poblacional y se suele utilizar para muestras grandes. Mientras que la prueba t se utiliza para muestras pequeñas con varianza poblacional desconocida. Definir que es una muestra grande o pequeña es complicado. Por lo que en este libro la decisión para utilizar una prueba u otra será la varianza poblacional.

Prueba t de Student para una media

La prueba t de Student fue desarrollada y publicada por William Sealy Gosse en 1908 bajo el seudónimo de Student, de ahí su nombre. La prueba t de Student es una de las pruebas más utilizadas en Estadística y se basa en la distribución de Student, la cual fue descrita en el [capítulo 12](#).

De forma general, podríamos decir que existen 4 tipos de pruebas t de Student:

1. La primera prueba de Student nos permitirá evaluar el valor de una media.
2. La segunda prueba t de Student nos servirá para comparar las medias de dos grupos independientes con varianzas iguales (ver “[Prueba \$t\$ de Student para grupos independientes con varianzas iguales](#)”).
3. La tercera, nos servirá para comparar las medias dos grupos independientes sin varianzas iguales (ver “[Test de Welch](#)”).
4. La última de estas pruebas t de Student nos servirá para comparar dos grupos de forma dependiente, es decir, grupos pareados (ver “[Comparación de una variable cuantitativa en grupos dependientes: pruebas paramétricas](#)”).

En este capítulo nos centraremos en la primera de ellas. Recuerde, para las pruebas t , utilizamos la misma moneda de cambio estimada de 4 formas diferentes.

Cuando el muestreo se realiza a partir de una población que sigue una distribución normal con una varianza desconocida la estadística de prueba para $H_0 : \mu = \mu_0$ es:

$$t_{n-1} = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{n}}}$$

Donde:

- ♦ t : el valor estimado de t con $n - 1$ grados de libertad.
- ♦ $\bar{x}$: es la media de los datos que estamos trabajando o que deseamos probar.
- ♦ μ_o : es la media hipotética con la que nos vamos a comparar.
- ♦ s : es la desviación estándar muestral.
- ♦ n : la cantidad de datos de la muestra.

Pasos para resolución de la prueba t de Student para una media

Vamos a resolver el ejemplo descrito al principio del capítulo.

- ♦ **Ejemplo 15.1:** uno de los factores que se pueden asociar a complicaciones en los recién nacidos es el bajo peso al nacer. En un hospital dado se ha identificado que los bebés con un peso menor de 3000 gramos son los que más gastos de hospitalización generan, ya que son más susceptibles de presentar complicaciones. Usted es el encargado de evaluar si la media de los recién nacidos en el último año en este hospital es menor a los 3000 gramos. Para cumplir su objetivo, el encargado del hospital le entrega los datos de los 40 bebés nacidos en ese hospital en el último año. Se sabe por antecedentes previos que los datos se comportan de una forma normal. Los pesos se encuentren en el siguiente objeto:

```
Pesos ← c(2859, 2606, 3231, 2608, 2845, 3465, 3063,
          3093, 2732, 3217, 2351, 2730, 3143, 2776,
          2923, 2466, 2954, 3156, 2648, 2619, 2513,
          2918, 2977, 3159, 2883, 3017, 3195, 2632,
          3227, 2948, 2586, 3280, 2944, 2500, 2746,
          2998, 2754, 2856, 2689, 2948)
```

Utilice $\alpha = 0.05$.

Paso 1. Formulación de preguntas

- ◇ ¿Qué quiero probar? La media de los pesos de los recién nacidos es menor a 3000 gr
- ◇ ¿Cuál es el objetivo? Evaluar la media de los pesos de los recién nacidos
- ◇ ¿Con qué datos cuento?
 - » $n = 40$
 - » Media hipotética de 3000 gr
 - » Distribución normal

Paso 2. Formulación de hipótesis

- ◇ H_o : la media de los pesos de los recién nacidos es mayor o igual a 3000 gr
- ◇ H_A : la media de los de los pesos de los recién nacidos es menor a 3000 gr

Otra forma de enunciar estas hipótesis es:

- ◇ H_o : μ recién nacidos ≥ 3000 gr
- ◇ H_A : μ recién nacidos < 3000 gr

Paso 3. Estadística descriptiva y entendimiento de los datos

- ◇ Estadística descriptiva: vamos utilizar la función `summary()` y `sd()` para realizar estadística descriptiva. Dado que los datos supuestamente se comportan normalmente, es importante obtener en este caso la media y desviación estándar, las cuales son las medidas que mejor me describen a mis datos. Sin embargo, tener otros datos como el valor mínimo, máximo y otros datos no serán de ayuda para comprender nuestros datos:

```
summary(Pesos) # Resumen de los datos
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2351	2679	2900	2881	3070	3465

```
sd(Pesos) # Para obtener la desviación estándar
[1] 258.2916
```

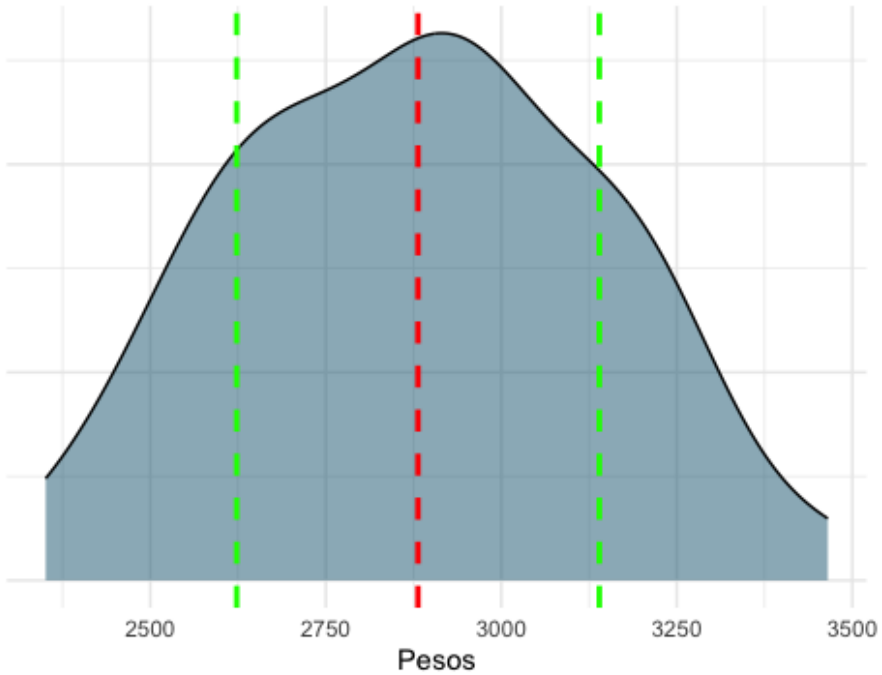
- ♦ Gráficas: en esta ocasión, vamos realizar un gráfico al que le vamos a añadir la media y las desviaciones estándar. Además, realizaremos un raincloud que nos permitirá conocer cómo es la distribución y la variación de los datos:

```
library(ggplot2) # Librería necesaria
# Calcular la media y la desviación estándar
media ← mean(Pesos)
desviacion ← sd(Pesos)

# Crear un dataframe con los datos
data ← data.frame(Pesos)

# Crear la gráfica
ggplot(data, aes(x = Pesos)) +
  geom_density(binwidth = 100,
              fill = "#19647E",
              alpha = 0.5, color = "black") +
  geom_vline(aes(xintercept = media),
             color = "red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = media + desviacion),
             color = "green", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = media - desviacion),
             color = "green", linetype = "dashed", size = 1) +
  labs(title = "Histograma de Pesos con Media y Desviación Estándar",
       x = "Pesos",
       y = "") +
  theme_minimal()+
  theme(axis.text.y = element_blank(), # Quitar los valores del eje
        axis.ticks.y = element_blank()) # Quitar los valores del eje
```

Histograma de Pesos con Media y Desviación Estándar

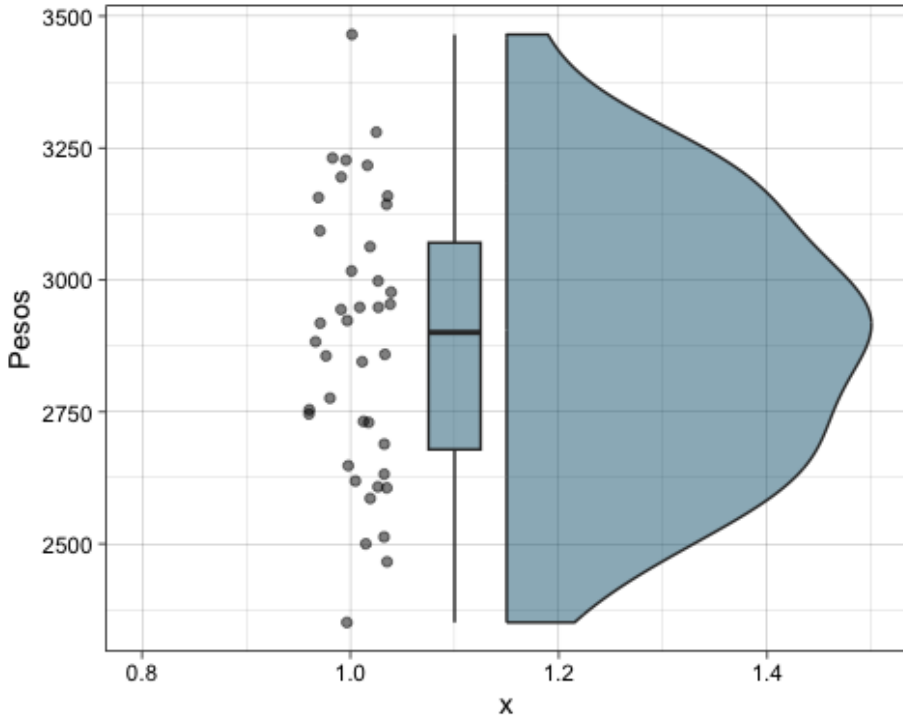


En esta gráfica nos podemos dar una idea de qué tan diferentes son los valores de 3000, pareciera que sí son menores. Además, en esta gráfica podemos ver que la distribución de los datos es muy similar a una distribución normal:

```
library(ggplot2)

Registered S3 methods overwritten by 'ggpp':
  method          from
heightDetails.titleGrob ggplot2
widthDetails.titleGrob  ggplot2

data|>
  ggplot(mapping = aes(x = 1, y = Pesos))+
  geom_rain(fill = "#19647E", alpha=0.5)+
  theme_linedraw()
```



Con el raincloud podemos darnos una idea de cómo es la variación de los datos. Con las gráficas no se identificaron valores atípicos ni datos que sugieran una evaluación extra. La información obtenida en las gráficas deberá corresponder a nuestra conclusión de la prueba de hipótesis. Por ejemplo, suponga que la media de los datos es de 4500 y en su prueba de hipótesis concluye que la media de sus datos es menor a 3000. Definitivamente algo está mal, ya que no coincide la información. La idea de hacer gráficas es confirmar su resultado de la prueba de hipótesis con lo representado en los datos y viceversa.

Paso 4. Selección de la prueba estadística

Dado que no conocemos la varianza poblacional y queremos evaluar una variable cuantitativa con distribución en un grupo, la prueba adecuada para este caso es la prueba t de Student para una media. Esto se puede comprobar en el [diagrama para la selección de pruebas](#).

Paso 5. Evaluación de los supuestos de la prueba estadística

La utilización de la prueba t de Student requiere que los datos se ajusten a una distribución normal. Por la información que se indica en el problema, podemos suponer que sí tenemos datos con una distribución normal. Sin embargo, si quisiera realizar la evaluación de la normalidad puede utilizar las técnicas revisadas en el [capítulo 13](#).

Paso 6. Valores críticos y regla de decisión

Antes de estimar los valores críticos debemos de explicar algunas funciones genéricas de R para la estimación de valores de distribuciones (moneda de cambio) y valores de probabilidad. En R para obtener el valor de una determinada distribución de probabilidad se utiliza el prefijo q seguido del sufijo de la distribución específica. Por ejemplo, para obtener los valores de para una distribución t de Student se utiliza la función $qt()$ donde q representa el valor de la distribución y la t la distribución t de la cual se obtendrá este valor. Para obtener los valores de probabilidad de una distribución utilice el prefijo p seguido del sufijo de cada distribución. Por ejemplo, para obtener el valor de probabilidad de una distribución t de Student utilice la función $pt()$. La función $qt()$ se alimenta entre otras cosas de valores de probabilidad para dar un valor de t y la función $pt()$ se alimenta de valores de para dar un valor de probabilidad. La [tabla 15.1](#) muestra las principales funciones para obtener valores de distribuciones de probabilidad y valores de p . Algunos de los argumentos solo son ilustrativos (por ejemplo Z , T , N , F).

TABLA 15.1. FUNCIONES PARA OBTENER VALORES DE UNA DISTRIBUCIÓN NORMAL Y SUS VALORES DE P

NOMBRE	PROBABILIDAD DE DENSIDAD	DISTRIBUCIÓN ACUMULADA	QUANTILE (MONEDA DE CAMBIO)
Normal	$dnorm(Z, mean, sd)$	$pnorm(Z, mean, sd)$	$qnorm(Q, mean, sd)$
Poisson	$dpois(N, lambda)$	$ppois(N, lambda)$	$qpois(Q, lambda)$
Binomial	$dbinom(N, size, prob)$	$pbinom(N, size, prob)$	$qbinom(Q, size, prob)$
Exponencial	$dexp(N, rate)$	$pexp(N, rate)$	$qexp(Q, rate)$
χ^2	$dchisq(X, df)$	$pchisq(X, df)$	$qchisq(Q, df)$
t-Student	$dt(T, df)$	$pt(T, df)$	$qt(Q, df)$
F	$df(F, df1, df2)$	$pf(F, df1, df2)$	$qf(Q, df1, df2)$

Para estimar el valor crítico vamos a utilizar la función `qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)` donde p es la valor de la probabilidad, df son los grados de libertad y `lower.tail` es un argumento para ver si el valor de t que se solicita es la parte izquierda o derecha de la distribución. Si es verdadero significa que: $P[X \leq x]$, es decir, la función devuelve la probabilidad acumulada desde el extremo inferior (izquierdo) de la distribución hasta el valor especificado. Poco a poco iremos comprendiendo estas funciones. La parte izquierda de la gráfica se utiliza cuando lo que se desea probar es si la media tiene un valor menor a x valor (3000 en este caso), mientras que la parte de la derecha de la curva se utiliza por ejemplo cuando se busca una media mayor a x valor.

Ahora sí, vamos estimar nuestro valor :

```
qt(p = 0.05, df = 39, lower.tail = T)
[1] -1.684875
```

En el código anterior se especificó $p = 0.05$ que corresponde a una hipótesis unilateral para un $\alpha = 0.05$. Para esta distribución los grados de libertad son $n - 1$, por ello especificamos $df = 39$. Dado que nuestra $H_A < 3000$ gr \$ es de nuestro interés buscar valores en la parte izquierda de la gráfica, y por tanto en este caso utilizamos `lower.tail = T`.

El valor crítico para este conjunto es -1.68, esto quiere decir que la moneda de cambio de nuestros datos deberá ser menor a este valor para poder rechazar H_0 .

Vamos a graficar estos valores para comprender más estos conceptos:

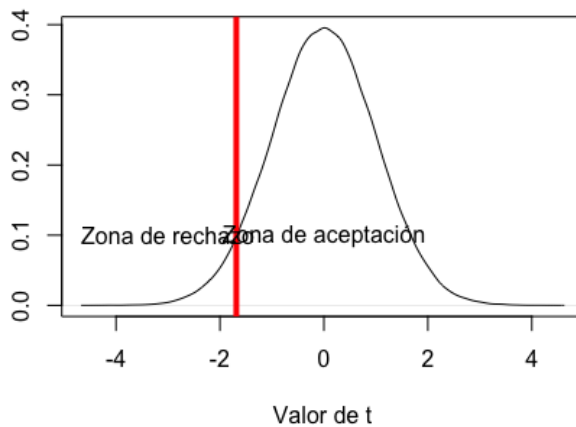
```
# Genera un gráfico de densidad para la distribución t con 100000
muestras y 99999 grados de libertad
plot(density(rt(100000, df=99999)),
      xlab = "Valor de t", ylab="",
      main = "Gráfica para regla de decisión")

# Añade una línea vertical en el valor crítico de t para el nivel de
significancia 0.05 con 39 grados de libertad
abline(v=(qt(0.05, df=39)), col="red", lw=4)
```

```
# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=-3, y=0.1, labels="Zona de rechazo")
```

Gráfica para regla de decisión



El objetivo de este código es generar una gráfica de densidad de la distribución de t y visualizar las zonas de aceptación y rechazo para una prueba de hipótesis. La línea vertical roja representa el valor crítico de t (-1.68), que separa la zona de aceptación (a la derecha de la línea) de la zona de rechazo (a la izquierda de la línea) para un nivel de significancia del 5% con 39 () grados de libertad.

La explicación del código es la siguiente:

- ♦ `rt(100000, df=99999)`: genera 100 000 muestras aleatorias de la distribución t con 99 999 grados de libertad. La distribución t con un número muy grande de grados de libertad se aproxima a la distribución normal estándar.
- ♦ `density(rt(100000, df=99999))`: calcula la densidad de las muestras generadas, que es una estimación de la función de densidad de probabilidad.
- ♦ `plot(density(rt(100000, df=99999)))`: grafica la densidad calculada.
- ♦ `xlab = "Valor de t"`: etiqueta el eje x con "Valor de t ".

- ◇ `ylab = ""`: deja el eje sin etiqueta.
- ◇ `main = "Gráfica para regla de decisión"`: título de la gráfica.
- ◇ `abline(v=(qt(0.05, df=39)), col="red", lw=4)`: añade una línea para el t estimado.
- ◇ `text(x=0, y=0.1, labels="Zona de aceptación")`: añade el texto "Zona de aceptación" en las coordenadas (0, 0.1) del gráfico.
- ◇ `text(x=-3, y=0.1, labels="Zona de rechazo")`: añade el texto "Zona de rechazo" en las coordenadas (-3, 0.1) del gráfico.

Paso 7. Estimación del estadístico de prueba

Utilizamos la fórmula:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Sustituyendo con los datos del ejercicio:

$$t = \frac{2881.375 - 3000}{258.2916/\sqrt{40}} = -2.9047$$

En R lo podemos estimar con el siguiente código:

```
(mean(Pesos)-3000)/(sd(Pesos)/sqrt(length(Pesos)))
[1] -2.904664
```

Paso 8. Decisión

Ahora debemos de comparar nuestra moneda (obtenida de nuestros datos) con la moneda de cambio de la distribución (valor crítico).

Nuestro valor de -2.9047 es menor que el valor crítico de -1.684875, por lo que podemos rechazar nuestra H_0 . Gráficamente los podemos ver:

```
# Genera un gráfico de densidad para la distribución
plot(density(rt(100000, df=99999)),
     xlab = "Valor de t", ylab="",
     main = "Gráfica para regla de decisión")

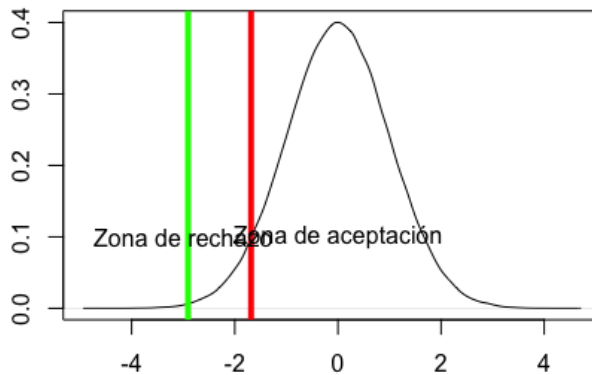
# Añade una línea vertical en el valor crítico de t
abline(v=(qt(0.05, df=39)), col="red", lw=4)
```

```
# Añade una línea vertical en el valor de t de los datos
abline(v=-2.9047, col="green", lw=4)

# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=-3, y=0.1, labels="Zona de rechazo")
```

Gráfica para regla de decisión



En esta gráfica la línea verde representa el valor de la moneda de cambio de nuestros datos, la cual cae en la zona de rechazo.

Paso 9. Intervalos de confianza

La forma más fácil de estimar los intervalos de confianza es utilizando la función `t.test()`, que explicaremos más adelante. Por ahora tome en cuenta que el IC inferior para la media es de $-\infty$ y el IC alto para la media es 2950.184. Este resultado indica que con 95% de confianza, la verdadera media de la población es menor que 2950.184. La inclusión de $-\infty$ indica que no hay un límite inferior específico en esta dirección debido a la naturaleza de la prueba de una cola.

Paso 10. Valor de p

Para estimar el valor de p de nuestra moneda de cambio utilizamos la función `pt()`, la cual se alimenta de un valor de t , en nuestro caso -2.9047 :

```
pt(q = -2.9047, df = 39, lower.tail = T)
[1] 0.003013295
```

Este valor de p indica la probabilidad de observar una media de 3 000 gr o mayor en los datos observados, un valor muy bajo.

Paso 11. Conclusión

Con 95% de confianza podemos concluir que la media del peso de los niños es menor a 3 000 gr. Esto lo podemos comprobar tanto con el valor de p y con los intervalos de confianza.

♦ **Ejemplo 15.2:** vamos a resolver el [ejemplo 15.1](#) utilizando R.

La función `t.test()` permite realizar la prueba t de Student. Esta función es un tanto peculiar, ya que permite realizar los 4 tipos de pruebas t de Student nombradas al inicio del capítulo. Para realizar una prueba de hipótesis para una media sin varianza conocida necesitará especificar en esta función los siguientes argumentos:

- » `x` es un vector numérico.
- » `alternative`: es un argumento que nos ayuda a indicarle a la función si queremos probar una hipótesis bilateral (“two.sided”), si buscamos probar que la media es mayor a una media hipotética (“greater”) o si queremos probar que una media es menor a una media hipotética (“less”).
- » `mu`: la media hipotética con la que nos queremos comparar, en este caso 3 000.
- » `conf.level`: Especifica el nivel de confianza. Por defecto está definido como 0.95.

El siguiente código le ayudará a resolver el ejemplo:

```
t.test(x=Pesos, mu = 3000, alternative = "less")
```

One Sample t-test

data: Pesos

t = -2.9047, df = 39, p-value = 0.003014

alternative hypothesis: true mean is less than 3000

95 percent confidence interval:

-Inf 2950.184

sample estimates:

mean of x

2881.375

El resultado anterior nos muestra el mismo valor de t que estimamos anteriormente, así como el mismo valor p . Además, presenta los IC al 95% para esta media que es consistente con el valor p y con el rechazo de la H_0 .

♦ **Ejemplo 15.3:** ahora resolveremos el [ejemplo 15.1](#) planteando una hipótesis bilateral. Es decir, ahora nos interesa probar si la media de los recién nacidos es diferente de 3000. Nuestras hipótesis ahora son:

» $H_0: \mu \text{ recién nacidos} = 3000 \text{ gr}$

» $H_A: \mu \text{ recién nacidos} \neq 3000 \text{ gr}$

En R solo debemos de cambiar el argumento `alternative` para realizar una prueba de hipótesis bilateral:

```
t.test(x=Pesos, mu = 3000, alternative = "two.sided")
```

One Sample t-test

data: Pesos

t = -2.9047, df = 39, p-value = 0.006027

alternative hypothesis: true mean is not equal to 3000

95 percent confidence interval:

2798.769 2963.981

```
sample estimates:
mean of x
2881.375
```

Note cómo ahora, en lugar de tener un valor de p de 0.003, tenemos un valor del doble, ya que estamos repartiendo el valor de t tanto para la diferencia positiva como para la negativa. Por lo tanto, necesitamos estimar un valor t positivo (a la derecha de la distribución) y otro negativo (a la izquierda de la distribución), y a ambos estimarles su respectivo valor de p , el cual fue de 0.003 para cada uno.

Para representar gráficamente la zona de rechazo y aceptación podemos utilizar el siguiente código:

```
# Genera un gráfico de densidad para la distribución
plot(density(rt(100000, df=99999)),
     xlab = "Valor de t", ylab="",
     main = "Gráfica para regla de decisión")

# Añade una línea vertical en el valor crítico de t
abline(v=(qt(0.025, df=39)), col="red", lw=4) # Negativo
abline(v=(qt(0.025, df=39, lower.tail = F)),
     col="red", lw=4) # Negativo

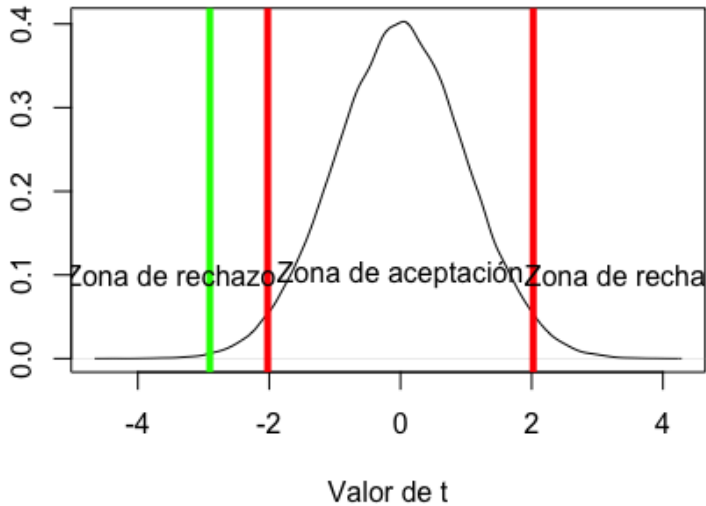
# Añade una línea vertical en el valor de t de los datos
abline(v=-2.9047, col="green", lw=4)

# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=-3.5, y=0.1, labels="Zona de rechazo") # Izquierda

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=3.5, y=0.1, labels="Zona de rechazo") # Derecha
```

Gráfica para regla de decisión



Ahora hay dos zonas de rechazo: una para los valores positivos y otra para los valores negativos. Esto sería el equivalente a buscar una diferencia negativa y una diferencia positiva; en nuestra hipótesis solo buscamos una diferencia, ya sea positiva o negativa. Además, note cómo ahora al añadir las líneas verticales rojas, en lugar de $p=0.05$ utilizamos $p=0.025$. El valor de α se reparte en ambos lados. Esto también hace que el valor crítico de t pase de -1.68 para la hipótesis unilateral del [ejemplo 15.1](#) a -2.02 , lo cual a su vez también va en relación al valor de p .

Prueba z

La segunda opción que tenemos para evaluar el valor de una media hipotética es la prueba z . Esta prueba requiere conocer el valor de la varianza poblacional, por lo que su uso es limitado. Para esta prueba el estadístico de prueba se estima mediante:

$$Z = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Donde: $\bar{x}$ es la media de mis datos, μ_0 es la media hipotética con la que me quiero comparar, σ es la desviación estándar poblacional y n el número de datos.

- ♦ **Ejemplo 15.4:** suponga ahora que junto con los datos que le otorga el encargado del hospital se encuentra la σ histórica, la cual es de 280 gr. Debe entonces repetir la prueba de hipótesis esta vez utilizando la prueba z. Utilice las siguientes hipótesis:

- » $H_0: \mu \text{ recién nacidos} \geq 3000 \text{ gr}$
- » $H_A: \mu \text{ recién nacidos} < 3000 \text{ gr}$

Utilice $\alpha = 0.05$.

En R se estima con el siguiente código:

```
# Si lo requiere instale la librería BSDA
BSDA::z.test(x = Pesos, alternative = "less",
             mu = 3000, sigma.x = 280)

One-sample z-Test

data: Pesos
z = -2.6795, p-value = 0.003687
alternative hypothesis: true mean is less than 3000
95 percent confidence interval:
    NA 2954.196
sample estimates:
mean of x
2881.375
```

El código anterior utiliza la función `z.test` del paquete `BSDA` en R para realizar una prueba. La explicación del código es la siguiente:

- ◇ `BSDA::z.test`: esta es la llamada a la función `z.test` del paquete `BSDA`. La función `z.test` se utiliza para realizar pruebas Z.
- ◇ `x = Pesos`: aquí se especifica el conjunto de datos sobre el que se realizará la prueba. `Pesos` es el vector que contiene los datos muestrales, en este caso, los pesos de los recién nacidos.
- ◇ `alternative = "less"`: este argumento define la hipótesis alternativa de la prueba. En este caso, `alternative = "less"` indica que la hipótesis alternativa es que la media de los pesos es menor que el valor especificado en , es decir, .
- ◇ `mu = 3000`: este es el valor de la media poblacional bajo la hipótesis nula (H_0). En esta prueba, estamos verificando si la media de los pesos es menor que 3000 gramos.
- ◇ `sigma.x = 280`: este es el valor de la desviación estándar poblacional de los pesos. En las pruebas Z, se asume que se conoce la desviación estándar de la población.

Ahora, una explicación de los resultados es la siguiente:

- ◇ El valor de $z = -2.6795$ es valor de z (mi moneda de cambio) de mis datos calculada con la fórmula:

$$Z = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

- ◇ El valor de p indica la probabilidad de observar resultados iguales o más extremos, cuando la H_0 es verdadera.
- ◇ Los intervalos de confianza con 95% de confianza, la verdadera media de la población es menor que 2954.196.

- ◇ **Ejemplo 15.5:** ahora vamos a resolver el problema anterior planteando una hipótesis bilateral.

Las hipótesis ahora son:

- » $H_0: \mu \text{ recién nacidos} = 3000 \text{ gr}$
- » $H_A: \mu \text{ recién nacidos} \neq 3000 \text{ gr}$

Y el código de R es:

```
BSDA::z.test(x = Pesos, alternative = "two.sided",
             mu = 3000, sigma.x = 280)
```

One-sample z-Test

```
data: Pesos
z = -2.6795, p-value = 0.007374
alternative hypothesis: true mean is not equal to 3000
95 percent confidence interval:
 2794.604 2968.146
sample estimates:
mean of x
 2881.375
```

Note cómo el valor de p es el del doble que el del [ejemplo 15.4](#).

Ejercicios

Los ejercicios de este capítulo serán desarrollados en su totalidad, es decir, se mostrará código de R y su resultado.

- ♦ **Ejercicio 15.1:** un investigador está interesado en determinar si un nuevo suplemento dietético influye significativamente en el nivel promedio de hierro en la sangre de individuos sanos. Basándose en estudios previos, se conoce que el nivel medio de hierro en la población general es de $60 \mu\text{g/dL}$ y que la desviación estándar es de $63 \mu\text{g/dL}$.

Se ha administrado el suplemento dietético a una muestra aleatoria de 50 individuos y, después de un mes de tratamiento, se ha medido el nivel de hierro en la sangre. La media de hierro en esta muestra es de $63 \mu\text{g/dL}$.

Con estos datos es posible concluir que en los individuos que recibieron el suplemento las concentraciones de hierro son mayores a la de la población. Utilice un nivel de confianza del 99% y asuma que los datos provienen de una distribución normal.

Resolución ejercicio 15.1

Dado que se conoce la variación poblacional se puede utilizar la prueba .

Las hipótesis son:

- ♦ H_0 : individuos con el suplemento tienen concentraciones de hierro iguales o menores a la de la población
- ♦ $H_0\mu \leq 60$
- ♦ H_1 : individuos con el suplemento tienen concentraciones de hierro mayores a la población
- ♦ $H_1\mu > 60$

Para obtener mi valor crítico:

```
qnorm(p = 0.01, lower.tail = F)
[1] 2.326348
```

El valor de Z de mis datos:

$$Z = \frac{63 - 60}{\frac{6}{\sqrt{50}}} = 3.54$$

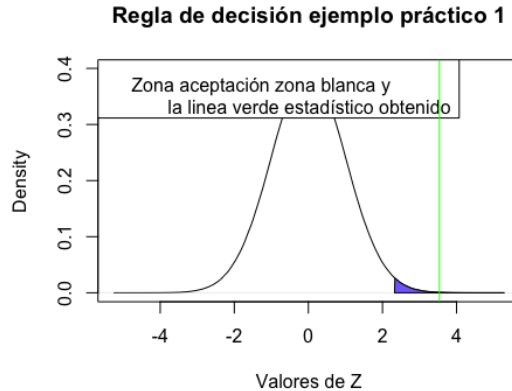
En R lo podemos estimar:

```
(63-60)/(6/sqrt(50))
[1] 3.535534
```

Para la gráfica de decisión se puede utilizar el siguiente código, que es una variación al revisado en el capítulo:

```
y <- (rnorm(10000000, mean=0, sd=1))
den <- density(y)
plot(den, main="Regla de decisión ejemplo práctico 1",
     xlab="Valores de Z")
value <- qnorm(p = 0.01, lower.tail = F)
polygon(c(den$x[den$x ≥ value ], value),
       c(den$y[den$x ≥ value ], 0),
```

```
col = "slateblue1",
border = 1)
legend(x="topleft", legend = "Zona aceptación zona blanca y
la linea verde estadístico obtenido")
abline(v=(63-60)/(6/sqrt(50)), col="green")
```



También puede utilizar este código, que es igual al revisado en el capítulo:

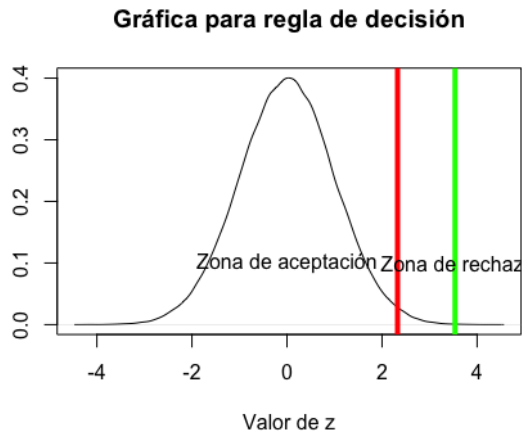
```
# Genera un gráfico de densidad para la distribución
plot(density(rnorm(100000, mean=0, sd=1)),
      xlab = "Valor de z", ylab="",
      main = "Gráfica para regla de decisión")

# Añade una línea vertical en el valor crítico de z
abline(v=(qnorm(p = 0.01, mean = 0, sd = 1, lower.tail = F)),
      col="red", lw=4)

# Añade una línea vertical en el valor de z de los datos
abline(v=(63-60)/(6/sqrt(50)), col="green", lw=4)

# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")
```

```
# Añade texto en el gráfico para indicar la zona de rechazo
text(x=3.6, y=0.1, labels="Zona de rechazo") # Izquierda
```



Nuestro valor de Z cae en la zona de rechazo, por lo que tenemos argumentos para rechazar la H_0 .

Para estimar el valor de p se utiliza el siguiente código:

```
pnorm(q = (63-60)/(6/sqrt(50)), lower.tail = F)
[1] 0.000203476
```

Dado que se busca probar si la media de mis datos es mayor a la media poblacional ($P[X \leq x]$) se debe elegir `lower.tail=F`.

- ♦ **Ejercicio 15.2:** en una planta de producción de medicamentos, se requiere que los comprimidos de un fármaco tengan un peso medio de 500 miligramos. Se sabe por especificaciones de calidad que la desviación estándar histórica del peso de los comprimidos es de 50 miligramos. Un control de calidad sospecha que una de las máquinas está descalibrada y está produciendo comprimidos con un peso diferente al estándar.

Usted es parte del equipo de aseguramiento de calidad y se le ha encargado realizar una prueba estadística para determinar si la máquina en cuestión está funcionando correctamente. Para ello, se toma una muestra aleatoria de 40 comprimidos producidos por la máquina y se encuentra que el peso medio de esta muestra es de 510 miligramos.

Determine si la máquina necesita recalibración. Utilice $\alpha = 0.05$ y asuma que los datos provienen de una distribución normal y que la desviación estándar histórica es la poblacional.

Resolución ejercicio 15.2

Dado que se conoce la variación poblacional se puede utilizar la prueba . Las hipótesis son:

- » H_0 : el equipo se encuentra calibrado con respecto al peso medio del fármaco de 500 mg. No necesita recalibración
- » $H_0: \mu = 500$ mg
- » H_1 : el equipo se encuentra descalibrado con respecto al peso medio del fármaco de 500 mg. Necesita recalibración
- » $H_0: \mu \neq 500$ mg

Para obtener mi valor crítico:

```
qnorm(p = 0.025, lower.tail = F)
[1] 1.959964
qnorm(p = 0.025, lower.tail = T)
[1] -1.959964
```

El valor de Z de mis datos:

$$Z = \frac{510 - 500}{\frac{50}{\sqrt{40}}} = 1.26$$

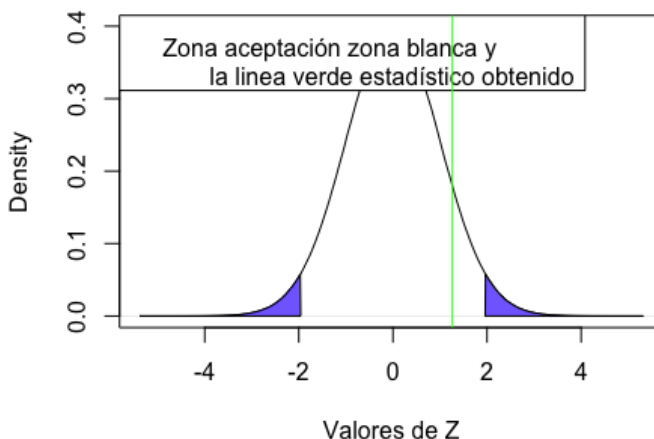
En R lo podemos estimar:

```
(510-500)/(50/sqrt(40))
[1] 1.264911
```

Para la gráfica de decisión se puede utilizar el siguiente código:

```
y ← (rnorm(10000000, mean=0, sd=1))
den ← density(y)
plot(den, main="Regla de decisión ejemplo práctico 1",
     xlab="Valores de Z")
value ← qnorm(p = 0.025, lower.tail = F)
polygon(c(den$x[den$x ≥ value ], value),
        c(den$y[den$x ≥ value ], 0),
        col = "slateblue1",
        border = 1)
value ← qnorm(p = 0.025, lower.tail = T)
polygon(c(den$x[den$x ≤ value ], value),
        c(den$y[den$x ≤ value ], 0),
        col = "slateblue1",
        border = 1)
legend(x="topleft", legend = "Zona aceptación zona blanca y
                             la linea verde estadístico obtenido")
abline(v=(510-500)/(50/sqrt(40)), col="green")
```

Regla de decisión ejemplo práctico 1



Nuestro valor de Z cae en la zona de aceptación, por lo que no tenemos argumentos para rechazar la H_0 .

Para estimar el valor de p se utiliza el siguiente código:

```
pnorm(q = (510-500)/(50/sqrt(40)), lower.tail = F)*2
[1] 0.2059032
```

- ♦ **Ejercicio 15.3:** la base de datos `Pima.tr2` contiene datos sobre el grupo de mujeres del pueblo de Pima y varios predictores de diabetes. Uno de los atributos medidos en esta base de datos es la glucosa en plasma a 2 horas en una prueba de tolerancia oral a la glucosa.

Se quiere investigar si el nivel medio de glucosa en plasma a 2 horas en el grupo de los indios Pima es significativamente diferente del valor de glucosa considerado como punto de corte para diagnóstico de diabetes, que es de 140 mg/dL. La desviación estándar de la glucosa en la población es conocida y es de 30 mg/dL.

Determine si hay una diferencia significativa entre la media muestral y el valor de corte de 140 mg/dl. Utilice $\alpha = 0.05$ y asuma que los datos provienen de una distribución normal.

Resolución ejercicio 15.3

Se conoce la desviación estándar poblacional, por lo que se debe de emplear la prueba .

- ♦ H_0 : la glucosa en plasma a 2 horas en las mujeres Pima es igual a 140 mg/dL
- ♦ $H_0: \mu = 140$
- ♦ H_1 : la glucosa en plasma a 2 horas en las mujeres Pima no es igual a 140 mg/dL
- ♦ $H_1: \mu \neq 140$

```
# Importar base
library(MASS)
data(Pima.tr2)
library(BSDA)
Loading required package: lattice
```

```
Attaching package: 'BSDA'
```

The following object is masked from ‘package:datasets’:

Orange

```
z.test(x = Pima.tr2$glu, alternative = "two.sided",
       mu = 140, sigma.x = 30)
```

One-sample z-Test

data: Pima.tr2\$glu

z = -9.3858, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 140

95 percent confidence interval:

120.3486 127.1381

sample estimates:

mean of x

123.7433

- ♦ **Ejercicio 15.4:** la base de datos birthwt del paquete MASS incluye datos de un estudio sobre factores de riesgo para el bajo peso al nacer. Entre las variables registradas está el peso del recién nacido en gramos (bwt).

Se desea examinar si el peso promedio al nacer en la muestra proporcionada es significativamente diferente del promedio nacional que se considera normal, el cual es de 3 400 gramos. Se conoce que la desviación estándar del peso al nacer a nivel nacional es de 500 gramos.

Realice una prueba z para determinar si existe una diferencia significativa entre la media muestral y el promedio nacional de 3 400 gramos.

Resolución ejercicio 15.4

Se conoce la desviación estándar poblacional, por lo que se debe de emplear la prueba .

- ♦ H_0 : el peso promedio al nacer de los recién nacidos de la base birthwt es igual de 3 400 gr

- ♦ $H_1: \mu = 3400$
- ♦ H_0 : el peso promedio al nacer de los recién nacidos de la base birthwt es diferente de 3400 gr
- ♦ $H_1: \mu \neq 3400$

```
library(MASS)
data("birthwt")
z.test(x = birthwt$bwt, mu = 3400,
       sigma.x = 500)
```

One-sample z-Test

```
data: birthwt$bwt
z = -12.522, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 3400
95 percent confidence interval:
 2873.304 3015.871
sample estimates:
mean of x
 2944.587
```

- ♦ **Ejercicio 15.5:** algunos investigadores suponen que la media de la temperatura normal del cuerpo es menor a 98.6 °F. ¿Existe evidencia estadística que demuestre que los investigadores tienen la razón? Para resolver este problema utilice la base de datos “BodyTemperature.txt”, que puede descargar [aquí](#). Utilice $\alpha = 0.05$.

Utilizando la base de datos “BodyTemperature.txt” evalúe:

- ♦ La hipótesis de que la media de la frecuencia cardíaca de la población es inferior a 75.
- ♦ La hipótesis de que la media de la frecuencia cardíaca de la población es diferente de 75.

Resolución ejercicio 15.5

```
df <- read.csv("Bases/BodyTemperature.txt", sep = "")
```

```
t.test(df$HeartRate, mu = 75, alternative = "less")
```

One Sample t-test

data: df\$HeartRate

t = -2.5222, df = 99, p-value = 0.006629

alternative hypothesis: true mean is less than 75

95 percent confidence interval:

-Inf 74.54215

sample estimates:

mean of x

73.66

```
t.test(df$HeartRate, mu = 75, alternative = "two.sided")
```

One Sample t-test

data: df\$HeartRate

t = -2.5222, df = 99, p-value = 0.01326

alternative hypothesis: true mean is not equal to 75

95 percent confidence interval:

72.60581 74.71419

sample estimates:

mean of x

73.66

- ♦ **Ejercicio 15.6:** una de las variables registradas en `Pima.tr2` es la presión sanguínea diastólica (bp). Se desea investigar si la presión sanguínea diastólica media de las mujeres en el estudio difiere del valor estándar considerado normal de 80 mmHg.

Determine si la presión sanguínea diastólica media en la muestra de Pima.tr2 es significativamente diferente de 80 mmHg. Utilice $\alpha = 0.05$ y asume una distribución normal. Resuelva este ejercicio por pasos.

Resolución ejercicio 15.6

```
library(MASS)
data("Pima.tr2")
```

Paso 1

Evaluar si la media de la presión sanguínea diastólica es diferente de 80. En el problema no se establece un valor para la variación poblacional y se trata de una hipótesis bilateral.

Paso 2

- ◇ $H_o: \mu$ Presión diastólica = 800 mmHg
- ◇ $H_A: \mu$ Presión diastólica $\neq$ 800 mmHg

Paso 3

```
summary(Pima.tr2$bp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.    NA's
 38.00  64.00   72.00   72.32  80.00  114.00     13

sd(Pima.tr2$bp)
[1] NA

# Cargar ggplot2
library(ggplot2)
library(gggrain)

# Calcular la media y la desviación estándar de bp en Pima.tr2
desviacion <- sd(Pima.tr2$bp, na.rm = TRUE)
media <- mean(Pima.tr2$bp, na.rm = TRUE)

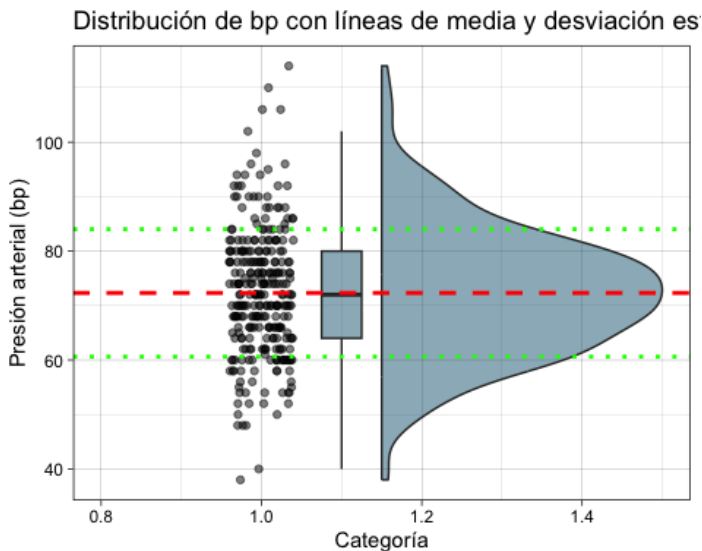
# Crear el gráfico con las líneas horizontales para la media y la
desviación estándar
Pima.tr2 |>
```

```
ggplot(mapping = aes(x = 1, y = bp)) +
  geom_rain(fill = "#19647E", alpha = 0.5) +
  theme_linedraw() +
  geom_hline(aes(yintercept = media),
             color = "red", linetype = "dashed", size = 1) +
  geom_hline(aes(yintercept = media + desviacion),
             color = "green", linetype = "dotted", size = 1) +
  geom_hline(aes(yintercept = media - desviacion),
             color = "green", linetype = "dotted", size = 1) +
  labs(title = "Distribución de bp con líneas de media y desviación estándar",
       x = "Categoría",
       y = "Presión arterial (bp)")
```

Warning: Removed 13 rows containing non-finite outside the scale range (`stat\_half\_ydensity()`).

Warning: Removed 13 rows containing non-finite outside the scale range (`stat\_boxplot()`).

Warning: Removed 13 rows containing missing values or values outside the scale range (`geom\_point\_sorted()`).



Paso 4

Prueba t para una media.

Paso 5

Se cumplen los supuestos.

Paso 6

```
# Estimación de los valores de t
qt(p=0.025, df = 286, lower.tail = T) → alto
qt(p=0.025, df = 286, lower.tail = F) → bajo

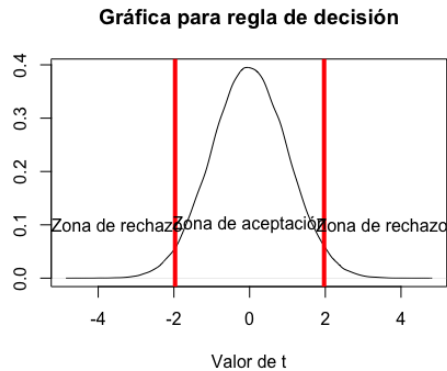
# Mostrar los valores de t
print(c(alto, bajo))
[1] -1.968293  1.968293

# Genera un gráfico de densidad para la distribución t con 100000
muestras y 99999 grados de libertad
plot(density(rt(100000, df=99999)),
      xlab = "Valor de t", ylab="", main = "Gráfica para regla de decisión")

# Añade una línea vertical en el valor crítico de t para el nivel de
significancia 0.05 co
abline(v=alto, col="red", lw=4)
abline(v=bajo, col="red", lw=4)

# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=-3.5, y=0.1, labels="Zona de rechazo")
text(x=3.5, y=0.1, labels="Zona de rechazo")
```



Paso 7

-3.8782

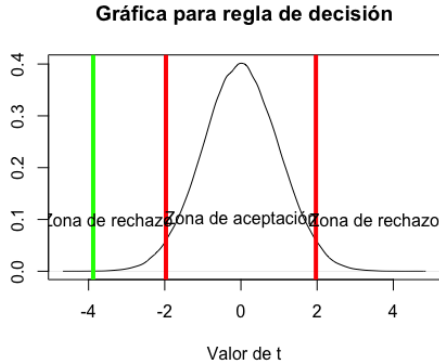
Paso 8

```
# Genera un gráfico de densidad para la distribución t con 100000
muestras y 99999 grados de libertad
plot(density(rt(100000, df=99999)),
      xlab = "Valor de t", ylab="",
      main = "Gráfica para regla de decisión")

# Añade una línea vertical en el valor crítico de t para el nivel de
significancia 0.05 co
abline(v=alto, col="red", lw=4)
abline(v=bajo, col="red", lw=4)
abline(v=-3.8782, col="green", lw=4)

# Añade texto en el gráfico para indicar la zona de aceptación
text(x=0, y=0.1, labels="Zona de aceptación")

# Añade texto en el gráfico para indicar la zona de rechazo
text(x=-3.5, y=0.1, labels="Zona de rechazo")
text(x=3.5, y=0.1, labels="Zona de rechazo")
```

Paso 9

De 70.96068 a 73.68044.

Paso 10

```
pt(q = -3.8782, df=286, lower.tail = T)*2
[1] 0.0001306212
```

Paso 11

Con 95% la media para la bp es diferente de 75.

En R:

```
t.test(x = Pima.tr2$bp, mu = 75, alternative = "two.sided")
```

One Sample t-test

data: Pima.tr2\$bp

t = -3.8782, df = 286, p-value = 0.0001306

alternative hypothesis: true mean is not equal to 75

95 percent confidence interval:

70.96068 73.68044

sample estimates:

mean of x

72.32056

- ♦ **Ejercicio 15.7:** repita el ejercicio anterior, pero esta vez pruebe si la bp es significativamente diferente de 72 mmHg. Utilice $\alpha = 0.05$ y asuma una distribución normal. ¿Qué pasa si se cambia a un nivel de confianza del 90%?

Resolución ejercicio 15.7

```
t.test(x = Pima.tr2$bp, mu = 72, alternative = "two.sided")
```

One Sample t-test

data: Pima.tr2\$bp

t = 0.46398, df = 286, p-value = 0.643

alternative hypothesis: true mean is not equal to 72

95 percent confidence interval:

70.96068 73.68044

sample estimates:

mean of x

72.32056

Si modifica el nivel de confianza, se modifican los IC. El valor de p es el mismo:

```
t.test(x = Pima.tr2$bp, mu = 72,
       alternative = "two.sided", conf.level = 0.90)
```

One Sample t-test

data: Pima.tr2\$bp

t = 0.46398, df = 286, p-value = 0.643

alternative hypothesis: true mean is not equal to 72

90 percent confidence interval:

71.18045 73.46067

sample estimates:

mean of x

72.32056

- ♦ **Ejercicio 15.8:** este ejercicio se centrará en las pacientes diagnosticadas con diabetes y explorará si su nivel medio de glucosa difiere del valor de referencia de 126 mg/dL, que es el umbral para el diagnóstico de diabetes según la Organización Mundial de la Salud (OMS). Realice lo siguiente:
1. Cargar la base de datos `Pima.tr2` del paquete `MASS`.
 2. Filtrar los datos para incluir solo a las pacientes con diabetes (variable `type` igual a "Yes").
 3. Seleccionar la variable `glu` para análisis.
 4. Formular las hipótesis nula y alternativa.
 5. Realizar una prueba *t* de una muestra para evaluar, si la media de glucosa en plasma en la muestra filtrada es significativamente diferente de 126 mg/dL.
 6. Establecer un nivel de significancia del 5% ($\alpha = 0.05$).
 7. Interpretar el resultado de la prueba.

Resolución ejercicio 15.8

```
Pima_dm <- Pima.tr2[Pima.tr2$type=="Yes",]
t.test(Pima_dm$glu, mu = 126)
```

One Sample t-test

```
data: Pima_dm$glu
t = 6.2282, df = 105, p-value = 9.911e-09
alternative hypothesis: true mean is not equal to 126
95 percent confidence interval:
 137.8386 148.8972
sample estimates:
mean of x
 143.3679
```

- ♦ **Ejercicio 15.9:**
- » ¿Qué indica un valor de 0.05 ?
 - » ¿Qué indica un valor de 0.10 ?
 - » ¿Qué indica un valor de 0.001 ?

Referencias

- Banda, G. (2018). A Brief Review of Independent, Dependent and One Sample t-test. *International Journal of Applied Mathematics and Theoretical Physics*, 4(2), 50-54. <https://doi.org/10.11648/j.ijamtp.20180402.13>
- Celis de la Rosa, A. de J. (2014). Pruebas de hipótesis para una y dos variables cuantitativas. En A. de J. Celis de la Rosa y V. Labrada Martagón, *Bioestadística* (3ª ed., pp. 135-143). Editorial El Manual Moderno.
- Hespanhol, L., Vallio, C S., Costa, L D C M. y Saragiotto, B T. (2019). Understanding and interpreting confidence and credible intervals around effect estimates. *Elsevier BV*, 23(4), 290-301. <https://doi.org/10.1016/j.bjpt.2018.12.006>
- Indrayan, A. y Malhotra, R. K. (2017). Inference from Means. En *Medical Biostatistics* (4ª ed., pp. 377-422). Chapman & Hall/CRC Biostatistics Series.
- Kim, J S. y Dailey, R J. (2007). Hypothesis Testing: One--Sample Case. En *Biostatistics for Oral Healthcare* (pp. 127-157). <https://doi.org/10.1002/9780470388303.ch8>
- King, G. (1986). *How Not to Lie with Statistics: Avoiding Common Mistakes in Quantitative Political Science*. Wiley-Blackwell, 30(3). <https://doi.org/10.2307/2111095>
- Rigby, A. (1999). Getting past the statistical referee: moving away from P-values and towards interval estimation. *Oxford University Press*, 14(6), 713-715. <https://doi.org/10.1093/her/14.6.713>
- Seoane, T., Martín, J., Martín-Sánchez, E., Lurueña-Segovia, S. y Moreno, F. A. (2007). Capítulo 7: Estadística: Estadística Descriptiva y Estadística Inferencial. *Elsevier BV*, 33(9), 466-471. [https://doi.org/10.1016/s1138-3593\(07\)73945-x](https://doi.org/10.1016/s1138-3593(07)73945-x)
- Weir, J. B. D. V. (1960). Significance of the Difference between Two Means when the Population Variances may be Unequal. *Nature Portfolio*, 187(4735), 438-438. <https://doi.org/10.1038/187438a0>
- Xu, M., Fralick, D., Zheng, J Z., Wang, B., Tu, X. y Feng, C. (2017). The Differences and Similarities Between Two-Sample T-Test and Paired T-Test. *National Institutes of Health*, 29(3), 184-188.

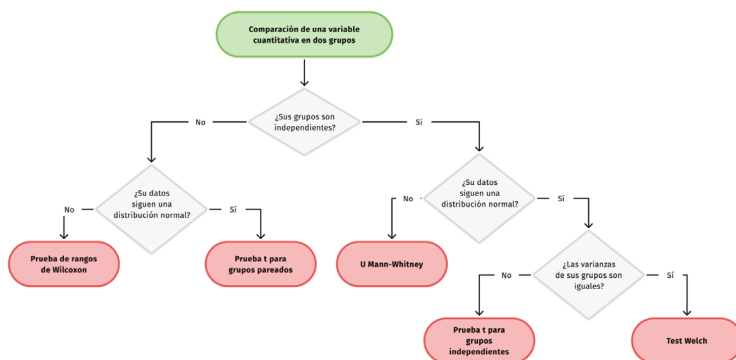
Pruebas de hipótesis para la comparación de variables cuantitativas en dos grupos

Edsaúl Emilio Pérez Guerrero

En este capítulo nos enfocaremos a revisar las pruebas de hipótesis para la comparación de una variable cuantitativa en dos grupos. Primero, revisaremos la prueba t de Student para comparar dos grupos independientes con variancias iguales; posteriormente, nos enfocaremos en el test de Welch, una prueba nos ayuda a comparar dos medias en dos grupos con variancias diferentes. En seguida nos enfocaremos en la prueba t de Student para grupos pareados (dependientes) y finalizaremos el capítulo revisando las versiones no paramétricas de estas pruebas.

La [figura 16.1](#) muestra una visión general sobre las pruebas que se revisarán en este capítulo. Además, permite la selección de la prueba estadística correcta basada en la normalidad de los datos y las características de los grupos.

FIGURA 16.1. PRUEBAS PARA LA COMPARACIÓN DE UNA VARIABLE CUANTITATIVA EN DOS GRUPOS



Otro aspecto importante que se abordará aquí es el del tamaño del efecto, una medida estadística que nos permite evaluar que tan grandes son o no las diferencias.

Comparación de una variable cuantitativa en grupos independientes: pruebas paramétricas

En el capítulo anterior vimos que hay 4 pruebas t de Student. Específicamente, en esta sección nos enfocaremos en dos:

- ♦ Prueba t de Student para grupos independientes con varianzas iguales
- ♦ Test de Welch (Prueba t de Student para grupos independientes con varianzas diferente)

La prueba t de Student para muestras independientes se utiliza para la comparación de medias de grupos independientes. En Estadística, un grupo independiente se refiere a una muestra o población que no está relacionada o influenciada por otra muestra o población en un estudio. Presenta las siguientes características:

- ♦ No hay conexión entre los grupos
- ♦ Los resultados de un grupo no afectan a los otros
- ♦ Se utilizan en estudios comparativos

Cuando se busca comparar dos medias las hipótesis que se pueden plantear son:

	H_0	H_A	TIPO DE HIPÓTESIS
1	$\mu_1 - \mu_2 = 0$	$\mu_1 - \mu_2 \neq 0$	Bilateral
2	$\mu_1 - \mu_2 \geq 0$	$\mu_1 - \mu_2 < 0$	Unilateral
3	$\mu_1 - \mu_2 \leq 0$	$\mu_1 - \mu_2 > 0$	Unilateral

Tome en cuenta que la formulación de hipótesis del punto 2 y 3 de la tabla anterior depende de a quién le sea asignado el grupo 1 y a quién el grupo 2, por tanto, las hipótesis se pueden invertir.

Prueba t Student para grupos independientes con varianzas iguales

Cuando no se conoce el valor de la varianza poblacional se puede utilizar la prueba *t* – *student*. Sin embargo, el tipo de estadístico dependerá si las varianzas muestrales son iguales o no.

Para varianzas iguales el valor del estadístico *t* se estima:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)_0}{\sqrt{\frac{s_p^2}{n_1} + \frac{s_p^2}{n_2}}}$$

Donde:

- ♦ $\bar{x}$: diferencia entre las medias muestrales de los dos grupos.
- ♦ $\bar{x}_1$: es la media muestral del primer grupo.
- ♦ $\bar{x}_2$ es la media muestral del segundo grupo.
- ♦ $(\mu_1 - \mu_2)_0$: diferencia hipotética entre las medias poblacionales bajo la hipótesis nula. Este valor se convierte en 0, dado que provienen de la misma población.
- ♦ S_p^2 : varianza combinada (o varianza ponderada) de las dos muestras. Se calcula como una combinación de las varianzas de las dos muestras. Se calcula como:

$$\Rightarrow S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

- ♦ S_1^2 es la varianza de la primera muestra.
- ♦ S_2^2 es la varianza de la segunda muestra.
- ♦ n_1 y n_2 son los tamaños de las muestras 1 y 2, respectivamente.

Cuando H_0 es verdadera, sigue una distribución *t* de Student con $n_1 + n_2 - 2$ grados de libertad.

Función `t.test()`

En R se utiliza la función `t.test()`. Esta función puede ser alimentada de diversas formas, las cuales vamos a explicar enseguida.

- ❖ **Opción 1:** la función `t.test()` puede ser alimentada con dos vectores numéricos, uno que contendría los valores del grupo 1 y otro con los valores numéricos del grupo 2. Estos valores se especifican dentro de la función como `x` y `y`. El código quedaría de la siguiente manera:

```
# Datos de ejemplo
x <- c(5.1, 4.9, 6.3, 5.8, 5.6) # Muestra 1
y <- c(6.2, 5.8, 6.5, 6.1, 6.3) # Muestra 2

# Realizar la prueba t para dos muestras independientes
result <- t.test(x, y, var.equal = TRUE)

# Imprimir el resultado
print(result)
```

- ❖ **Opción 2:** utilizar la función `t.test()` mediante la fórmula. Esta opción se utiliza cuando se tiene una variable con los datos numéricos de ambos grupos y otra variable que me permite agrupar estos datos. Esta es la opción que se utiliza en la mayoría de los casos cuando se toman datos de un *Dataframe*, por ejemplo: `t.test(Variable de prueba~Variable agrupación)`. Símbolo `~` podría decirse que indica agrupación. Esta sería la opción que se utilizaría si quisiera comparar la concentración de glucosa entre las mujeres de Pima. El siguiente código ejemplifica esta situación:

```
library(MASS) # Cargar Dataframe
data("Pima.tr2")

t.test(Pima.tr2$glu~Pima.tr2$type, var.equal = TRUE)
```


La expresión `t.test(Pima.tr2$glu~Pima.tr2$type)` puede leerse como prueba t de muestras independientes para la glucosa en ayuno `glu` agrupada por la presencia o ausencia de diabetes `type`.

El argumento `var.equal` es sumamente importante, ya que es el que permite indicarle a R que la prueba que queremos realizar es para varianzas iguales cuando es verdadero y el test de Welch cuando es falso.

Prueba t Student para grupos independientes con varianzas iguales en R

- ♦ **Ejemplo 16.1:** un grupo de investigadores desea comparar las concentraciones de un biomarcador, entre pacientes con y sin hipertensión. Su intención es demostrar que las concentraciones del biomarcador son diferentes en ambos grupos. Las hipótesis para este estudio son las siguientes:
 - » $H_0: \mu \text{ biomarcador pacientes con hipertensión} = \mu \text{ biomarcador pacientes sin hipertensión}$
 - » $H_A: \mu \text{ biomarcador pacientes con hipertensión} \neq \mu \text{ biomarcador pacientes sin hipertensión}$

O bien, si consideramos a los pacientes con hipertensión como grupo 1 y a los pacientes sin hipertensión como grupo 2, la hipótesis podría plantearse:

$$H_0: \mu_1 - \mu_2 = 0 \quad H_A: \mu_1 - \mu_2 \neq 0$$

Los datos de estos biomarcadores se encuentran en dos vectores que se muestran en el siguiente código:

```
Biomarcador_hta <- c(33.2, 27.8, 22.6, 53.0, 34.8, 30.6, 31.6,
                    24.6, 37.2, 24.0, 48.2, 33.0, 43.6, 32.6,
                    46.8, 37.6)
Biomarcador_sanos <- c(36.2, 12.0, 21.6, 22.0, 15.4, 35.8, 17.0,
                      26.0, 37.8)
```

Utilice $\alpha = 0.02$, asuma que las varianzas son iguales en ambos grupos y que siguen una distribución normal.

De acuerdo con las características de los datos y el objetivo del estudio la opción indicada para evaluar la hipótesis es la prueba t para muestras independientes con varianzas iguales.

Antes de realizar la prueba es prudente hacer algo de estadística descriptiva:

```
summary(Biomarcador_hta)
      Min.   1st Qu. Median   Mean   3rd Qu.   Max.
      22.60   29.90   33.10   35.08   39.10   53.00

sd(Biomarcador_hta)
[1] 8.950493

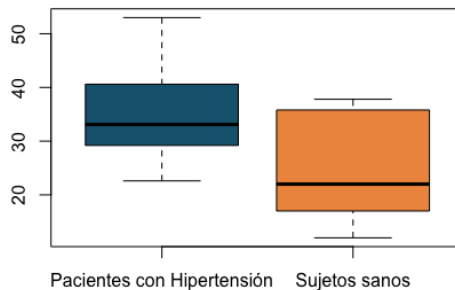
summary(Biomarcador_sanos)
      Min.   1st Qu. Median   Mean   3rd Qu.   Max.
      12.00   17.00   22.00   24.87   35.80   37.80

sd(Biomarcador_sanos)
[1] 9.698453
```

Y también algunas gráficas, aunque los boxplot no son las más adecuadas para evaluar medias, vamos a utilizarlas para obtener información sobre los datos:

```
boxplot(Biomarcador_hta, Biomarcador_sanos,
        main= "Concentraciones de un biomarcador entre pacientes con
y sin hipertensión", names=c("Pacientes con Hipertensión", "Sujetos sanos"),
col=c("#19647E", "#EE964B"))
```

Concentraciones de un biomarcador entre pacientes con y sin hipertensión



Ahora podemos realizar nuestra prueba t con el siguiente código:

```
t.test(x = Biomarcador_hta, y = Biomarcador_sanos,
       var.equal = T, conf.level = 0.98)
```

Two Sample t-test

```
data: Biomarcador_hta and Biomarcador_sanos
t = 2.658, df = 23, p-value = 0.01405
alternative hypothesis: true difference in means is not equal to 0
98 percent confidence interval:
 0.607241 19.809426
sample estimates:
mean of x mean of y
35.07500  24.86667
```

La interpretación de los resultados sería la siguiente:

- ♦ $t = 2.658$: el valor del estadístico t calculado.
- ♦ $df = 23$: los grados de libertad para la prueba. Se calculan como $n_1 + n_2 - 2$, donde n_1 y n_2 -son los tamaños de las dos muestras.
- ♦ $p\text{-value} = 0.01405$: indica la probabilidad de observar una diferencia tan grande (o más grande) entre las medias muestrales bajo la hipótesis nula.
- ♦ El intervalo de confianza al 98% para la diferencia de medias. Esto significa que estamos 98% seguros de que la verdadera diferencia de medias entre los dos grupos se encuentra entre 0.607241 y 19.809426.
- ♦ Estimación de medias:
 - » mean of $x = 35.0750$: la media de la muestra x (Biomarcador_hta).
 - » mean of $y = 24.8667$: la media de la muestra y (Biomarcador_sanos).

Hasta ahora hemos explicado cómo utilizar la función `t.test()` alimentándola con dos vectores. Ahora vamos a utilizar esta función mediante fórmula.

- ♦ **Ejemplo 16.2:** vamos a crear un *Dataframe* para poder utilizar la función `t.test()` mediante fórmula, copie y pegue el siguiente código:

```
# Combina los valores de los biomarcadores de ambos grupos en un solo vector.
Biomarcador <- c(Biomarcador_hta, Biomarcador_sanos)

# Crea un vector que indica el grupo al que pertenece cada valor de
Biomarcador_hta.
# La longitud del vector es igual al número de elementos en Biomarcador_hta.
Grupo_hta <- rep("Hipertensión", length(Biomarcador_hta))

# Crea un vector que indica el grupo al que pertenece cada valor de
Biomarcador_sanos.
# La longitud del vector es igual al número de elementos en Biomarcador_sanos.
Grupo_sanos <- rep("Sujetos sanos", length(Biomarcador_sanos))

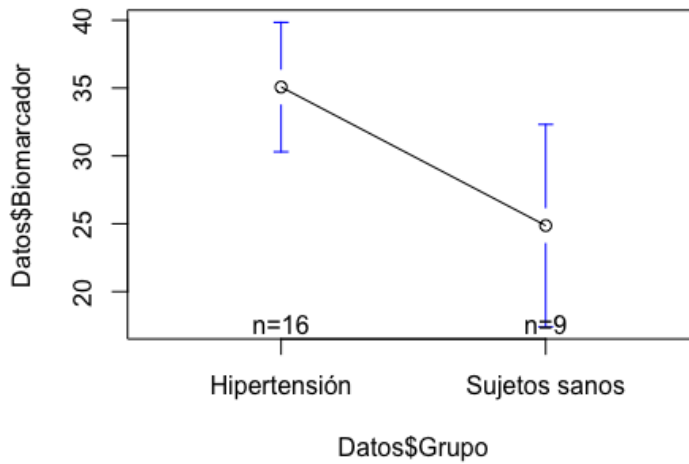
# Combina los dos vectores de grupo en uno solo.
# Este vector tendrá la misma longitud que el vector Biomarcador y
# indicará a qué grupo pertenece cada valor del biomarcador.
Grupo <- c(Grupo_hta, Grupo_sanos)

# Crea un data frame con dos columnas: Biomarcador y Grupo.
# La columna Biomarcador contiene los valores combinados de ambos grupos.
# La columna Grupo contiene las etiquetas de grupo correspondientes.
Datos <- data.frame(Biomarcador, Grupo)
```

Ya con nuestro *Dataframe* podemos realizar gráficas y la prueba .

Hay una gráfica que es más adecuada para cuando se van a comparar medias. La librería *gplots* ofrece una función para graficar las medias de dos grupos y sus intervalos de confianza al 95%.

```
library(gplots)
plotmeans(Datos$Biomarcador~Datos$Grupo, ci.label = F)
```



Esta gráfica ofrece información extra y adecuada para nuestros fines. El punto central es la media de cada grupo, mientras que las barras son intervalos de confianza para las medias. Tome en cuenta que estos intervalos no se estiman utilizando la distribución t , por lo que difieren a los de la prueba. Sin embargo, nos sirven para darnos una idea de las diferencias en los grupos:

```
t.test(Datos$Biomarcador~Datos$Grupo,
       var.equal = T, conf.level = 0.98)
```

Two Sample t-test

data: Datos\$Biomarcador by Datos\$Grupo

t = 2.658, df = 23, p-value = 0.01405

alternative hypothesis: true difference in means between group Hipertensión and group Sujetos sanos is not equal to 0

98 percent confidence interval:

0.607241 19.809426

sample estimates:

mean in group Hipertensión mean in group Sujetos sanos

35.07500

24.86667

Obtenemos el mismo resultado del [ejemplo 16.1](#). Recuerde que `Datos$Biomarcador~Datos$Grupo` puede leerse como las concentraciones del biomarcador agrupado por la variable grupo.

Test de Welch

El test de Welch, o prueba t de varianzas desiguales, es una prueba de ubicación de dos muestras que se utiliza para probar la hipótesis de que dos poblaciones tienen medias iguales. Lleva el nombre de su creador, Bernard Lewis Welch, es una adaptación de la prueba t de Student y es más confiable cuando las dos muestras tienen varianzas desiguales y tamaños de muestra desiguales.

El estadístico t se calcula mediante:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Donde:

- ♦ $\bar{x}_1$ es la media muestral del primer grupo.
- ♦ $\bar{x}_2$ es la media muestral del segundo grupo.
- ♦ s_1^2 y s_2^2 : varianzas muestrales de los dos grupos.
- ♦ n_1 y n_2 : tamaños de las dos muestras.

Sin embargo, los grados de libertad se estiman mediante:

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}$$

Esta fórmula de los grados de libertad permite corregir y obtener un valor de t adecuado para varianzas diferentes. Esta fórmula permite calcular un número efectivo de grados de libertad, que no es necesariamente un número entero, y que refleja la incertidumbre en las estimaciones de varianza.

Supongamos que tenemos los siguientes datos:

- ♦ Media del primer grupo $\bar{x}_1 = 35.075$

- ♦ Media del segundo grupo $\bar{x}_2 = 24.867$
- ♦ Varianza del primer grupo $s_1^2 = 50$
- ♦ Varianza del segundo grupo $s_2^2 = 30$
- ♦ Tamaño de la muestra del primer grupo $n_1 = 16$
- ♦ Tamaño de la muestra del segundo grupo $n_2 = 9$

Para estos datos el valor de t es:

$$t = \frac{35.075 - 24.867}{\sqrt{\frac{50}{16} + \frac{30}{9}}} = 4.02$$

Y los grados de libertad:

$$df = \frac{\left(\frac{50}{16} + \frac{30}{9}\right)^2}{\frac{\left(\frac{50}{16}\right)^2}{15} + \frac{\left(\frac{30}{9}\right)^2}{8}} = 20.44$$

Test de Welch en R

Para realizar el test de Welch en R cambie el argumento `var.equal=F`.

- ♦ **Ejemplo 16.3:** se sabe por antecedentes previos que el índice de masa corporal (bmi) de las mujeres de Pima se distribuye de una manera normal, su interés ahora es comparar el bmi entre las mujeres de Pima con y sin diabetes. Suponga que las varianzas son diferentes:

```
library(MASS) # Cargar Dataframe
data("Pima.tr2")
t.test(Pima.tr2$bmi~Pima.tr2$type, var.equal = F)
```

Welch Two Sample t-test

```
data: Pima.tr2$bmi by Pima.tr2$type
t = -5.4141, df = 243.68, p-value = 1.474e-07
alternative hypothesis: true difference in means between group No
```

```
and group Yes is not equal to 0
95 percent confidence interval:
-5.331380 -2.486953
sample estimates:
mean in group No mean in group Yes
30.67083          34.58000
```

Podemos concluir que el índice de masa corporal es diferente entre las mujeres con y sin diabetes. Note cómo no se especificó el nivel de α , si no se especifica utilice 0.05. Lo mismo sucede con la hipótesis, si no se especifica utilice hipótesis bilateral. Además, los intervalos de confianza al 95% son negativos, no incluyen al 0, lo que nos permite asegurar que las medias son diferentes para ambos grupos.

El test de Welch es mucho más estricto que la prueba t con varianzas iguales. Para ilustrarlo vamos a realizar el test de Welch con los datos del biomarcador para los pacientes con y sin hipertensión:

```
t.test(Datos$Biomarcador~Datos$Grupo,
       var.equal = F, conf.level = 0.98)
```

Welch Two Sample t-test

```
data: Datos$Biomarcador by Datos$Grupo
t = 2.5964, df = 15.593, p-value = 0.01978
alternative hypothesis: true difference in means between group Hi-
pertensión and group Sujetos sanos is not equal to 0
98 percent confidence interval:
 0.02176444 20.39490222
sample estimates:
mean in group Hipertensión mean in group Sujetos sanos
35.07500          24.86667
```

Note cómo el valor p de 2.658 para la prueba con varianzas iguales a 2.5964 para el test de Welch.

Homogeneidad de varianzas

Hasta ahora hemos realizado la prueba t para varianzas iguales y el test de Welch sin haber evaluado antes si las varianzas de los grupos son iguales (homogeneas), lo cual es un error.

Existen varias pruebas para evaluar la homogeneidad de varianzas, la [tabla 16.1](#) muestra una comparación rápida de las principales pruebas.

**TABLA 16.1. COMPARACIÓN DE LAS PRUEBAS DE HIPÓTESIS
PARA EVALUAR LA HOMOGENEIDAD DE VARIANZAS**

PRUEBA	TIPO	ROBUSTEZ A NO NORMALIDAD	FUNCIÓN EN R
Prueba de Levene	Paramétrica	Sí	leveneTest
Prueba de Bartlett	Paramétrica	No	bartlett.test
Prueba de Fligner-Killeen	No paramétrica	Sí	fligner.test

Prueba de Levene

La prueba de Levene es una de las pruebas más utilizadas para evaluar la igualdad de varianzas. Es robusta a las desviaciones de la normalidad.

- ♦ H_0 : las varianzas de los grupos son iguales.
- ♦ H_A : al menos una varianza es diferente.

En R puede usar la función `leveneTest` del paquete `car`. Se muestra un código de ejemplo:

```
# Instalar y cargar el paquete car si no está instalado
if(!require(car)) install.packages("car")
library(car)

# Ejemplo de datos
group <- factor(c(rep(1, 10), rep(2, 10), rep(3, 10)))
value <- c(rnorm(10, mean = 5, sd = 1), rnorm(10, mean = 5, sd = 1.5),
rnorm(10, mean = 5, sd = 2))
```

```
# Realizar la prueba de Levene
leveneTest(value ~ group)
Levene's Test for Homogeneity of Variance (center = median)
```

	Df	F	value	Pr(>F)
group	2	0.66	0.525	
	27			

Para este ejemplo, las varianzas no son iguales.

Test de Bartlett

La prueba de Bartlett se utiliza para evaluar la igualdad de varianzas cuando los datos se distribuyen normalmente.

- ♦ H_0 : las varianzas de los grupos son iguales.
- ♦ H_A : al menos una varianza es diferente.

En R puede usar la función `bartlett.test()`. Puede utilizar el siguiente código de ejemplo para realizar el test de Bartlett:

```
# Ejemplo de datos
group <- factor(c(rep(1, 10), rep(2, 10), rep(3, 10)))
value <- c(rnorm(10, mean = 5, sd = 1),
           rnorm(10, mean = 5, sd = 1.5), rnorm(10, mean = 5, sd = 2))

# Realizar la prueba de Bartlett
bartlett.test(value ~ group)
```

Bartlett test of homogeneity of variances

data: value by group

Bartlett's K-squared = 14.422, df = 2, p-value = 0.0007384

Para este ejemplo, las varianzas son iguales.

Prueba de Fligner-Killeen

La prueba de Fligner-Killeen es una prueba no paramétrica para la homogeneidad de varianzas y es robusta a las desviaciones de la normalidad.

- ♦ H_0 : las varianzas de los grupos son iguales.
- ♦ H_a : al menos una varianza es diferente.

Utilice el siguiente código de ejemplo para estimar la homogeneidad de varianzas con la prueba Fligner-Killeen:

```
# Ejemplo de datos
group <- factor(c(rep(1, 10), rep(2, 10), rep(3, 10)))
value <- c(rnorm(10, mean = 5, sd = 1),
           rnorm(10, mean = 5, sd = 1.5), rnorm(10, mean = 5, sd = 2))

# Realizar la prueba de Fligner-Killeen
fligner.test(value ~ group)
```

Fligner-Killeen test of homogeneity of variances

data: value by group

Fligner-Killeen:med chi-squared = 1.5339, df = 2, p-value = 0.4644

Para estos datos las varianzas son iguales.

La elección de la prueba adecuada depende de las características de sus datos, especialmente de la normalidad. La prueba de Levene y la prueba de Fligner-Killeen son generalmente recomendadas por su robustez a las desviaciones de la normalidad.

Comparación de una variable cuantitativa en grupos dependientes: pruebas paramétricas

Un método que se utiliza con frecuencia para averiguar la eficacia de un tratamiento, procedimiento experimental o medir los cambios a lo largo del tiempo, es aquel que hace uso de observaciones relacionadas que resultan de muestras no independientes, por ejemplo:

- ♦ Mediciones a través del tiempo
- ♦ Dos mediciones en el mismo sujeto
- ♦ Diseños cruzados

Estos son ejemplos de grupos pareados. Un grupo pareado o dependiente en estadística se refiere a un tipo de diseño de estudio en el que:

- ♦ Hay dos conjuntos de observaciones relacionadas.
- ♦ Cada observación en un grupo tiene una correspondencia específica con una observación en el otro grupo.

En estos casos la prueba de hipótesis más adecuada es la prueba t de Student para muestras pareadas. Dado que se trabaja con la diferencia de las medias, no es necesario realizar prueba de homogeneidad de varianzas. ¡Es la misma muestra!

La estimación del estadístico, se basa en el siguiente fórmula donde:

$$t = \frac{\bar{d} - \mu d_0}{s_{\bar{d}}}$$

Donde:

- ♦ $\bar{d}$ es la diferencia de la media muestral.
- ♦ μd_0 es la diferencia de la media poblacional supuesta.
- ♦ $s_{\bar{d}}$ es la desviación estándar de las diferencias dividida entre la raíz de n , es decir, $\frac{s_d}{\sqrt{n}}$

Vamos a resolver un ejemplo.

- ♦ **Ejemplo 16.4:** un grupo de investigadores de la licenciatura en Nutrición de la Universidad de Guadalajara implementaron una intervención dietética y de ejercicio en un grupo de 17 mujeres con sobrepeso con el objetivo de evaluar la eficacia de esta intervención a los 3 meses. Se pesaron a las pacientes antes y después de la intervención.

Con 95% de confianza y tomando en cuenta que los pesos de las mujeres que se incluyeron en el estudio se distribuyen normalmente, podrían concluir que hubo una reducción de peso.

Los pesos son los siguientes:

```
# Pesos iniciales y finales
pesos_iniciales <- c(80, 82, 77, 85, 78, 81, 79, 83,
                    80, 77, 84, 82, 79, 81, 76, 83, 78)
pesos_finales <- c(76, 79, 74, 82, 75, 78, 76, 80,
                  77, 74, 81, 79, 76, 78, 73, 80, 75)
```

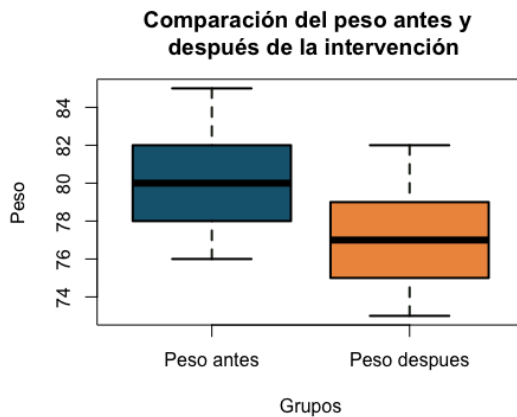
Tratamos de probar si:

- ♦ Si existe la suficiente evidencia para concluir que el programa de dietas es eficaz.
- ♦ Si es posible rechazar la hipótesis nula que indica que el cambio en la media de la población μ_d es cero o positivo.

Es necesario graficar. Las gráficas nos muestran mucha más información (nunca olvide que las gráficas son gratis). Para este ejemplo en particular, construir la gráfica utilizando ggplot2 es un poco complicado, ya que no tenemos un *Dataframe*, por lo que utilizaremos el paquete base. El siguiente código, además tiene como objetivo mostrar todos los argumentos de la función `boxplot` que podrá utilizar en el futuro. Usted puede copiar y pegar o hacer un gráfico más sencillo:

```
boxplot(pesos_iniciales, pesos_finales, # Datos
        horizontal = FALSE, # Gráfica horizontal o vertical
        lwd = 2, # Ancho de las líneas
        col = c("#19647E", "#EE964B"), # Color
```

```
xlab = "Grupos", # Título de las x
ylab = "Peso", # Título de las y
main = "Comparación del peso antes y \n después de la in-
tervención", # Título
border = "black", # Color del borde
outpch = 25, # Símbolo para los outliers
outbg = "green", # Outliers color
whiskcol = "#1C2319", # Bigotes color
whisklty = 2, # Bigotes tipo de linea
names=c("Peso antes", "Peso despues"), # Nombres de los grupos
lty = 1) #
```



Para realizar la prueba , puede emplear el siguiente código:

```
t.test(x=pesos_iniciales, y=pesos_finales, alternative = "greater",
paired = T, var.equal = T)
```

Paired t-test

data: pesos_iniciales and pesos_finales

t = 52, df = 16, p-value < 2.2e-16

```
alternative hypothesis: true mean difference is greater than 0
95 percent confidence interval:
 2.956124      Inf
sample estimates:
mean difference
 3.058824
```

Note cómo el grupo 1 es el peso inicial y el grupo 2 es el peso final, por lo tanto, la hipótesis que buscamos probar es:

$$H_o = \mu \text{ Peso inicial} \leq \mu \text{ Peso final}$$

$$H_A = \text{Peso inicial} < \mu \text{ Peso final}$$

Si la elección de los grupos fuera a la inversa, es decir, si el grupo 1 fuera el peso final y el grupo 2 fuera el peso inicial. Las hipótesis deberían plantearse al revés que la descrita anteriormente y el argumento alternative debería ser less. El argumento alternative va sobre la media del grupo 1.

En el caso que nuestra hipótesis sea a dos colas, en R debe de plantearse:

```
t.test(x=pesos_iniciales, y=pesos_finales,
       alternative = "two.sided",
       paired=T, var.equal = T)
```

Paired t-test

```
data: pesos_iniciales and pesos_finales
t = 52, df = 16, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 2.934123 3.183524
sample estimates:
mean difference
 3.058824
```

Note que el único argumento que se debe de cambiar para indicarle a R que la muestra es pareada es: `paired=T`. Este argumento, precisamente, nos permite diferenciar entre una prueba para muestras independientes.

Comparación de una variable cuantitativa en grupos independientes y dependientes: pruebas no paramétricas

Cuando se quiere comparar una variable de tipo cuantitativo u ordinal entre dos o entre más de dos grupos, pero esta no presenta una distribución normal no se puede emplear la prueba t-student, ya que esta prueba no es libre de parámetros y requiere que se cumplan algunos supuestos. Por lo tanto, debe de recurrirse a ciertas alternativas de pruebas no-paramétricas (libres de parámetros) para poder realizar esta comparación. Las pruebas a saber son:

1. Prueba U-Mann-Whitney. Prueba que compara la suma de rangos de variables en dos grupos independientes.
2. Prueba de los rangos de wilcoxon. Prueba que compara la suma de rangos de dos variables medidas en dos grupos dependientes.

La función `wilcox.test()` en R realiza tanto la prueba U de Mann-Whitney como la prueba de rangos con signo de Wilcoxon. El argumento `paired=TRUE` se utiliza para esta última prueba, de lo contrario no se debería de especificar el argumento.

Se describen a continuación cada una de estas pruebas.

Prueba U-Mann-Whitney

La prueba U de Mann-Whitney es una prueba estadística que se utiliza para determinar si 2 grupos son significativamente diferentes entre sí en la variable de interés (cuantitativa). La variable a comparar debe ser por lo menos de tipo ordinal. Además, los grupos deberán ser independientes (no relacionados entre sí) y se debe tener una muestra suficiente (más de 5 valores en cada grupo, aunque también depende de qué tan grande sea la diferencia entre los grupos).

La prueba U de Mann-Whitney es el equivalente a la prueba Student para muestras independientes. Necesita el cumplimiento de los siguientes supuestos: 1) las muestras son aleatorias e independientes, 2) la escala de medición es la menos ordinal, 3) la forma de los datos debe ser similar entre los grupos, 4) se deben de tener datos suficientes (más de 5).

La prueba U de Mann-Whitney se basa en la clasificación de todos los datos combinados de ambas muestras, asignándoles un rango. Luego se suman los rangos de cada muestra y se calcula un estadístico U que permite evaluar si las distribuciones de las dos muestras difieren significativamente. Un valor de U bajo indica que la primera muestra tiende a tener valores más altos que la segunda y viceversa. La [tabla 16.2](#) muestra un ejemplo de los rangos asignados a las horas de sueño de dos grupos de individuos (ver [ejemplo 16.5](#)).

TABLA 16.2. RANGOS PARA LAS HORAS DE SUEÑO DE DOS GRUPOS DE INDIVIDUOS

HORAS DE SUEÑO	POSICIÓN	RANGO	GRUPO
5	1	1	clonazepan
6	2	2	clonazepan
7	3	3	lorazepam
8	4	4.5	clonazepan
8	5	4.5	lorazepam
9	6	7	clonazepan
9	7	7	clonazepan
9	8	7	lorazepam
10	9	10	clonazepan
10	10	10	clonazepan
10	11	10	lorazepam
11	12	13	clonazepan
11	13	13	clonazepan
11	14	13	lorazepam
12	15	15.5	clonazepan
12	16	15.5	lorazepam
13	17	17.5	clonazepan

HORAS DE SUEÑO	POSICIÓN	RANGO	GRUPO
13	18	17.5	lorazepam
14	18	19	lorazepam
15	19	20	lorazepam
16	20	21	lorazepam
17	21	22	lorazepam

En la [tabla 16.2](#) los datos fueron ordenados de menor a mayor (horas de sueño) y se les asignó una posición y luego un rango. La posición se refiere al orden en el que encuentran los datos ordenados y el rango es muy similar a la posición, pero en el caso de valores de horas de sueño iguales deberá estimarse un promedio. La siguiente tabla muestra todos los datos correctos. Finalmente, con los rangos se estima el estadístico de prueba U para ambos grupos y selecciona el menor de ellos.

Para el grupo 1:

$$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

Para el grupo 2:

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

Finalmente se estima el valor mínimo:

$$U = \min(U_1, U_2)$$

En R el estadístico de prueba es que se estima:

$$U = W - \frac{n_1(n_1 + 1)}{2}$$

El fundamento de esta prueba consiste en que si las muestras provienen de la misma población sus rangos deberán distribuirse aleatoriamente, como se muestra en la [figura 16.2](#), mientras que si los datos provienen de una población distinta (hay algún factor que este interviniendo) los rangos de las muestras se van a separar (ver [figura 16.3](#)).

FIGURA 16.2. RANGOS DISTRIBUIDOS ALEATORIAMENTE

Muestras procedentes de la misma población

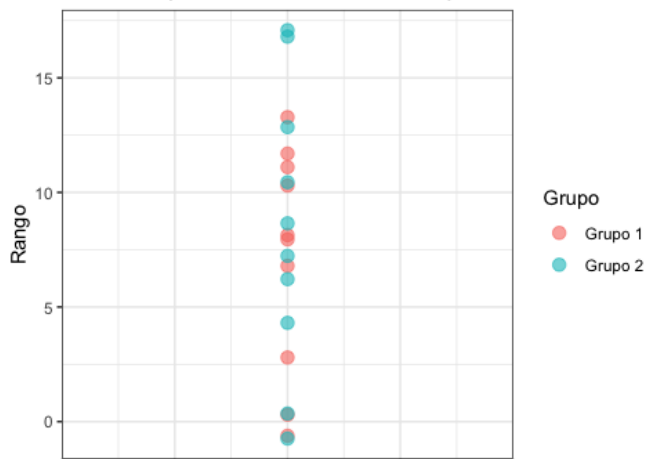
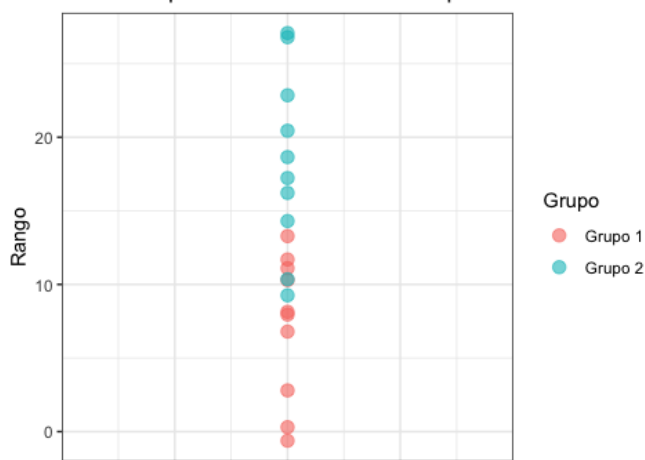


FIGURA 16.3. RANGOS DE DOS MUESTRAS SEPARADAS

Muestras procedentes de la misma población



- ♦ **Ejemplo 16.5:** un investigador desea conocer si las horas de sueño son distintas en dos grupos en grupo de pacientes tratados con benzodicepinas. El grupo 1 corresponde a pacientes medicados con clonazepam y el grupo 2 a pacientes con lorazepam. Las horas de sueño se muestran enseguida.

CLONZEPAM	LORAZEPAM
10	13
6	17
8	14
10	12
12	10
13	9
11	15
9	16
5	11
11	8
9	7

Se sabe que los datos no se ajustan a una distribución normal, por lo que no puede emplearse la prueba *t* de Student. Por tanto, la prueba de elección es la prueba de U-Mann-Whintey.

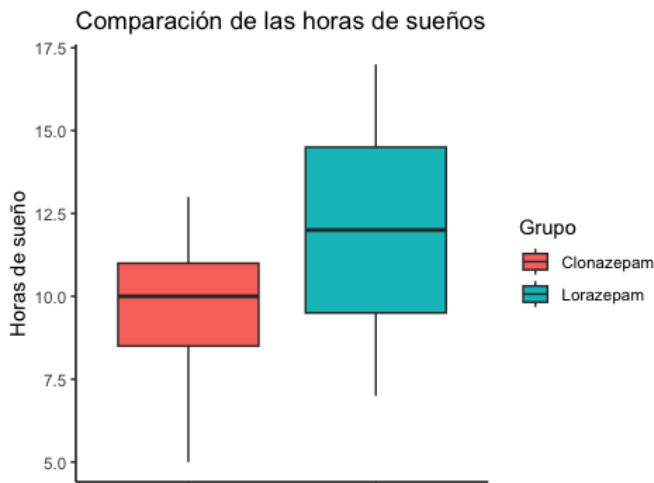
Para construir un *Dataframe* con las horas de sueño utilice el siguiente código:

```
clonazepam ← c(10, 6, 8, 10, 12, 13, 11, 9, 5, 11, 9)
lorazepam ← c(13, 17, 14, 12, 10, 9, 15, 16, 11, 8, 7)

data ← data.frame(
  Horas = c(clonazepam, lorazepam),
  Grupo = rep(c("Clonazepam", "Lorazepam"), each = 11)
)
```

Podemos hacer algunas gráficas:

```
library(ggplot2)
data|>
  ggplot(mapping = aes(x=Grupo, y=Horas, fill = Grupo))+
  geom_boxplot()+
  theme_classic()+
  labs(title = "Comparación de las horas de sueños", x="Grupos",
y="Horas de sueño")
```



Ahora sí ya podemos realizar nuestra prueba estadística:

```
wilcox.test(data$Horas~data$Grupo)
Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...):
cannot
compute exact p-value with ties
```

Wilcoxon rank sum test with continuity correction

data: data\$Horas by data\$Grupo

W = 34.5, p-value = 0.09265

alternative hypothesis: true location shift is not equal to 0

Donde:

- ❖ Valor del estadístico de 34.5 es el resultado de la suma de los rangos de una de las dos muestras.
- ❖ El valor p de 0.09265 indica la probabilidad de observar un resultado tan extremo como el obtenido, bajo la hipótesis nula de que no hay diferencia entre las dos muestras. Dado que este valor es mayor que el nivel de significancia que se utiliza por constumbre (0.05), no se puede rechazar la hipótesis nula. Esto sugiere que no hay suficiente evidencia para afirmar que hay una diferencia significativa en las horas de sueño entre los dos grupos de pacientes que tomaron diferentes benzodiacepinas.

Note cómo además obtuvimos un warning. Esta advertencia indica que hay valores empatados (ties) en los datos, lo cual impide calcular un valor p exacto. En su lugar, el test utiliza una aproximación para calcular el valor p . Pacientes de los dos grupos tienen exactamente las mismas horas de sueño.

Rangos de Wilcoxon

Es el equivalente no paramétrico de la prueba t-student para muestras pareadas. La prueba de los rangos de Wilcoxon, también conocida como el test de Wilcoxon, es una prueba no paramétrica utilizada para comparar dos muestras relacionadas o pareadas. En la prueba de los rangos con signo de Wilcoxon, se asignan rangos a las diferencias entre las observaciones pareadas y se calcula un estadístico que permite determinar si los dos grupos de datos son diferentes. Su realización es muy similar a la U de Mann-Whitney, sin embargo, a diferencia de la primera, lo que se ordena y a lo que se le suman los rangos son las diferencias de cada par de datos.

Para realizar esta prueba se deben de cumplir con los siguientes supuestos:

1. Datos obtenidos de forma aleatoria.
2. Las observaciones proceden de una población con distribución simétrica. Es decir, las muestras provienen de poblaciones con la misma forma de distribución.
3. Variable al menos ordinal.

- ♦ **Ejemplo 16.6:** manualmente la prueba de Rangos de Wilcoxon se realizaría de la siguiente forma: unos investigadores desean conocer si el tratamiento experimental influye en el peso de dos grupos de ratones. La siguiente tabla muestra los pesos de los ratones antes y después del tratamiento experimental

PESO ANTES	PESO DESPUÉS
90	80
95	89
80	72
79	84
89	94
95	80
99	89
84	72
78	84
69	64
74	80
95	89
99	72
84	84
78	94
69	84
74	94

Los pasos para resolver manualmente este ejemplo son:

1. Calcular la diferencia entre cada par eliminando los 0:

PESO ANTES	PESO DESPUÉS	DIFERENCIAS
90	80	10
95	89	6
80	72	8
79	84	-5

PESO ANTES	PESO DESPUÉS	DIFERENCIAS
89	94	-5
95	80	15
99	89	10
84	72	12
78	84	-6
69	64	5
74	80	-6
95	89	6
99	72	27
84	84	0
78	94	-16
69	84	-15
74	94	-20

2. Estimar el valor absoluto para cada diferencia:

PESO ANTES	PESO DESPUÉS	DIFERENCIAS	DIFERENCIAS VALOR ABSOLUTO
90	80	10	10
95	89	6	6
80	72	8	8
79	84	-5	5
89	94	-5	5
95	80	15	15
99	89	10	10
84	72	12	12
78	84	-6	6
69	64	5	5
74	80	-6	6
95	89	6	6
99	72	27	27

PESO ANTES	PESO DESPUÉS	DIFERENCIAS	DIFERENCIAS VALOR ABSOLUTO
84	84	0	0
78	94	-16	16
69	84	-15	15
74	94	-20	20

3. Sin tener en cuenta el signo de la diferencia, los pares se ordenan de menor a mayor considerando la magnitud absoluta. A la par con la menor diferencia se le asigna la primera posición, al de mayor diferencia se le asigna la última:

PESO ANTES	PESO DESPUÉS	DIFERENCIAS	DIFERENCIAS VALOR ABSOLUTO	RANGO
84	84	0	0	
79	84	-5	5	1
89	94	-5	5	2
69	64	5	5	3
95	89	6	6	4
78	84	-6	6	5
74	80	-6	6	6
95	89	6	6	7
80	72	8	8	8
90	80	10	10	9
99	89	10	10	10
84	72	12	12	11
95	80	15	15	12
69	84	-15	15	13
78	94	-16	16	14
74	94	-20	20	15
99	72	27	27	16

A las variables con el mismo valor, se estima el promedio:

PESO ANTES	PESO DESPUÉS	DIFERENCIAS	DIFERENCIAS VALOR ABSOLUTO	RANGO	RANGO
84	84	0	0		
79	84	-5	5	1	2
89	94	-5	5	2	2
69	64	5	5	3	2
95	89	6	6	4	5.5
78	84	-6	6	5	5.5
74	80	-6	6	6	5.5
95	89	6	6	7	5.5
80	72	8	8	8	8
90	80	10	10	9	9.5
99	89	10	10	10	9.5
84	72	12	12	11	11
95	80	15	15	12	12.5
69	84	-15	15	13	12.5
78	94	-16	16	14	14
74	94	-20	20	15	15
99	72	27	27	16	16

4. Se suman los valores de los rangos para las diferencias positivas y para las diferencias negativas:

$T_{\text{positivo}} \leftarrow 2+7+5.5+8+9.5+9.5+11+12.5+16$

$T_{\text{negativo}} \leftarrow 15+14+12.5+5.5+5.5+2+2$

- Valor de T para valores positivos: 81
- Valor de T para valores negativos: 56.5

5. Se compara el valor de más grande con el valor de tablas y se obtiene el valor de p .

♦ **Ejemplo 16.7:** en R la prueba de rangos de wilcoxon se realiza de la siguiente forma:

1. Se crean dos objetos, uno para peso antes y otro para peso después:

```
Peso_Antes ← c(74, 78, 69, 78, 74, 79, 89, 84, 69, 95, 95, 80, 90,
99,84,95, 99)
Peso_Despues ← c(94, 94, 84, 84, 80, 84, 94, 84, 64, 89, 89, 72, 80,
89, 72, 80, 72)
```

2. Se utiliza la función `wilcox.test` con el argumento `paired=T`:

```
wilcox.test(Peso_Antes, Peso_Despues, paired = T, conf.int = T)
```

Wilcoxon signed rank test with continuity correction

data: Peso_Antes and Peso_Despues

V = 79.5, p-value = 0.5685

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-5.499992 9.000023

sample estimates:

(pseudo)median

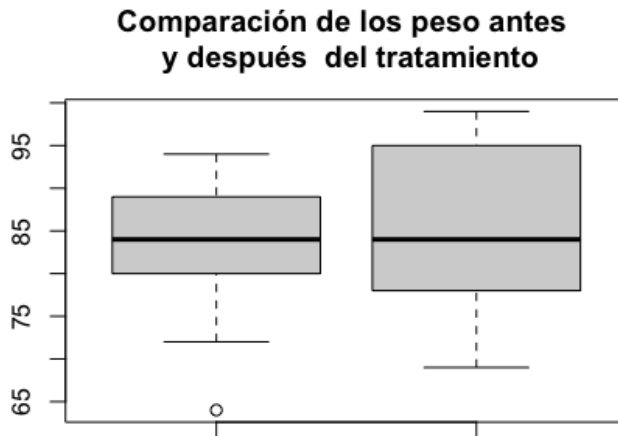
1.849605

La interpretación de los resultados es la siguiente:

- Valor del estadístico V: el estadístico V de 79.5 es el resultado de la suma de los rangos positivos o negativos de las diferencias entre las dos muestras.
- El valor p de 0.5685 indica la probabilidad de observar un resultado tan extremo como el obtenido, bajo la hipótesis nula de que no hay diferencia entre las dos muestras. Dado que este valor p es mayor que el nivel de α de 0.05, no se puede rechazar la hipótesis nula. Esto sugiere que no hay suficiente evidencia para afirmar que hay una diferencia significativa en los pesos antes y después del tratamiento.

- El intervalo de confianza del 95% (-5.499992, 9.000023) incluye el valor 0, lo que respalda aún más la conclusión de que no hay una diferencia significativa.
 - Estimación de la mediana: la mediana de las diferencias es aproximadamente 1.849605, pero este valor no es significativamente diferente de 0 dado el valor p alto y el intervalo de confianza.
3. Hacer un boxplot para visualizar sus resultados:

```
boxplot(Peso_Despues, Peso_Antes, main="Comparación de los peso antes \n y después del tratamiento")
```



Tamaño del efecto

Hasta ahora, hemos utilizado pruebas de hipótesis e intervalos de confianza para ver diferencias en las medias o en los rangos de dos grupos de datos. Sin embargo, esto es sumamente simplista. Solo podemos establecer si hay o no diferencias sin saber cuán tan grande o importante es esta diferencia. En estadística existe una medida, el tamaño del efecto, que ayuda a interpretar la importancia práctica de

los resultados, no solo su significancia estadística. En otras palabras, el tamaño del efecto nos permite conocer la magnitud de una diferencia o de una asociación. Entre más grande sea el tamaño del efecto más importancia clínica podría tener.

Debe tomar en cuenta al tamaño del efecto junto con la significancia estadística al interpretar los resultados de un estudio. Es común que los resultados sean estadísticamente significativos, pero se observe un tamaño del efecto apenas relevante. Un tamaño de efecto pequeño puede indicar que la relación o diferencia encontrada no tiene relevancia práctica.

Aunque existen una gran cantidad de medidas para el tamaño del efecto en este libro nos enfocaremos en los siguientes:

- ♦ **Cohen's d**: usado para comparar las diferencias entre las medias de dos grupos
- ♦ **Hedges' g**: corrección para Cohen's d
- ♦ **r (correlación de Pearson)**: a menudo transformado para evaluar tamaños de efecto en contextos de correlación
- ♦ ω^2 : comúnmente utilizado en ANOVA

De momento explicaremos los primero dos, cuando llegue su tiempo explicaremos ω^2 y r .

Cohen's d

Una de las medidas del efecto más utilizadas es la d de Cohen o Cohen's d. Esta medida se basa en la diferencia estandarizada entre dos medias mediante la siguiente fórmula:

$$d = \sqrt{\frac{\bar{x}_1 - \bar{x}_2}{s_p}}$$

Donde: $\bar{x}_1$ y $\bar{x}_2$ son las medias de los dos grupos, y s_p es la desviación estándar combinada, calculada como:

$$S_p = \sqrt{\frac{(n_1 - 1) s_1^2 + (n_2 - 1) s_2^2}{n_1 + n_2 - 2}}$$

Se utiliza comúnmente para evaluar el tamaño del efecto en una prueba , aunque también puede utilizarse en otros escenarios que involucren la comparación de grupos. La interpretación de la Cohen's d es la siguiente:

- ♦ Efecto pequeño (0.2)
- ♦ Efecto mediano (0.5)
- ♦ Efecto grande (0.8)

Hedges'g

Una medida del tamaño del efecto que utilizaremos regularmente aquí es g de Hedges o Hedges'g. Esta medida es muy similar a la de Cohen, sin embargo, es particularmente útil para grupos con tamaño de muestra pequeños. De hecho, la g de Hedges corrige el sesgo que puede ocurrir en la d de Cohen cuando trabaja con muestras pequeñas. Se calcula mediante:

$$g = d \left(1 - \frac{3}{4(N - 2) - 1} \right)$$

Donde: d es el tamaño del efecto de Cohen y N es el tamaño total de la muestra ($N = n_1 + n_2$).

La interpretación para la g de Hedges es la siguiente:

- ♦ **Menor que 0.2:** un efecto pequeño
- ♦ **Entre 0.2 y 0.5:** un efecto moderado
- ♦ **Entre 0.5 y 0.8:** un efecto grande

Rank biserial

Una medida que se utiliza para pruebas no paramétricas es el Rank biserial, el cual puede aplicarse para evaluar el tamaño del efecto en situaciones que involucran la prueba U de Mann-Whitney. La interpretación de la correlación biserial de rango se basa en la magnitud del coeficiente de correlación. Normalmente, una correlación biserial de rango cercana a 0 indica una asociación débil, mientras que valores más cercanos a 1 o -1 sugieren una relación lineal más fuerte entre las variables.

ggstatsplot

Una de las grandes ventajas que tenemos en R es la construcción de gráficas. Son gratis. En este sentido existe una librería llamada ggstatsplot que permite la construcción de gráficas con pruebas estadísticas, intervalos de confianza y tamaño del efecto. ggstatsplot es una extensión del paquete ggplot2 para crear gráficos con detalles de pruebas estadísticas incluidos en los propios gráficos ricos en información. Suele generar muchos problemas para su instalación, le recomendamos altamente revisar a detalle [la página oficial](#) de ggstatsplot para encontrar ayuda y una explicación más a detalle de esta librería.

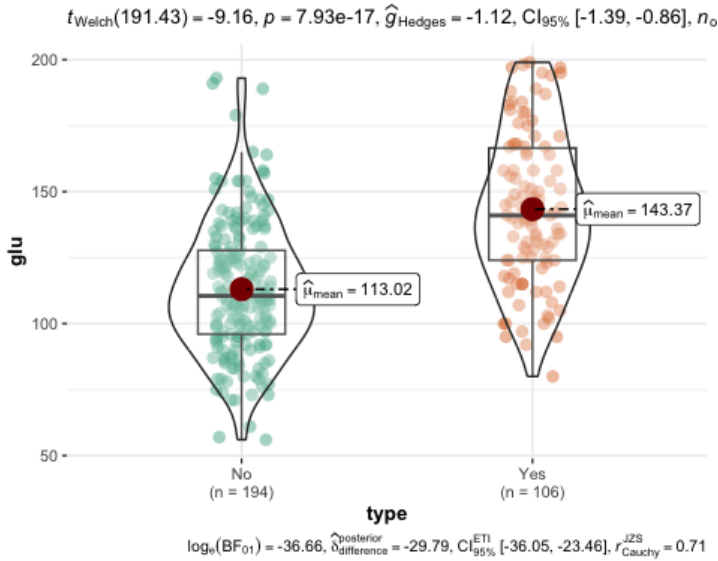
En el mejor de los casos, el siguiente código le permitirá instalar ggstatsplot:

```
install.packages("ggstatsplot")
```

Las funciones que utilizaremos en este libro se describen brevemente a continuación:

- ♦ `ggbetweenstats`: para comparaciones de medios entre grupos (por ejemplo, boxplots con resultados de t-test).
- ♦ `ggwithinstats`: para comparaciones dentro de grupos (por ejemplo, paired samples t-tests).
- ♦ `ggscatterstats`: para análisis de correlación y regresión lineal.
- ♦ `ggpiestats`: para análisis de datos categóricos (por ejemplo, tabla de contingencia con chi-squared test).
- ♦ **Ejemplo 16.8:** vamos a retomar los datos del [ejemplo 16.3](#) para utilizar ggstatsplot y mostrar sus bondades:

```
ggstatsplot::ggbetweenstats(
  data=Pima.tr2, # Es necesario indicarle donde tomar los datos
  x=type, # Variable de agrupación
  y=glu, # Variable numérica
  type = "parametric" # Prueba paramétrica
)
```



Note cómo la función `ggbetweenstats` realizó la comparación mediante un test de Welch sin la necesidad de indicarle que necesitábamos esa comparación. Si colocamos la variable correcta en `x` y en `y` no tendremos problemas. En esta función, en `x` se coloca un factor o variable de agrupación y en `y` la variable numérica.

La información que muestra la gráfica es bastante, la vamos desglosar:

1. Distribución de los datos:

- » La gráfica muestra dos gráficos de violín que representan la distribución de los niveles de glucosa (glu) para las mujeres de Pima con y sin diabetes.
- » Los puntos dentro de los violines indican los valores individuales de glucosa para cada observación en los dos grupos.

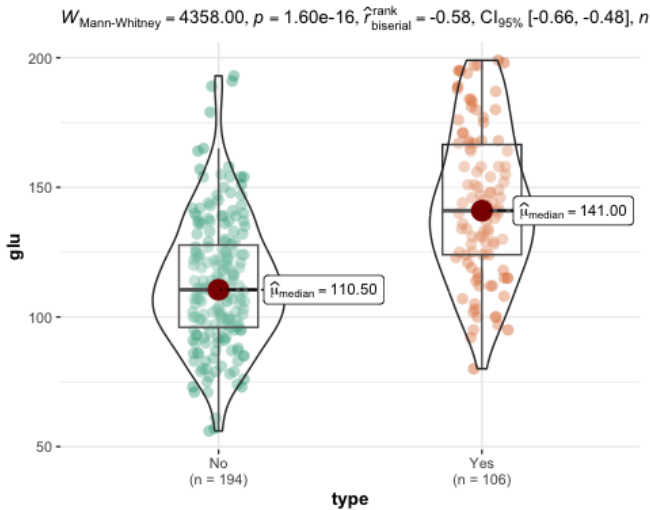
2. Estadísticas descriptivas:

- » En cada violín, hay un punto rojo grande que indica la media de los niveles de glucosa para cada grupo.
- » Para el grupo "No" ($n = 194$), la media (μ) es 113.02.
- » Para el grupo "Yes" ($n = 106$), la media (μ) es 143.37.
- » Boxplot: la caja representa el IQR, que contiene el 50% central de los datos. Las líneas que se extienden desde la caja (bigotes) indican la extensión de los datos dentro de 1.5 veces el IQR desde los cuartiles.

3. Prueba t de Welch:

- » Se muestra el valor de t de -9.16 para 191.43 grados de libertad.
 - » Valor de p de 7.93×10^{-17} indica una diferencia estadísticamente significativa entre los dos grupos.
4. Tamaño del efecto:
 - » El valor para $\hat{g}$ de Hedges es 1.12 que sigue un efecto grande.
 5. Intervalos de confianza:
 - » El intervalo de confianza del 95% para la diferencia de medias es , por lo que podemos decir que la diferencia de medias es diferente de 0.
- ♦ **Ejemplo 16.9:** si es de nuestro interés realizar estadística no paramétrica, lo único que debemos de modificar en el código es el argumento `type`:

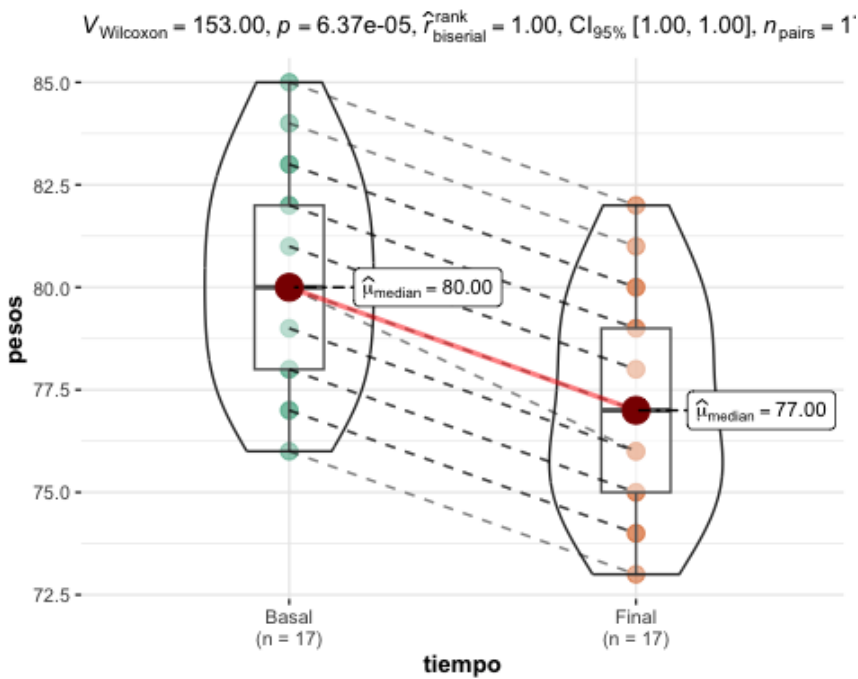
```
ggstatsplot::ggbetweenstats(  
  data=Pima.tr2, # Es necesario indicarle donde tomar los datos  
  x=type, # Variable de agrupación  
  y=glu, # Variable numérica  
  type = "np" # Prueba paramétrica  
)
```



Para interpretar la gráfica puede basarse en el ejemplo anterior.

- ♦ **Ejemplo 16.10:** vamos a utilizar los datos del [ejemplo 16.4](#) para utilizar la función `ggwithinstats` de la librería `ggstatsplot`:

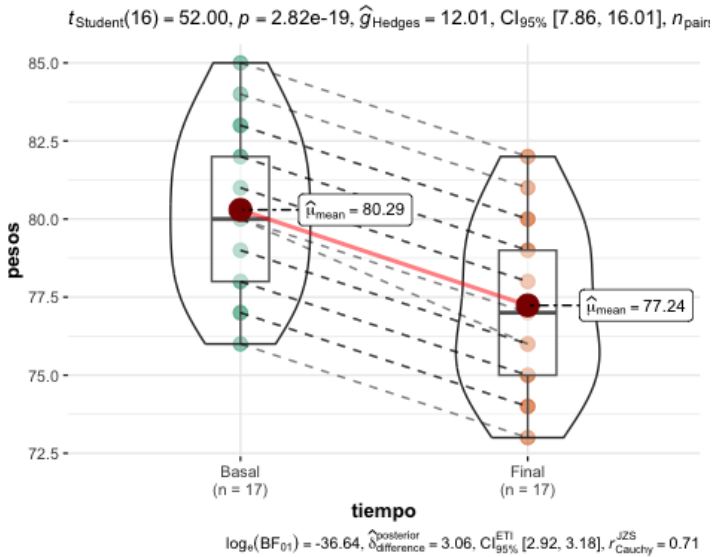
```
#Construcción de la base de datos
pesos <- c(pesos_iniciales, pesos_finales)
tiempo <- rep(c("Basal", "Final"), each=17)
datos <- data.frame(tiempo,pesos)
# Rangos de wilcoxon
datos|>
  ggstatsplot::ggwithinstats(x = tiempo,
                             y= pesos,
                             type = "np")
```



Y para una prueba *t* pareada:

```
datos|>
```

```
ggstatsplot::ggwithinstats(x = tiempo,
                             y = pesos,
                             type = "parametric")
```



Ejercicios

- ♦ **Ejercicio 16.1:** utilizando la base de datos “Pima.tr” describa si hay diferencias en los niveles de glucosa, la presión arterial, en *sikin*, índice de masa corporal, en *ped* y edad entre las mujeres con diabetes y las mujeres sin diabetes. Para este problema utilice únicamente la función `t.test` de R, no utilice otra prueba estadística. Para cada una de las variables entregue los siguiente:
 - a) Realice un boxplot para cada variable agrupando entre las mujeres con diabetes y sin diabetes.

- b) Realice un gráfico de violín para cada variable agrupando entre mujeres con diabetes y sin diabetes.
- c) Realice un gráfico de `plotmeans` para cada variable agrupando entre mujeres con diabetes y sin diabetes.
- d) Resultado de la prueba t – *student*.
- e) Conclusión.

Resolución ejercicio 16.1

Se muestran solo resultados para glucosa:

```
# Importar base
library(MASS)
data("Pima.tr")
# Gráficos
## Boxplot
boxplot(Pima.tr2$glu~Pima.tr2$type, # Datos
        lwd = 2, # Ancho de lineas
        col = c("#76EEC6", "#FFE4C4"), # Color
        xlab = "Grupos", # Etiqueta del eje de las x
        ylab = "Concentraciones de glucosa", # Eje de las Y
        main = "Comparación de la glucosa entre \n mujeres de Pima
sin y con diabetes", # Título
        border = "black", # Color del borde de la caja
        outpch = 25, # Símbolo outlier
        outbg = "green", # Color outliers
        whiskcol = "blue", # Color del bigote
        whisklty = 2, # Estilo lista del bigote
        names=c("Mujeres sin diabetes",
                "Muejres con diabetes"), #Nombres de los grupos
        lty = 1)
# Gráficos
## Violín
vioplot::vioplot(Pima.tr2$glu~Pima.tr2$type, # Datos
                 col = c("#76EEC6", "#FFE4C4"), # Color
```

```

xlab = "Grupos", # Etiqueta del eje de las x
ylab = "Concentraciones de glucosa", # Etiqueta ejes Y
main = "Comparación de la glucosa entre \n mujeres de Pima
sin y con diabetes", # Título
names=c("Mujeres sin diabetes", "Muejres con diabetes"))
# Gráficos
## Plot means
gplots::plotmeans(Pima.tr2$glu~Pima.tr2$type, # Datos
  ylab = "Concentraciones de glucosa", # Etiqueta ejes Y
  xlab = "Grupos de estudio",
  main = "Comparación de la glucosa entre \n mujeres de Pima
sin y con diabetes", # Título
)
# Evaluación de la homgenidad de varianzas
car::leveneTest(Pima.tr2$glu~Pima.tr2$type)

t.test(Pima.tr2$glu~Pima.tr2$type, var.equal=F)

```

- ♦ **Ejercicio 16.2:** utilizando la base de datos “Pima.tr” describa si hay diferencias en los niveles de glucosa y la presión arterial entre las mujeres con más de 40 años y las mujeres con 40 años o menos.

Resolución ejercicio 16.2

Se muestra la resolución solo para glucosa:

```

Pima.tr2$Edad_40 ← cut (Pima.tr2$age, breaks = c(0,40,Inf),
  labels = c("Menos de 40", "Más de 40"))
# Evaluación de la normalidad
nortest::lillie.test(Pima.tr2$glu)
library(ggrain)
library(ggplot2)
Pima.tr2|>
  ggplot(aes(x=1, y = glu))+
  geom_rain()

```

La edad no sigue una distribución normal, por lo que se debe de utilizar U de Mann Whitney:

```
wilcox.test(Pima.tr2$glu~Pima.tr2$Edad_40)
```

♦ **Ejercicio 16.3:** la base de datos [S1Dataset.xlsx](#) es una base de datos pública en la que se intentó asociar las concentraciones de adipocinas con la nefritis lúpica y los valores de proteinuria en pacientes con lupus eritematoso sistémico. Esta base es un libro de Excel con varias hojas:

1. DATASET PONE-D-17-14360R1: contiene todos los datos y mediciones de los pacientes.
2. VARIABLES INFORMATION: contiene una descripción breve de las variables de la hoja anterior. Puede encontrar más información en: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0184056>

Utilizando la base de datos [S1Dataset.xlsx](#), que puede descargar [en el siguiente link](#), replique los resultados de la tabla 2 del artículo: [Serum levels of adiponectin and leptin as biomarkers of proteinuria in lupus nephritis](#). La tabla deberá de incluir estadística descriptiva y pruebas de hipótesis para variables cuantitativas. Asuma estadística paramétrica.

Resolución ejercicio 16.3

Si baja su base de datos a su carpeta de “Bases”, el siguiente código debería de funcionar:

```
## Importar base
df <- readxl::read_excel("Bases/S1Dataset.xlsx",
                        sheet = "DATABASE PONE-D-17-14360R1")

## Filtrar pacientes
df <- df[df$Groups_SLEvsCL==1,]

## Cambiar factores
df$Groups_NLSLEvsSLE <- factor(df$Groups_NLSLEvsSLE,
                              labels = c("No LN", "LN"))
```

Se muestran los resultados solo para la variable Age:

```
tapply(X = df$Age, INDEX = df$Groups_NLSLEvsSLE, mean)
tapply(X = df$Age, INDEX = df$Groups_NLSLEvsSLE, sd)
car::leveneTest(df$Age~df$Groups_NLSLEvsSLE)
t.test(df$Age~df$Groups_NLSLEvsSLE, var.equal=T)
```

- ♦ **Ejercicio 16.4:** repita la tabla del [ejercicio 16.3](#) utilizando estadística no paramétrica, cambie además la media por la mediana y la desviación estándar por el IQR.

Resolución ejercicio 16.4

Se muestran los resultados solo para la variable Age:

```
tapply(X = df$Age, INDEX = df$Groups_NLSLEvsSLE, median)
tapply(X = df$Age, INDEX = df$Groups_NLSLEvsSLE, IQR)
wilcox.test(df$Age~df$Groups_NLSLEvsSLE)
```

- ♦ **Ejercicio 16.5:** explique e investigue por qué cambian los valores de p entre las pruebas paramétricas y no paramétricas. ¿Qué es lo que sucede con el valor de p ?
- ♦ **Ejercicio 16.6:** la “Base_Prueba_t_pareada” es una base de datos que contiene la mediciones de adipocinas en pacientes con insuficiencia hepática. Contiene mediciones de estas adipocinas séricas a distintos tiempos: Basales, 3 meses, 6 y 12 meses. Puede descargar la base de datos [aquí](#).

Utilizando la base de datos “Base_Prueba_t_pareada” resuelva:

¿Existe evidencia para demostrar que los pacientes con insuficiencia hepática presentaron algún cambio entre las mediciones basales de leptina (Leptin) y las mediciones de leptina a los 12 meses (Leptin12M)? Asuma distribución normal.

- Realice los gráficos adecuados.
- Realice la prueba de hipótesis pertinente para responder la pregunta.

Resolución ejercicio 16.6

```
df <- read.csv("Bases/Base_Prueba_t_pareada.csv")

boxplot(df$Leptin, df$Leptin12M, col=c("#19647E", "#28AFB0"),
        names = c("Leptina basasl", "Leptina 12 meses"))

t.test(df$Leptin, df$Leptin12M, paired = T)
```

- ♦ **Ejercicio 16.7:** los investigadores asignaron una intervención para reducir las concentraciones de adiponectina en los pacientes con insuficiencia hepática, para ello realizaron mediciones basales de esta adipocina (Adiponectin) y mediciones a los 6 meses (Adiponectin6M) y a los 12 meses (Adiponectin12M). ¿Pueden concluir los investigadores que se presentó una disminución de las concentraciones séricas de adiponectina?
- Realice los gráficos adecuados.
 - Realice la prueba de hipótesis pertinente para responder la pregunta.

Resolución ejercicio 16.7

Dado que no se sabe nada sobre la normalidad de los datos, es necesario hacer una evaluación de la normalidad:

```
nortest::lillie.test(df$Adiponectin)
df|>
  ggplot(aes(x=1, y = Adiponectin))+
  geom_rain()

nortest::lillie.test(df$Adiponectin12M)
df|>
  ggplot(aes(x=1, y = Adiponectin12M))+
  geom_rain()
```

No se sigue una distribución normal:

```
wilcox.test(df$Adiponectin, df$Adiponectin12M, paired = T)
```


- ♦ **Ejercicio 16.8:** los investigadores a cargo del estudio realizaron la medición de un biomarcador que se asocia a una enfermedad más grave en los pacientes con insuficiencia hepática; se ha validado que a mayores concentraciones de este biomarcador se presenta una enfermedad más grave. ¿Existe evidencia para demostrar que los pacientes empeoraron a los 6 meses y a los 12 meses con respecto a la medición basal? La variable “Biomarcador” tiene las mediciones basales del biomarcador de interés, mientras que variables “Biomarcador6M” y “Biomarcador12M” las mediciones a los 6 y 12 meses.
 - a) Realice los gráficos adecuados.
 - b) Realice la prueba de hipótesis pertinente para responder la pregunta.

Resolución ejercicio 16.8

Se resuelve igual que el [ejercicio 16.7](#)

- ♦ **Ejercicio 16.9:** repita los ejercicios [ejercicio 16.1](#) y [ejercicio 16.2](#) utilizando ggstatsplot.

Resolución ejercicio 16.9

Se muestra solo la resolución para glucosa:

```
Pima.tr2|>
  ggstatsplot::ggbetweenstats(x = type,
                              y = glu,
                              type = "parametric")
```

Referencias

- Adebayo, O. P. (2024). Determine robust procedure for testing variance equality using type I error rate and power. <https://doi.org/10.21203/rs.3.rs-4358203/v1>
- Allingham, D. (2012). Testing equality of variances for multiple univariate normal populations. *Journal of Statistical Theory and Practice*, 6(3), 524-535. <https://doi.org/10.1080/15598608.2012.695703>

- Biau, D., Kernéis, S. y Porcher, R. (2008). Statistics in Brief: The Importance of Sample Size in the Planning and Interpretation of Medical Research. Lippincott Williams & Wilkins, 466(9), 2282-2288. <https://doi.org/10.1007/s11999-008-0346-9>
- Bravo, F. (2006). Bartlett-type adjustments for empirical discrepancy test statistics. *Journal of Statistical Planning and Inference*, 136(3), 537-554. <https://doi.org/10.1016/j.jspi.2004.08.010>
- Celis de la Rosa, A. J. (2014). Pruebas de hipótesis para una y dos variables cuantitativas. En A. de J. Celis de la Rosa y V. Labrada Martagón, *Bioestadística* (3ª ed., pp. 135-143). Editorial El Manual Moderno.
- Charway, H. (2007). Testing multiple-group variance equality with randomization procedures. *Journal of Statistical Computation and Simulation*, 77(9), 797-803. <https://doi.org/10.1080/10629360600681025>
- Corty, R. W. (2018). Mean-variance QTL mapping on a background of variance heterogeneity. <https://doi.org/10.1101/276980>
- Dowdy, A., Tincani, M. y Schneider, W. J. (2020). Evaluation of publication bias in response interruption and redirection: A meta-analysis. *Journal of Applied Behavior Analysis*, 53(4), 2151-2171. <https://doi.org/10.1002/jaba.724>
- Dreison, K. C., Luther, L., Bonfils, K. A., Sliter, M. T., McGrew, J. H. y Salyers, M. P. (2018). Job burnout in mental health providers: A meta-analysis of 35 years of intervention research. *Journal of Occupational Health Psychology*, 23(1), 18-30. <https://doi.org/10.1037/ocp0000047>
- Fay, M. P. y Malinovsky, Y. (2018). Confidence intervals of the Mann-Whitney parameter that are compatible with the Wilcoxon-Mann-Whitney test. *Statistics in Medicine*, 37(27), 3991-4006. <https://doi.org/10.1002/sim.7890>
- Festing, M. F. y Altman, D. G. (2002). Guidelines for the design and statistical analysis of experiments using laboratory animals. *ILAR Journal*, 43(4), 244-258. <https://doi.org/10.1093/ilar.43.4.244>
- Fields, S., Lange, K., Ramos, A., Thamotharan, S. y Rassu, F. (2014). The relationship between stress and delay discounting. *Behavioural Pharmacology*, 25(5 and 6), 434-444. <https://doi.org/10.1097/fbp.0000000000000044>
- Francis, G. y Jakicic, V. (2023). Equivalent statistics for a one-sample t-test. *Behavior Research Methods*, 55(1), 77-84. <https://doi.org/10.3758/s13428-021-01775-3>

- Geronikolou, S., Bacopoulou, F., Chryssanthopoulos, S. y Cokkinos, D. V. (2022). Hypertension predisposition and thermoregulation delays in adolescents with polycystic ovary syndrome: A pilot study. *Children*, 9(3), 316. <https://doi.org/10.3390/children9030316>
- Hazra, A. y Gogtay, N. (2016). Biostatistics Series Module 3: Comparing Groups: Numerical Variables. *Indian Journal of Dermatology*, 61(3), 251-260. <https://doi.org/10.4103/0019-5154.182416>
- Hines, W. G. (2000). Increased power with modified forms of the Levene (med) test for heterogeneity of variance. *Biometrics*, 56(2), 451-454. <https://doi.org/10.1111/j.0006-341X.2000.00451.x>
- Hoekstra, R., Kiers, H. A. L., & Johnson, A. (2012, January 1). Are Assumptions of Well-Known Statistical Techniques Checked, and Why (Not)? *Frontiers Media*, 3. <https://doi.org/10.3389/fpsyg.2012.00137>
- Hothorn, L. A. (2012). Model-specific tests on variance heterogeneity for detection of potentially interacting genetic loci. *BMC Genetics*, 13(1), 59. <https://doi.org/10.1186/1471-2156-13-59>
- Islam, T. (2018). Preliminary tests of homogeneity - type I error rates under non-normality. *Biostatistics and Biometrics Open Access Journal*, 6(5). <https://doi.org/10.19080/bboaj.2018.06.555699>
- Jensen, R. E., Moinpour, C. M., Potosky, A. L., Lobo, T., Hahn, E. A., Hays, R. D. y Eton, D. T. (2016). Responsiveness of 8 patient--reported outcomes measurement information system (promis) measures in a large, community--based cancer study cohort. *Cancer*, 123(2), 327-335. <https://doi.org/10.1002/cncr.30354>
- Kakizawa, Y. (2010). Comparison of Bartlett-type adjusted tests in the multiparameter case. *Journal of Multivariate Analysis*, 101(7), 1638-1655. <https://doi.org/10.1016/j.jmva.2010.03.006>
- Katsileros, A. (2024). A comparison of tests for homoscedasticity using simulation and empirical data. *Communications for Statistical Applications and Methods*, 31(1), 1-35. <https://doi.org/10.29220/csam.2024.31.1.001>
- Kelly, B. J., Groß, R., Bittinger, K., Sherrill-Mix, S., Lewis, J. D., Collman, R. G. y Li, H. (2015). Power and sample-size estimation for microbiome studies using pairwise distances and permanova. *Bioinformatics*, 31(15), 2461-2468. <https://doi.org/10.1093/bioinformatics/btv183>
- Keyes, T. K. (1997). Analysis of Levene's test under design imbalance. *Journal of Educational and Behavioral Statistics*, 22(2), 227-236. <https://doi.org/10.3102/10769986022002227>

- Kim, J. S. y Dailey, R. J. (2007). Hypothesis Testing: One--Sample Case. En *Biostatistics for Oral Healthcare* (pp. 127-157). <https://doi.org/10.1002/9780470388303.ch8>
- Kim, T. K. y Park, J. H. (2019). More about the basic assumptions of t-test: Normality and sample size. *Korean Journal of Anesthesiology*, 72(4), 331-335. <https://doi.org/10.4097/kja.d.18.00292>
- King, G. (1986). *How Not to Lie with Statistics: Avoiding Common Mistakes in Quantitative Political Science*. Wiley-Blackwell, 30(3). <https://doi.org/10.2307/2111095>
- Kroes, A. D. A. y Finley, J. R. (2023). Demystifying omega squared: Practical guidance for effect size in common analysis of variance designs. *Psychological Methods*. <https://doi.org/10.1037/met0000581>
- Langenberg, B., Janczyk, M., Koob, V., Kliegl, R. y Mayer, A. (2023). A tutorial on using the paired t-test for power calculations in repeated measures ANOVA with interactions. *Behavior Research Methods*, 55(5), 2467-2484. <https://doi.org/10.3758/s13428-022-01902-8>
- Lee, H. B., Katz, G. S. y Restori, A. F. (2010). A Monte Carlo study of seven homogeneity of variance tests. *Journal of Mathematics and Statistics*, 6(3), 359-366. <https://doi.org/10.3844/jmssp.2010.359.366>
- Levine, T. R. (2002). Eta squared, partial eta squared, and misreporting of effect size in communication research. *Human Communication Research*, 28(4), 612-625. <https://doi.org/10.1093/hcr/28.4.612>
- Lewin, M., Price, C. y Nester, C. (2023). Can a shoe--mounted imu identify the effects of orthotics in ways comparable to gait laboratory measurements? *Journal of Foot and Ankle Research*, 16(1). <https://doi.org/10.1186/s13047-023-00654-8>
- Li, H. y Johnson, T. (2014). Wilcoxon's signed-rank statistic: What null hypothesis and why it matters. *Pharmaceutical Statistics*, 13(5), 281-285. <https://doi.org/10.1002/pst.1628>
- Miller E. (1969). The signed-rank (Wilcoxon) test. *Lancet*, 1(7590), 371. [https://doi.org/10.1016/s0140-6736\(69\)91330-0](https://doi.org/10.1016/s0140-6736(69)91330-0)
- Moser, K. F. (2009). The influence of microtopography on soil nutrients in created mitigation wetlands. *Restoration Ecology*, 17(5), 641-651. <https://doi.org/10.1111/j.1526-100x.2008.00393.x>
- Nagarsenker, P. B. (1984). On Bartlett's test for homogeneity of variances. *Biometrika*, 71(2), 405. <https://doi.org/10.2307/2336260>
- O'Brien, R. G. (1981). A simple test for variance effects in experimental designs. *Psychological Bulletin*, 89(3), 570-574. <https://doi.org/10.1037/0033-2909.89.3.570>

- O'Neill, M. E. (2000). Theory & methods: a weighted least squares approach to Levene's test of homogeneity of variance. *Australian & New Zealand Journal of Statistics*, 42(1), 81-100. <https://doi.org/10.1111/1467-842X.00109>
- Odoi, B. y Samita, S. (2019). Efficiency of Bartlett and Levene's tests for testing homogeneity of variance under varying number of replicates and groups in one-way ANOVA. *International Journal of Innovative Technology and Exploring Engineering*, 8(6S4), 1219-1223. <https://doi.org/10.35940/ijitee.f1250.o486s419>
- Okada, K. (2013). Is omega squared less biased? A comparison of three major effect size indices in one-way ANOVA. *Behaviormetrika*, 40(2), 129-147. <https://doi.org/10.2333/bhmk.40.129>
- Olejnik, S. y Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods*, 8(4), 434-447. <https://doi.org/10.1037/1082-989X.8.4.434>
- Paré, G. (2010). On the use of variance per genotype as a tool to identify quantitative trait interaction effects: a report from the Women's Genome Health Study. *PLoS Genetics*, 6(6), e1000981. <https://doi.org/10.1371/journal.pgen.1000981>
- Parker, S., Jernigan, R. W. y Lansky, J. M. (2020). When is the Wilcoxon-Mann-Whitney procedure a test of location? Implications for effect-size measures. *The British Journal of Mathematical and Statistical Psychology*, 73(1), 170-183. <https://doi.org/10.1111/bmsp.12162>
- Parra-Frutos, I. (2009a). The behaviour of the modified Levene's test when data are not normally distributed. *Computational Statistics*, 24(4), 671-693. <https://doi.org/10.1007/s00180-009-0154-z>
- Parra-Frutos, I. (2009b). The impact of Levene's test of equality of variances on statistical theory and practice. *Statistical Science*, 24(3). <https://doi.org/10.1214/09-STS301>
- Pedersen, S. (2003). Effect sizes and "what if" analyses as supplements to statistical significance tests. *Journal of Early Intervention*, 25(4), 310-319. <https://doi.org/10.1177/105381510302500406>
- Romero-García, C., García, O. B., Cristóbal, M. S. S. y Asencio, E. N. (2020). Evaluación de un programa para la mejora del aprendizaje y la competencia digital en futuros docentes empleando metodologías activas. *Estudios Sobre Educación*, 39, 179-205. <https://doi.org/10.15581/004.39.179-205>

- Rosner, B., Glynn, R. J. y Lee, M. L. (2006). The Wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, 62(1), 185-192. <https://doi.org/10.1111/j.1541-0420.2005.00389.x>
- Rubia, J. M. (2022). Note on rank-biserial correlation when there are ties. *Open Journal of Statistics*, 12(05), 597-622. <https://doi.org/10.4236/ojs.2022.125036>
- Shukor, M. Y. (2021). Testing for the equality of variance for the four-parameter logistic (4PL) model for the detection of *Vibrio cholerae* DNA with polystyrene-coacrylic acid composite nanospheres. *Bulletin of Environmental Science and Sustainable Management*, 5(2), 5-7. <https://doi.org/10.54987/bessm.v5i2.647>
- Struchalin, M. V. (2010). Variance heterogeneity analysis for detection of potentially interacting genetic loci: method and its limitations. *BMC Genetics*, 11(1). <https://doi.org/10.1186/1471-2156-11-92>
- Tao, D., Xie, L., Wang, T. y Wang, T. (2014). A meta-analysis of the use of electronic reminders for patient adherence to medication in chronic disease care. *Journal of Telemedicine and Telecare*, 21(1), 3-13. <https://doi.org/10.1177/1357633x14541041>
- Verma, J. P. (2016). Nonparametric Tests in Sports Research. En *Sports Research with Analytical Solution Using* (pp. 209-245). <https://doi.org/10.1002/9781119206767.ch8>
- Wang, H. (2019). Genotype-by-environment interactions inferred from genetic effects on phenotypic variability in the UK Biobank. <https://doi.org/10.1101/519538>
- Wang, Y. (2017). Comparison of variance homogeneity tests for different distributions. *Turkiye Klinikleri Journal of Biostatistics*, 9(3), 197-212. <https://doi.org/10.5336/biostatic.2017-56316>
- Weir, J. B. D. V. (1960). Significance of the Difference between Two Means when the Population Variances may be Unequal. *Nature Portfolio*, 187(4735), 438-438. <https://doi.org/10.1038/187438a0>
- Xu, M., Fralick, D., Zheng, J. Z., Wang, B., Tu, X. y Feng, C. (2017, June 25). The Differences and Similarities Between Two-Sample T-Test and Paired T-Test. *National Institutes of Health*, 29(3), 184-188. <https://doi.org/10.11919/j.issn.1002-0829.21707>
- Zimmerman, D. W. (2004). A note on preliminary tests of equality of variances. *British Journal of Mathematical and Statistical Psychology*, 57(1), 173-181. <https://doi.org/10.1348/000711004849222>

Pruebas de hipótesis para la comparación de una variable cuantitativa en más de dos grupos

Edsaúl Emilio Pérez Guerrero

En este capítulo se revisarán dos pruebas de hipótesis utilizadas para comparar una variable cuantitativa en más de dos grupos: el ANOVA de una vía, la opción paramétrica, y el test de Kruskal-Wallis, la opción libre de parámetros.

El ANOVA se basa en descomponer la variabilidad total en dos componentes: uno proveniente de la variabilidad entre los grupos (suma de cuadrados entre grupos) y otro proveniente de la variabilidad dentro de los grupos (suma de cuadrados dentro de los grupos). Se trata de descubrir si las diferencias observadas entre los grupos pueden atribuirse a un factor específico o son simplemente producto del azar. Mientras que el test de Kruskal-Wallis se basa en la suma de rangos.

En este capítulo comenzaremos explicando los fundamentos teóricos de la prueba de ANOVA. Posteriormente explicaremos cómo realizar una prueba de ANOVA en R y cómo utilizar la librería `ggstatplot` para realizar un ANOVA. En la segunda parte del capítulo explicaremos los fundamentos teóricos de la prueba de Kruskal-Wallis, cómo realizarla en R y finalmente su aplicación en la librería `ggstatplot`.

ANOVA de una vía

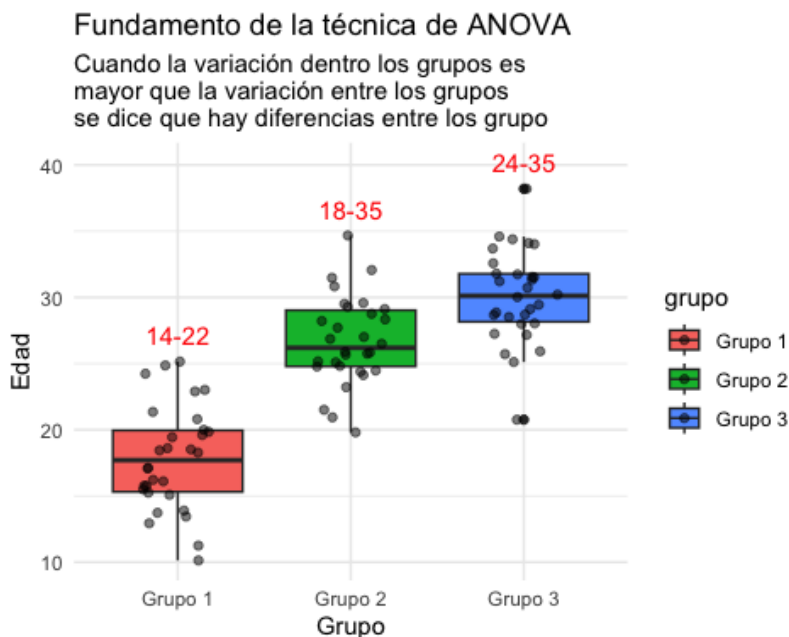
El análisis de varianza (ANOVA) de una vía es una de las técnicas estadísticas más utilizadas en investigación y en ciencias de la salud. Esta prueba permite comparar una variable numérica en tres o más grupos y establecer si existen diferencias entre estos grupos. Sin embargo, el ANOVA no puede establecer qué grupo(s) es el que presenta la diferencia.

El ANOVA de una vía compara la varianza entre los grupos con la varianza dentro de los grupos. Si la variabilidad entre los grupos es significativamente mayor que la variabilidad dentro de los grupos, se concluye que al menos un grupo es diferente de los demás en términos de la variable de interés. El ANOVA se basa en la distribución F, por lo que se debe de estimar un estadístico para esta distribución, entre más alto sea el valor de F quiere decir que la variación entre los grupos es mayor a la variación dentro de los grupos.

La [figura 17.1](#) explica de manera gráfica y simplista el fundamento de la técnica de ANOVA. Su propósito es ilustrar cómo el ANOVA descompone la variabilidad total en componentes que permiten comparar las medias de más de dos grupos. Los datos representados son las edades de tres grupos diferentes. Cada grupo tiene 30 observaciones, generadas aleatoriamente con diferentes medias y una desviación estándar similar.

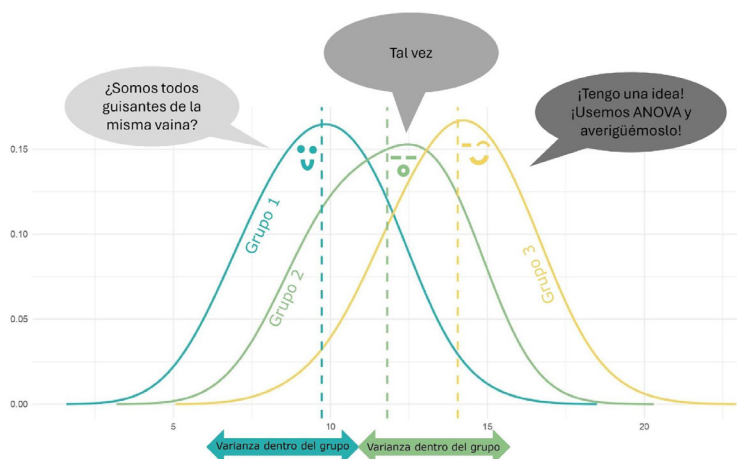
La variabilidad dentro de los grupos se observa en la dispersión de los puntos y el rango de las cajas en cada boxplot. Cada grupo muestra una variación interna que puede influir en los resultados del ANOVA. Por otro lado, la variabilidad entre los grupos se ve en las diferencias de las medianas y las posiciones de las cajas. El ANOVA compara estas diferencias para determinar si al menos un grupo tiene una media significativamente diferente de los demás. Para que el ANOVA detecte una diferencia significativa entre los grupos, la variabilidad entre los grupos debe ser mayor que la variabilidad dentro de los grupos. En este caso entre más amplias sean las diferencias de los boxplot de los grupos comparadas con la dispersión dentro del grupo (puntos), el ANOVA será significativo.

FIGURA 17.1. FUNDAMENTO DEL ANOVA



Ahora quizá el ejemplo anterior lo pueda comprender mejor con la [figura 17.2](#).

FIGURA 17.2. REPRESENTACIÓN CARICATURESCA DEL FUNDAMENTO DEL ANOVA



Dentro del “lenguaje” del ANOVA debemos entender algunas definiciones:

- ❖ **Factor:** es lo que se va a probar (variable de agrupación o variable independiente).
- ❖ **Variable de respuesta:** es lo que se va a medir (variable dependiente o de respuesta).
- ❖ **Tratamiento:** es el número de niveles o tipos de la variable de agrupación (¿cuántos grupos tengo?).

La [tabla 17.1](#) muestra la tabla clásica para realizar un ANOVA, aquí no la realizaremos manualmente, pero es conveniente explicar algunos puntos de esta.

TABLA 17.1. N = TOTAL DE DATOS, A = NÚMERO DE TRATAMIENTOS. TABLA DE ANOVA

FUENTE DE VARIACIÓN	SUMA DE CUADRADOS	GRADO DE LIBERTAD	CUADRADOS MEDIOS	FO (F CALCULADA)
Tratamientos	SC trat.	a-1	SC trat./(a-1)	CM trat./CM error
Error	SC error	N-a	SC error/(N-a)	-
Total	SC total	N-1	-	-

En la [tabla 17.1](#) podemos observar una celda “SC trat.” la cual representa la variación debida a las diferencias entre los grupos (factores). El valor de esta celda, junto con los grados de libertad (celda a-1), permite la estimación de los cuadrados medios (celda “SC trat./(a-1)”). La celda “SC error” permite a su vez estimar los cuadrados medios para el error (celda “SC error/(N-a)”), dividiendo ambos cuadrados medios, numerador para la variación entre los factores y denominador para la variación dentro de los factores, podemos estimar el valor de F. Cuanto más grande sea el valor de F, mayor será la variación entre los factores en comparación con la variación dentro de los factores y, por tanto, la prueba será significativa.

Supuestos para el ANOVA de una vía

Para aplicar correctamente una prueba de ANOVA, se deben cumplir varios supuestos:

- ♦ **Independencia de las observaciones:** cada grupo o muestra debe ser independiente de los demás. Esto significa que la selección de un individuo en un grupo no debe influir en la selección de individuos en otro grupo. El ANOVA requiere que los grupos sean independientes.
- ♦ **Normalidad:** los datos en cada grupo deben distribuirse normalmente.
- ♦ **Homogeneidad de varianzas:** las varianzas de los grupos deben ser aproximadamente iguales. Este supuesto, conocido como homocedasticidad, es importante porque la ANOVA compara las varianzas entre los grupos. Si las varianzas son muy diferentes, las conclusiones del ANOVA podrían no ser válidas. No podríamos atribuirles el error a los factores de forma correcta. Sin embargo, pueden realizarse correcciones sobre el valor de F para minimizar el efecto de las varianzas diferentes.
- ♦ **Nivel de medición:** las variables dependientes deben ser medidas al menos a nivel de intervalo o de razón.

ANOVA de una vía en R

Los pasos para realizar un ANOVA de una vía en R son los siguientes:

1. Preparar base de datos.
 2. Evaluar la homogeneidad de las varianzas (homocedasticidad).
 3. Si las varianzas son iguales, emplear:
 - » `aov()`
 4. Si son diferentes, utilizar:
 - » `oneway.test()`
- ♦ **Ejemplo 17.1:** este ejemplo muestra cómo realizar una prueba de ANOVA para varianzas iguales. Un grupo de investigadores están interesados en conocer el efecto de tres tipos de dietas sobre un grupo de ratones. Para ello diseñan un experimento en el que se pesan a los ratones después de 2 semanas de tener alguno de los 3 tipos de alimentación. A los investigadores solo les interesa la evaluación final en el peso. Los pesos de los ratones al final del estudio se encuentran en la [BasesANOVA.xlsx](#).

Evalúe si los pesos al final del estudio permiten concluir que el peso en los tres grupos de pacientes con las dietas A, B y C son diferentes. Asuma normalidad de los datos y varianzas iguales.

Siguiendo el [diagrama](#), podemos ver que la prueba ideal para este caso es el ANOVA.

Importe y descargue la base de datos [BaseANOVA.xls](#).

Una vez descargada, guárdela en su carpeta de bases e impórtela:

```
df <- readxl::read_excel("Bases/BasesANOVA.xlsx", sheet = "Dieta")
```

Puede realizar gráficas y estadística descriptiva:

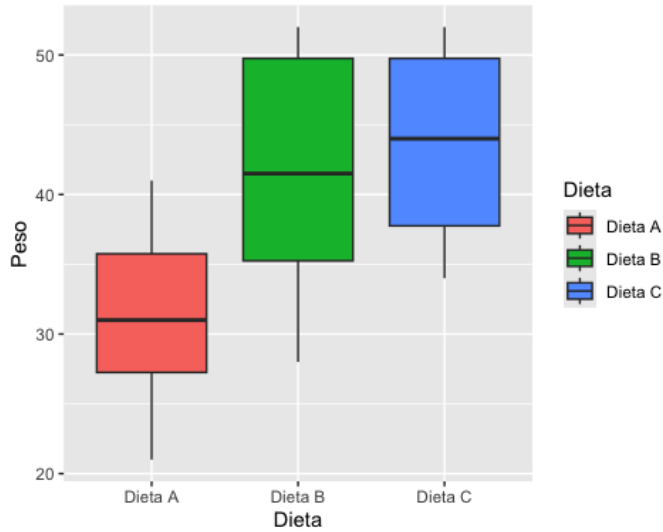
```
tapply(df$Peso, INDEX = df$Dieta, summary)

$`Dieta A`
      Min.   1st Qu. Median   Mean   3rd Qu.   Max.
 21.00    27.25    31.00    31.00    35.75    41.00

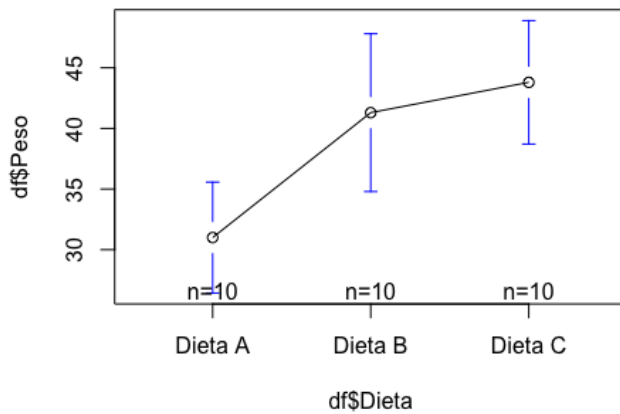
$`Dieta B`
      Min.   1st Qu. Median   Mean   3rd Qu.   Max.
 28.00    35.25    41.50    41.30    49.75    52.00

$`Dieta C`
      Min.   1st Qu. Median   Mean 3rd Qu.   Max.
 34.00    37.75    44.00    43.80    49.75    52.00

library(ggplot2)
df |>
  ggplot(mapping = aes(x=Dieta, y=Peso, fill = Dieta)) +
  geom_boxplot()
```

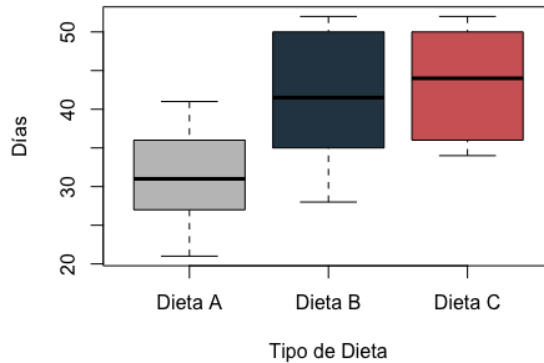


```
gplots::plotmeans(df$Peso~df$Dieta)
```



```
paleta ← c("#C1C1C1", "#2C4251", "#D16666", "#B6C649")
```

```
boxplot(df$Peso~df$Dieta, col=paleta, main="",
        ylab = "Días", xlab = "Tipo de Dieta")
```



Función aov:

```
aov(df$Peso~df$Dieta)|>
summary()
```

	Df	Sum	Sq Mean	Sq F	value	Pr(>F)
df\$Dieta	2	920.6	460.3	7.928	0.00196	**
Residuals	27	1567.7	58.1			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Los resultados anteriores indican que los tipos de dieta tienen un efecto significativo en el peso final de los ratones. Específicamente, el valor F de 7.928 y el valor p de 0.00196 sugieren que las diferencias observadas entre las medias de los grupos no son debidas al azar. Por lo tanto, podemos concluir que los pesos en los tres grupos de ratones con las dietas A, B y C son significativamente diferentes. Pero no sabemos qué grupo o qué grupos son los diferentes.

Por otro lado, en los resultados también podemos ver lo siguiente:

♦ **Grados de libertad (Df):**

- » *Dietas (df\$Dieta)*: tiene 2 grados de libertad. Esto se debe a que hay 3 tipos de dietas, y los grados de libertad entre los grupos siempre se estiman como (a-1).
- » *Residuals (Error)*: tiene 27 grados de libertad (N-a).

♦ **Suma de cuadrados (Sum Sq):**

- » *Dietas*: la suma de cuadrados para las dietas es 920.6 (variación entre los grupos).
- » *Residuals*: la suma de cuadrados residual es 1567.7 (variación dentro los grupos).

♦ **Cuadrados medios (Mean Sq):**

- » *Dietas*: los cuadrados medios para las dietas son 460.3.
- » *Residuals*: los cuadrados medios residuales son 58.1.

♦ **Estadístico F (F value):** el valor F es 7.928, calculado a partir de los cuadrados medios.

♦ **Valor P (Pr(>F)):** el valor asociado con el valor F es 0.00196.

La función `aov` asume varianzas iguales, si es necesario evaluar la homogeneidad podemos utilizar la función `bartlett.test()` o `car::leveneTest()`:

```
bartlett.test(df$Peso~df$Dieta)
```

Bartlett test of homogeneity of variances

data: df\$Peso by df\$Dieta

Bartlett's K-squared = 1.1525, df = 2, p-value = 0.562

```
car::leveneTest(df$Peso~df$Dieta)
```

Warning in `leveneTest.default(y = y, group = group, ...)`: group coerced to factor.

Levene's Test for Homogeneity of Variance (center = median)

	Df	F value	Pr(>F)
group	2	1.247	0.3034
	27		

Si se presentan varianzas diferentes, utilice la función: `oneway.test()`:

```
oneway.test(df$Peso~df$Dieta)
```

One-way analysis of means (not assuming equal variances)

data: df\$Peso and df\$Dieta

F = 9.5867, num df = 2.000, denom df = 17.674, p-value = 0.001515

Note cómo el valor de F cambia y es más pequeño. Esta prueba es más estricta.

- ♦ **Ejemplo 17.2:** este ejemplo está diseñado para realizar e interpretar la prueba de ANOVA de una vía utilizando ggstatsplot.

Vamos a retomar el [ejemplo 17.1](#) y vamos a resolverlo utilizando ggstatsplot:

```
library(ggstatsplot)
```

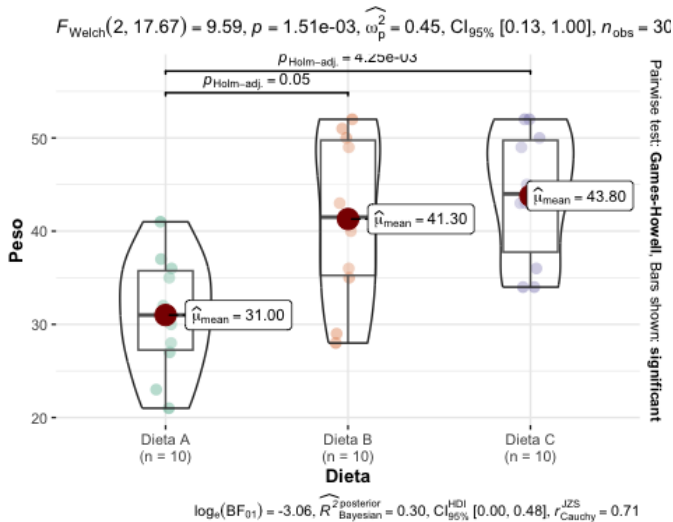
You can cite this package as:

Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.

Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167

df|>

```
ggbetweenstats( x = Dieta,
                 y = Peso,
                 type = "parametric")
```



En el gráfico podemos observar el valor de F de 9.59 para 2 y 17.76 grados de libertad. Estos valores corresponden al ajuste del valor de F que se obtuvo con la función `oneway.test()`. Se muestran las medias para grupo el valor de ω^2 de 0.45 (ver más adelante) y sus intervalos de confianza. Además, muestra el valor de p ajustado para las comparaciones múltiples. No se preocupe si de momento no comprende estos conceptos, se revisarán en el [capítulo 18](#).

Tamaño del efecto ω^2

La medida de tamaño del efecto ω^2 es una medida comúnmente utilizada en el contexto de los ANOVA. Específicamente ω^2 da información sobre la magnitud del efecto cuantificando el porcentaje total de la varianza. La interpretación de ω^2 no está bien estandarizada. Algunos autores (Casaló *et al.*, 2015) sugieren que los resultados de omega cuadrado se clasifican como asociación baja ($\omega^2 \leq 0.01$), media ($0.01 < \omega^2 < 0.14$) y alta ($\omega^2 \geq 0.14$). Mientras que otra interpretación (Rusticus, Hubley y Zumbo, 2008) sugiere que un tamaño del efecto pequeño oscila entre 0.010 y 0.058, un tamaño del efecto mediano oscila entre 0.059 y 0.137 y un tamaño del efecto grande es superior a 0.137.

Prueba de Kruskal-Wallis

Cuando se desea comparar una variable cuantitativa u ordinal en tres o más grupos y esta variable no se comporta de forma normal o la muestra es muy pequeña se utiliza la prueba de Kruskal-Wallis. Esta prueba es el equivalente no paramétrico del AONVA de una vía.

Los supuestos de esta prueba son:

1. Cada una de las muestras ha sido tomada al azar de su población.
2. Hay independencia entre las observaciones dentro de cada muestra, así como entre las muestras.
3. Los datos representan medidas por lo menos en una escala ordinal.

Los pasos para realizar la prueba de Kruskal Wallis se pueden simplificar en:

1. Tomar muestras aleatorias independientes de cada población.
2. Organizar los datos en una tabla en la que cada columna represente los grupos.
3. Asignar rangos a todas las observaciones combinadas.
4. Ordenar de menor a mayor.
5. Asignar rangos del 1 al N.
6. Para empates, usar el promedio de los rangos.
7. Estimar el estadístico KW mediante:

$$KW = \left[\frac{12}{N(N+1)} \sum_{j=1}^k n_j R_j^2 \right] - 3(N+1)$$

En la que K representa el número de grupos, n_j es el tamaño de la muestra o grupo j -ésimo, N es el total de elementos en las muestras y R_j^2 es el promedio de las posiciones en la muestra o grupo j -ésimo.

Cuando se presentan empates, la fórmula es más compleja:

$$KW = \frac{\left[\frac{12}{N(N+1)} \sum_{j=1}^k n_j R_j^2 \right] - 3(N+1)}{1 - \left[\sum_{i=1}^g (t_i^3 - t_i) \right] / (N^3 - N)}$$

En esta fórmula, g es el número de agrupaciones empatadas en la misma posición y t_i es el número de observaciones empatadas en la misma posición.

8. Valores críticos, en la mayoría de las situaciones el valor del estadístico KW se obtiene de una distribución X^2 con $k - 1$ *grados de libertad*. Esta estimación es la que utiliza R.

♦ **Ejemplo 17.3:** vamos a ejemplificar los puntos descritos anteriormente y realizar un ejemplo.

Se parte de los siguientes datos donde se muestran las edades de tres grupos de pacientes.

GRUPO A	GRUPO B	GRUPO C
23	45	18
41	55	30
54	60	34
66	70	40
78	72	44

Se desea conocer si las edades son distintas entre los tres grupos:

1. Ordenar todos los datos sin importar el grupo al que pertenecen:

18	Grupo C
23	Grupo A
30	Grupo C
34	Grupo C
40	Grupo C
41	Grupo A
44	Grupo C
45	Grupo B
54	Grupo A
55	Grupo B
60	Grupo B
66	Grupo A
70	Grupo B
72	Grupo B
78	Grupo A

2. Se le asigna un rango a cada uno de los datos:

VALOR	RANGO	GRUPO
18	1	Grupo C
23	2	Grupo A
30	3	Grupo C
34	4	Grupo C
40	5	Grupo C

VALOR	RANGO	GRUPO
41	6	Grupo A
44	7	Grupo C
45	8	Grupo B
54	9	Grupo A
55	10	Grupo B
60	11	Grupo B
66	12	Grupo A
70	13	Grupo B
72	14	Grupo B
78	15	Grupo A

3. Se suman los rangos de cada grupo:
 - » Para el Grupo A: $2+6+9+12+15=44$
 - » Para el Grupo B: $8+10+11+13+14=56$
 - » Para el Grupo C: $1+2+4+5+7=20$
4. De acuerdo o no la presencia de empates se estiman los valores de la estadística de prueba.

Kruskal Wallis en R

En R se utiliza la función `kruskal.test()`.

- ♦ **Ejemplo 17.4:** la base de datos *Cushings* contiene las concentraciones urinarias de 24 de dos esteroides medidos en 4 grupos de pacientes con distinto tipo de Cushing, codificado a (adenoma), b (hiperplasia bilateral), c (carcinoma) o u (para desconocido).

¿Los niveles de *Tetrahydrocortisone* son iguales entre los distintos tipos de Cushing?

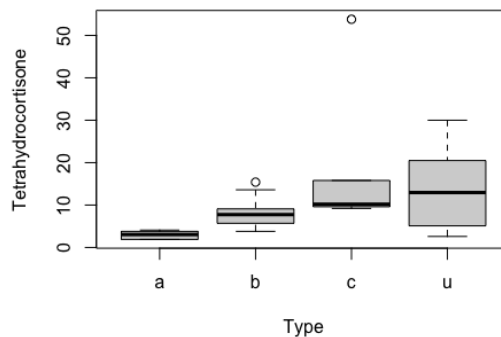
Para realizar este ejemplo cargue la base de datos “Cushings” que se encuentra en el paquete “MASS”. Si es necesario instale utilizando el comando: `library(“MASS”)`:

```
library(MASS)# Cargar la librería
data(Cushings)# Cargar la base de datos
attach(Cushings)# Adjuntar base de datos
names(Cushings)# Para conocer como se llaman las variables
[1] "Tetrahydrocortisone" "Pregnanetriol"      "Type"
```

Siempre es importante realizar gráficas para ver la distribución, valores atípicos y tendencia de los datos:

```
boxplot(Tetrahydrocortisone~Type, main="Niveles de Tetrahydrocorti-
sone en los pacientes con Cushing")
```

Niveles de Tetrahydrocortisone en los pacientes con C



Ahora usamos la función `kruskal.test()`. Esta función requiere del uso de fórmula para su realización, es decir, los datos deben de ser cargados en la función de la forma *Variable de interés~Variable de agrupación*:

```
kruskal.test(Tetrahydrocortisone~Type)
```

Kruskal-Wallis rank sum test

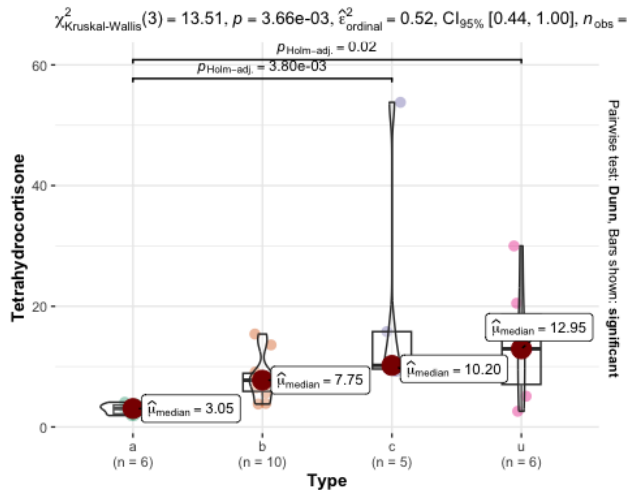
data: Tetrahydrocortisone by Type

Kruskal-Wallis chi-squared = 13.509, df = 3, p-value = 0.003655

Los resultados nos indican que el valor del estadístico KW es de 13.509 con 3 grados de libertad (df). El valor de p indica que existe evidencia para rechazar la H_0 y por tanto podrías concluir que los niveles de *Tetrahydrocortisone* son diferentes entre los tipos de Chushing.

Utilizando `ggstatplot` la resolución de este ejemplo sería:

```
Cushings|>
  ggbetweenstats(x = Type,
                 y = Tetrahydrocortisone,
                 type = "np")
```



Ejercicios

- ♦ **Ejercicio 17.1:** se sabe por antecedentes previos que los grupos de edad de menores de 30, de 31 a 45 y más de 45 presentan niveles de glucosa diferentes en la mujeres de Pima. Pruebe esta afirmación utilizando una prueba de hipótesis. Utilice `Pima.tr2` y un $\alpha = 0.05$.

Resolución ejercicio 17.1

```
library(MASS)
data("Pima.tr2")
Pima.tr2$Edad_Factor ←
  cut(Pima.tr2$age,
      breaks = c(0,30,45,Inf),
      labels = c("Menos de 30",
                  "31-45",
                  "Mas de 45"))
```

Se evalúa la normalidad:

```
nortest::lillie.test(Pima.tr2$glu)
library(ggplot2)
library(ggrain)
Pima.tr2|>
  ggplot(mapping = aes(x=1, y=glu))+
  geom_rain()
```

Claramente no tienen una distribución normal, por lo tanto, debemos de utilizar Kruskal Wallis:

```
kruskal.test(Pima.tr2$glu~Pima.tr2$Edad_Factor)
```

- ♦ **Ejercicio 17.2:** Ahora pruebe si hay diferencias en el IMC.

Resolución ejercicio 17.2

```
nortest::lillie.test(Pima.tr2$bmi)
Pima.tr2|>
  ggplot(mapping = aes(x=1, y=bmi))+
  geom_rain()
```

Parece que los datos siguen una distribución normal, por lo tanto, debemos de utilizar ANOVA:

```
car::leveneTest(Pima.tr2$bmi~Pima.tr2$Edad_Factor)
```

Los grupos no tienen varianzas iguales:

```
oneway.test(Pima.tr2$bmi~Pima.tr2$Edad_Factor)
```

- ♦ **Ejercicio 17.3:** repita este análisis para las variables skin, bp, npreg y ped.
Resolución ejercicio 17.3

Se resuelve igual que el [ejercicio 17.1](#) y el [ejercicio 17.2](#).

- ♦ **Ejercicio 17.4:** utilice la base de datos [S1Daraset.xlsx](#) del [ejercicio 16.3](#) para responder las siguiente preguntas:

1. ¿La edad es igual entre los pacientes con lupus con proteinuria (SLE with proteinuria), pacientes con lupus sin nefropatía (SLE without LN) y el grupo control (Control)? Utilice la variable Groups_NLSLEvsSLE para agrupar.
2. ¿Las concentraciones de leptina son iguales entre los pacientes con lupus con proteinuria (SLE with proteinuria), pacientes con lupus sin nefropatía (SLE without LN) y el grupo control (Control)? Utilice la variable Groups_NLSLEvsSLE para agrupar.
3. ¿Las concentraciones de adiponectina son iguales entre los pacientes con lupus con proteinuria (SLE with proteinuria), pacientes con lupus sin nefropatía (SLE without LN) y el grupo control (Control)? Utilice la variable Groups_NLSLEvsSLE para agrupar.
4. ¿El índice de masa corporal es igual entre los pacientes con lupus con proteinuria (SLE with proteinuria), pacientes con lupus sin nefropatía (SLE without LN) y el grupo control (Control)? Utilice la variable Groups_NLSLEvsSLE para agrupar.

Resolución ejercicio 17.4

Este ejercicio se puede resolver utilizando las resoluciones de [ejercicio 16.3](#), [ejercicio 17.1](#) y [ejercicio 17.2](#).

Referencias

- Acar, E. F. y Sun, L. (2013). A generalized Kruskal-Wallis test incorporating group uncertainty with application to genetic association studies. *Biometrics*, 69(2), 427-435. <https://doi.org/10.1111/biom.12006>
- Afzal, A. y Awan, M. (2011). Computationally intelligent statistical techniques for causal inferences: An application to the impact of export on GDP of Pakistan. *Procedia Computer Science*, 3, 255-259. <https://doi.org/10.1016/j.procs.2010.12.232>
- Burger, T. (2023). Controlling for false discoveries subsequently to large scale one-way ANOVA testing in proteomics: Practical considerations. *Proteomics*, 23(18), e2200406. <https://doi.org/10.1002/pmic.202200406>
- Casaló, L. V., Flavián, C., Guinalú, M. y Ekinci, Y. (2015). Avoiding the dark side of positive online consumer reviews: enhancing reviews' usefulness for high risk-averse travelers. *Journal of Business Research*, 68(9), 1829-1835. <https://doi.org/10.1016/j.jbusres.2015.01.010>
- Celis de la Rosa, A. de J. (2014). Pruebas de hipótesis para una y dos variables cuantitativas. En A. de J. Celis de la Rosa y V. Labrada Martagón, *Bioestadística* (3ª ed., pp. 135-143). Editorial El Manual Moderno.
- Chan, Y. y Walmsley, R. P. (1997). Learning and understanding the Kruskal-Wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups. *Physical therapy*, 77(12), 1755-1762. <https://doi.org/10.1093/ptj/77.12.1755>
- Chatzi, A. y Doody, O. (2023). The one-way ANOVA test explained. *Nurse researcher*, 31(3), 8-14. <https://doi.org/10.7748/nr.2023.e1885>
- Clark, J. S. C., Kulig, P., Podsiadło, K., Rydzewska, K., Arabski, K., Białecka, M., Safranow, K. y Ciechanowicz, A. (2023). Empirical investigations into Kruskal-Wallis power studies utilizing Bernstein fits, simulations and medical study datasets. *Scientific reports*, 13(1), 2352. <https://doi.org/10.1038/s41598-023-29308-2>
- Fan, C. y Zhang, D. (2012). A note on power and sample size calculations for the Kruskal-Wallis test for ordered categorical data. *Journal of biopharmaceutical statistics*, 22(6), 1162-1173. <https://doi.org/10.1080/10543406.2011.578313>
- Fan, C., Zhang, D. y Zhang, C. H. (2011). On sample size of the kruskal-wallis test with application to a mouse peritoneal cavity study. *Biometrics*, 67(1), 213-224. <https://doi.org/10.1111/j.1541-0420.2010.01407.x>

- Hazra, A. y Gogtay, N. (2016). Biostatistics Series Module 3: Comparing Groups: Numerical Variables. *Indian journal of dermatology*, 61(3), 251-260. <https://doi.org/10.4103/0019-5154.182416>
- Indrayan, A. y Malhotra, R. K. (2017). *Inference from means* (4ª ed., pp. 377-422). Chapman & Hall/CRC Biostatistics Series.
- Jan, S. L. y Shieh, G. (2014). Sample size determinations for Welch's test in one-way heteroscedastic ANOVA. *The British journal of mathematical and statistical psychology*, 67(1), 72-93. <https://doi.org/10.1111/bmsp.12006>
- Kim, T. K. (2017). Understanding one-way ANOVA using conceptual figures. *Korean journal of anesthesiology*, 70(1), 22-26. <https://doi.org/10.4097/kjae.2017.70.1.22>
- Mishra, P., Singh, U., Pandey, C. M., Mishra, P. y Pandey, G. (2019). Application of student's t-test, analysis of variance, and covariance. *Annals of cardiac anaesthesia*, 22(4), 407-411. https://doi.org/10.4103/aca.ACA_94_19
- Pérez-Guerrero, E. E., Guillén-Medina, M. R., Márquez-Sandoval, F., Vera-Cruz, J. M., Gallegos-Arreola, M. P., Rico-Méndez, M. A., Aguilar-Velázquez, J. A. y Gutiérrez-Hurtado, I. A. (2024). Methodological and statistical considerations for cross-sectional, case-control, and cohort studies. *Journal of Clinical Medicine*, 13, 4005. <https://doi.org/10.3390/jcm13144005>
- Rusticus, S. A., Hubley, A. M. y Zumbo, B. D. (2008). Measurement invariance of the appearance schemas inventory-revised and the body image quality of life inventory across age and gender. *Assessment*, 15(1), 60-71. <https://doi.org/10.1177/1073191107306805>
- Sherwani, R. A. K., Shakeel, H., Awan, W. B., Faheem, M. y Aslam, M. (2021). Analysis of COVID-19 data using neutrosophic Kruskal Wallis H test. *BMC medical research methodology*, 21(1), 215. <https://doi.org/10.1186/s12874-021-01410-x>
- Theodorsson-Norheim, E. (1986). Kruskal-Wallis test: BASIC computer program to perform nonparametric one-way analysis of variance and multiple comparisons on ranks of several independent samples. *Computer methods and programs in biomedicine*, 23(1), 57-62. [https://doi.org/10.1016/0169-2607\(86\)90081-7](https://doi.org/10.1016/0169-2607(86)90081-7)
- Zhang, B. y Zhang, Y. (2009). Mann-Whitney U test and Kruskal-Wallis test should be used for comparisons of differences in medians, not means: comment on the article by van der Helm-van Mil et al. *Arthritis and rheumatism*, 60(5), 1565. <https://doi.org/10.1002/art.24497>

Pruebas post hoc

Edsaúl Emilio Pérez Guerrero

Después de obtener un resultado significativo y no aceptar la H_0 en un ANOVA o en una prueba de Kruskal-Wallis, el siguiente objetivo metodológico es determinar cuál de los grupos bajo estudio contribuye a esta significación estadística. Recordemos que estas pruebas solo nos dicen que la variable de estudio es diferente entre los grupos, pero no nos dice entre cuáles. Específicamente, se busca identificar qué grupos muestran diferencias significativas entre sí.

En una situación donde se comparan cuatro grupos distintos, se necesitarían efectuar un total de seis comparaciones emparejadas:

- ♦ Grupo 1 vs. Grupo 2: Comparación 1
- ♦ Grupo 1 vs. Grupo 3: Comparación 2
- ♦ Grupo 1 vs. Grupo 4: Comparación 3
- ♦ Grupo 2 vs. Grupo 3: Comparación 4
- ♦ Grupo 2 vs. Grupo 4: Comparación 5
- ♦ Grupo 3 vs. Grupo 4: Comparación 6

Para abordar estas comparaciones múltiples, se pueden emplear dos estrategias principales:

1. Aplicar seis pruebas de Student independientes. ¡Error!
2. Utilizar pruebas *post hoc* que permiten la corrección y disminución de errores.

En la primera estrategia, el uso repetido de pruebas t de Student aumenta considerablemente la probabilidad de incurrir en un error de Tipo I (es decir, detectar diferencias significativas donde en realidad no las hay). Matemáticamente, la probabilidad acumulada de cometer un error de Tipo I se calcula mediante la fórmula: $r = 1 - (1 - \alpha)^n$, donde: α es el nivel de significancia individual y n es el número de pruebas.

Por ejemplo, con seis pruebas y un α de 0.05, el error acumulado de Tipo I sería calculado como $r = 1 - (1 - 0.05)^6 = 0.26$. Si se incrementa el número de pruebas a 15, el riesgo de error de Tipo I superaría el 50%, haciendo que la estrategia sea comparable a lanzar una moneda al aire.

A continuación, se presentan ejemplos de cálculos para diferentes números de comparaciones:

```
1-(1-.05)^4 # para 4 comparaciones
[1] 0.1854938
1-(1-.05)^6 # para 6 comparaciones
[1] 0.2649081
1-(1-.05)^10 # para 10 comparaciones
[1] 0.4012631
1-(1-.05)^15 # para 15 comparaciones
[1] 0.5367088
1-(1-.05)^20 # para 20 comparaciones
[1] 0.6415141
```

Cuando se permite un error de Tipo I para cada comparación individual, este nivel de error se denomina tasa de error de comparación. En contraposición, el error total acumulado a lo largo de todas las comparaciones se conoce como tasa de error de familiar (Familywise Error Rate/Alpha Inflation). Para mantener el error Tipo I experimental dentro de un límite preestablecido (de 0.05), existen diversas técnicas para comparaciones múltiples (comparaciones *post hoc*). Estos procedimientos suelen llevar el nombre del científico que los desarrolló inicialmente y su elección depende de múltiples factores. Entre los métodos *post hoc* más comunes se encuentran Bonferroni, Tukey, Scheffe, Fisher LSD, Newman-Keuls, Duncan y Dunnett.

¿Cómo seleccionar una prueba *post hoc*?

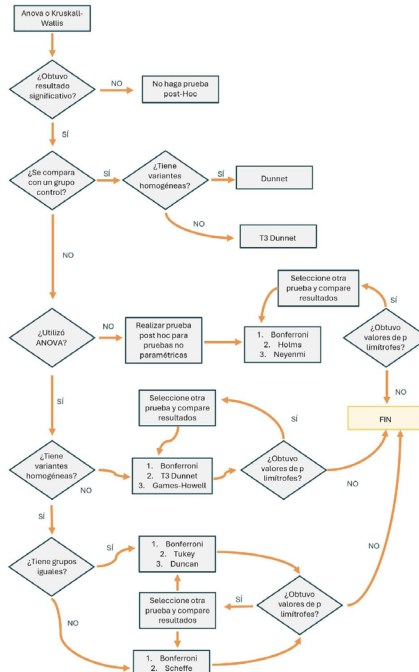
La respuesta es extremadamente complicada. No existe una única respuesta.

Antes de proceder con la selección de un test *post hoc*, recomiendo responder a las siguientes interrogantes:

1. ¿Cuántos grupos se van a comparar?
2. ¿Los grupos presentan varianzas homogéneas?
3. ¿Existe un grupo que servirá como referencia?
4. ¿Los datos se ajustan a una distribución normal?

La [figura 18.1](#) muestra un árbol de decisión para la selección de una prueba *post hoc*. Este árbol no contiene todas las respuestas ni todas las pruebas *post hoc* disponibles. Recomendamos que solo sea tomado como ayuda y no como una ley. Es importante recordar que la [figura 18.1](#) es una guía, es meramente orientativa y no debe considerarse como regla estricta.

FIGURA 18.1. ÁRBOL DE DECISIÓN PARA PRUEBAS POST-HOC



Los siguientes apartados describen algunas de las pruebas *post hoc* donde se describe el fundamento, ventajas, desventajas y su implementación en R.

Método de Bonferroni

Introducción al método de Bonferroni

El método de Bonferroni constituye una estrategia elemental pero eficaz para el control del error de Tipo I en múltiples comparaciones estadísticas. Específicamente, el nivel de significancia global se divide entre el número total de hipótesis secundarias (H) que están siendo examinadas. En otras palabras, el valor crítico ajustado de se determina como α/H .

Consideraciones sobre la potencia estadística

El método de Bonferroni es extremadamente conservador, lo cual lleva a una alta tasa de errores de Tipo II, especialmente cuando el número de comparaciones (H) es grande. Entre más comparaciones se hagan más estricto será el método de Bonferroni.

En algunas ocasiones H se limita al número de comparaciones específicas de interés. Por ejemplo, si se tienen cuatro grupos y solo se desea comparar ciertos grupos entre sí, entonces H se ajustará acorde.

Ventajas y desventajas del método de Bonferroni

Ventajas

- ♦ Simplicidad
- ♦ H no excede el número total de comparaciones de interés
- ♦ Especialmente útil en los casos que se requieren pocas comparaciones
- ♦ Es el comodín de las pruebas *post hoc*

Desventajas

- ♦ Disminución en la potencia del test al aumentar el número de comparaciones
- ♦ Entre más comparaciones se realicen, más estricta es la prueba
- ♦ No es una prueba flexible
- ♦ No es útil para comparaciones complejas

Implementación del método de Bonferroni en R

Para implementar el método de Bonferroni en R, se puede utilizar la función `pairs.t.test`. A continuación, se describen los principales argumentos de esta función:

```
# Descripción de los argumentos de pairs.t.test
- x: variable cuantitativa, vector de respuesta
- g: variable de agrupación
- alternative: prueba de hipótesis bilateral o unilateral: "two.sided"
  (default), "greater" or "less"
- paired: valor lógico para grupos pareados
- p.adjusted.method; Métodos para la obtención de el valor ajusta de
  $p$ "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr",
  "none"
```

La función `pairs.t.test` en R se usa para realizar pruebas t de Student por pares en datos con dos o más grupos. Esta función es especialmente útil cuando se quiere comparar las medias de varios grupos entre sí, después de haber encontrado diferencias significativas en un análisis de varianza (ANOVA).

Procedimiento de bonferroni en R

- ♦ **Ejemplo 18.1:** utilizando la librería MASS y la base de datos Cushings realice una prueba de ANOVA e indique entre qué grupos del tipo Cushings se presentan la diferencias. Utilice el procedimiento de corrección de Bonferroni.

Para resolver este ejemplo utilizaremos la base de datos Cushings del paquete MASS.

1. Cargar base de datos:

```
library(MASS)
data("Cushings")
head(Cushings) # Visualizar los primeros 6 datos
```

	Tetrahydrocortisone	Pregnanetriol	Type
a1	3.1	11.70	a
a2	3.0	1.30	a
a3	1.9	0.10	a
a4	3.8	0.04	a
a5	4.1	1.10	a
a6	1.9	0.40	a

2. Correr ANOVA para:

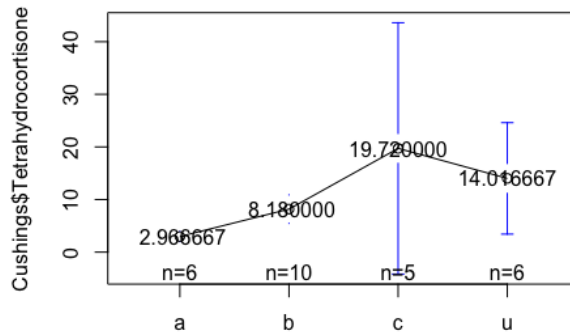
```
aov(Tetrahydrocortisone~Type, data=Cushings)|>
summary()
```

	Df	Sum	Sq	MeanSq	F	value	Pr(>F)
Type	3	893.5	297.84	3.226	0.0412	*	
Residuals	23	2123.6	92.33				

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3. Graficar medias utilizando la función plotmeans de la librería (gplots).
Puede utilizar cualquier otra gráfica:

```
install.packages("gplots") # Instale si es necesario
library(gplots)
plotmeans(Cushings$Tetrahydrocortisone~Cushings$Type,
          mean.labels = T)
```

```
# El último argumento muestra los valores de las medias
```

4. Calcular el valor de p ajustado:

```
pairwise.t.test(Cushings$Tetrahydrocortisone,Cushings$Type,
                p.adjust.method = "bonferroni",
                pool.sd=FALSE)
```

Pairwise comparisons using t tests with non-pooled SD

data: Cushings\$Tetrahydrocortisone and Cushings\$Type

	a	b	c
b	0.01	-	-
c	0.74	1.00	-
u	0.26	1.00	1.00

P value adjustment method: bonferroni

En el código anterior es necesario cambiar el argumento `pool.sd` para que no tome un solo valor de desviación estándar y se tome el valor para cada comparación. La desviación estándar agrupada es un promedio ponderado de las desviaciones estándar de dos o más grupos. Las desviaciones estándar individuales se promedian, con más “peso” dado a tamaños de muestra más grandes.

Los resultados anteriores indican:

- ♦ La comparación del grupo a vs. grupo b es significativa, ya que su valor de p ajustado por el método de bonferroni es 0.01
- ♦ La comparación del grupo a vs. grupo c no es significativa, ya que su valor de p ajustado por el método de bonferroni es 0.74
- ♦ La comparación del grupo a vs. grupo u no es significativa, ya que su valor de p ajustado por el método de bonferroni es 0.26
- ♦ La comparación del grupo b vs. grupo c no es significativa, ya que su valor de p ajustado por el método de bonferroni es 1.00
- ♦ La comparación del grupo b vs. grupo u no es significativa, ya que su valor de p ajustado por el método de bonferroni es 1.00
- ♦ La comparación del grupo c vs. grupo u no es significativa, ya que su valor de p ajustado por el método de bonferroni es 1.00

Los valores sin ajustar de para las comparaciones del grupo a vs. b fueron:

Welch Two Sample t-test

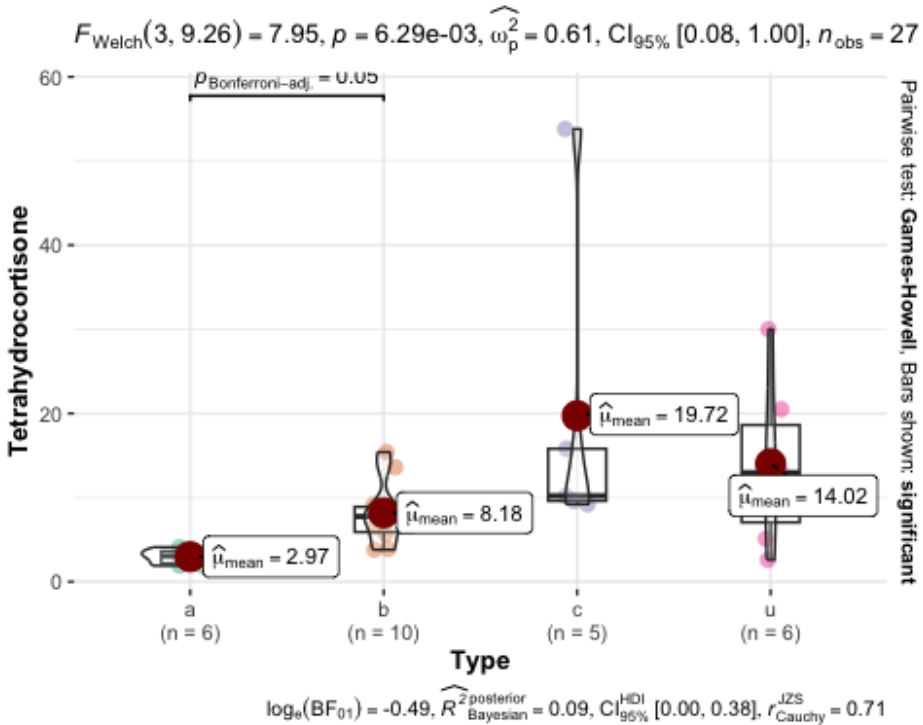
```
data: ab$Tetrahydrocortisone by ab$Type
t = -4.1499, df = 10.685, p-value = 0.001719
alternative hypothesis: true difference in means between group a and
group b is not equal to 0
95 percent confidence interval:
 -7.988307 -2.438360
sample estimates:
mean in group a mean in group b
    2.966667      8.180000
```

Si multiplicamos el valor de p anterior por el número de comparaciones obtenemos el valor de p ajustado. $0.001719 \times 6 = 0.0103$

- ♦ **Ejemplo 18.2:** ejemplo del procedimiento de Bonferroni utilizando ggstatplot:

```
Cushings|>
```

```
ggstatsplot::ggbetweenstats(x=Type,
                             y=Tetrahydrocortisone,
                             type = "parametric",
                             p.adjust.method = "bonferroni")
```



♦ **Ejemplo 18.3:** ejemplo utilizando la librería `rstatix`.

`rstatix` es un paquete R que proporciona un conjunto de funciones para simplificar las operaciones de análisis de datos estadísticos. El paquete sigue el enfoque de “tidy data”, lo que facilita la manipulación de datos y análisis cuando se usa en combinación con otros paquetes del ecosistema `tidyverse`.

Algunas de las funciones más utilizadas en `rstatix` son:

- ♦ `anova_test()`: para realizar una prueba ANOVA.
- ♦ `t_test()`: para realizar pruebas t.

- ◇ `kruskal_test()`: para realizar la prueba de Kruskal-Wallis, que es una alternativa no paramétrica a la ANOVA.
- ◇ `compare_means()`: para realizar comparaciones *post hoc* después de una ANOVA.
- ◇ `correlation_test()`: para realizar pruebas de correlación.

Se muestra el código para la realización de la corrección de Bonferroni utilizando la librería `rstatix`:

```
Cushings|>
  rstatix::t_test(Tetrahydrocortisone~Type,
                  p.adjust.method = "bonferroni")

# A tibble: 6 × 10
   .y. group1 group2 n1  n2 statistic df      p    p.adj p.adj.signif
* <chr>   <chr> <chr> <int> <int>   <dbl> <dbl> <dbl> <dbl>   <chr>
1 Tetrahydro... a     b      6    10    -4.15  10.7  0.002  0.01    *
```

	.y.	group1	group2	n1	n2	statistic	df	p	p.adj	p.adj.signif
1	Tetrahydro...	a	b	6	10	-4.15	10.7	0.002	0.01	*
2	Tetrahydro...	a	c	6	5	-1.95	4.02	0.123	0.738	ns
3	Tetrahydro...	a	u	6	6	-2.67	5.08	0.044	0.262	ns
4	Tetrahydro...	b	c	10	5	-1.33	4.16	0.252	1	ns
5	Tetrahydro...	b	u	10	6	-1.36	5.86	0.224	1	ns
6	Tetrahydro...	c	u	5	6	0.598	5.80	0.573	1	ns

Con el código anterior se realizan una serie de pruebas t de Student para comparar las medias de la variable `Tetrahydrocortisone` entre diferentes grupos especificados por la variable `Type`. Además, ajusta los valores *p* utilizando el método de Bonferroni para controlar la tasa de error de Tipo I.

Tukey HSD Test

También conocido como Tukey HSD (“honestly significant difference” o “honest significant difference”). Este es el procedimiento más adecuado cuando el interés está en realizar todas las comparaciones por pares. Requiere igualdad de varianzas.

Para esta prueba se necesitan los valores de J y de v , además de un valor de tablas (Q). Se utiliza la siguiente fórmula:

$$Tukey Test = D_{ij} = Q \sqrt{\frac{MSE}{2} \left(\frac{1}{n_i} + \frac{1}{n_j} \right)}$$

Donde:

- ♦ n_i y n_j son el número de sujetos en el i -ésimo y j -ésimo grupo respectivamente.
- ♦ J es el número de grupos a comparar.
- ♦ v son los grados de libertad asociados a los cuadrados medios del error (MSE) en la tabla ANOVA.
- ♦ Q proviene de una distribución del rango estudentizado para un especificado.

En R puedes utilizar la función `qtukey()` para estimar Q .

- ♦ **Ejemplo 18.4:** para resolver este ejemplo, partimos de los siguientes datos que muestran el peso medio de los recién nacidos en grupos de madres que fumaron a distintos tiempos durante el embarazo:

TIEMPO DE DURACIÓN DEL TABAQUISMO EN EL EMBARAZO	PESO EN (KG)
< 18 semanas	3.44 (n = 34)
18–31 semanas	3.39 (n = 24)
32 + semanas	3.27 (n = 77)

Después de realizar un ANOVA se sabe que el peso se asocia con la duración del tabaquismo y es estadísticamente significativo. Se procede a comparar los pesos medios al nacer para diferentes duraciones de tabaquismo. Los investigadores están interesados en las diferencias entre cualquiera de los grupos. Por lo tanto, dado que se necesitan todas las comparaciones por pares, la prueba de Tukey sería la más apropiada. Se conoce la siguiente información:

- ♦ $J = 3$
- ♦ $n = 135$
- ♦ $MSE = 0.022$ obtenido del ANOVA
- ♦ $v = 126$
- ♦ $Q = 3.36$ para $\alpha = 0.05$

Ahora se procede al cálculo de D_{ij} (la diferencia mínima esperada):

$$D_{12} = 3.36 \sqrt{\frac{0.022}{2} \left(\frac{1}{34} + \frac{1}{24} \right)} = 0.0940$$

$$D_{13} = 3.36 \sqrt{\frac{0.022}{2} \left(\frac{1}{34} + \frac{1}{77} \right)} = 0.0726$$

$$D_{23} = 3.36 \sqrt{\frac{0.022}{2} \left(\frac{1}{24} + \frac{1}{77} \right)} = 0.0824$$

Interpretación

D_{ij} es la diferencia crítica que debe estar presente entre las medias de los grupos i y j para que sea estadísticamente significativa en . Por ejemplo:

$$| \bar{x}_1 - \bar{x}_2 | = 0.10 > 0.0940 \Rightarrow \text{Se Rechaza } H_0$$

$$| \bar{x}_1 - \bar{x}_3 | = 0.001 > 0.0726 \Rightarrow \text{Se Acepta } H_0$$

El valor de p ajustada corresponde a la probabilidad de la distribución de Tukey. En R, esto se estima mediante la función `ptukey`.

Ventajas y desventajas de la prueba de Tukey

ASPECTO	VENTAJAS	DESVENTAJAS
Error Tipo I	Controla eficazmente la tasa de error tipo I.	–
Tipo de comparaciones	Útil para comparaciones de todos contra todos.	No es eficiente para comparaciones planeadas o específicas.
Simplicidad y comodidad	Fácil de realizar y entender.	–
Poder estadístico	Mayor poder estadístico en comparación con Bonferroni.	–
Supuestos estadísticos	Requiere pocos supuestos y es robusta a pequeñas desviaciones.	Necesita que los datos cumplan con normalidad y homocedasticidad.
Sensibilidad a atípicos	–	Sensible a valores atípicos.
Carga computacional	–	Puede ser computacionalmente intensivo con muchos grupos.

Supuestos para la prueba de Tukey

- ◊ Normalidad de los datos
- ◊ Homogeneidad de varianzas
- ◊ n igual para los grupos

Tukey en R

- ◊ **Ejemplo 18.5:** en R podemos utilizar la función `TukeyHSD` y aplicarla a un ANOVA, o bien la función `tukey_hsd` de la librería `rstastix`. Si retomamos el ejemplo del ejercicio anterior:

```
aov(Cushings$Tetrahydrocortisone~Cushings$Type)|>
  TukeyHSD()
Tukey multiple comparisons of means
 95% family-wise confidence level

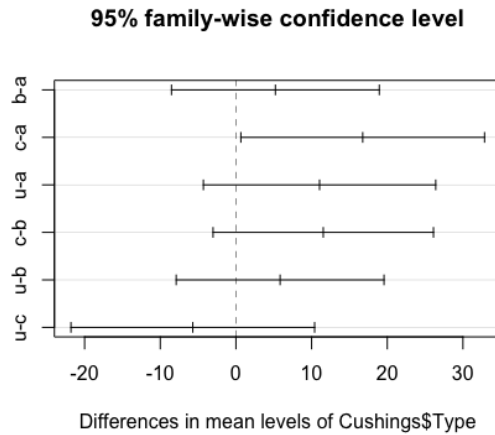
Fit: aov(formula = Cushings$Tetrahydrocortisone ~ Cushings$Type)

$`Cushings$Type`
      diff      lwr      upr      p adj
b-a  5.213333 -8.5181810 18.94485 0.7219961
c-a 16.753333  0.6517055 32.85496 0.0393613
u-a 11.050000 -4.3022998 26.40230 0.2200574
c-b 11.540000 -3.0244704 26.10447 0.1551716
u-b  5.836667 -7.8948477 19.56818 0.6473881
u-c -5.703333 -21.8049612 10.39829 0.7619359
```

Los resultados anteriores indican la diferencia de las medias estimadas, los intervalos de confianza y el valor de p ajustado por el método de Tukey. Las medias y los intervalos de confianza corresponden a la resta de la media de un grupo vs. la de otro. Por ejemplo, si se comparan el grupo a vs. el grupo b, las diferencias de medias sería: $2.9666667(\text{media del grupo a}) - 8.18(\text{media del grupo b}) = -5.2133333$. La interpretación para la comparación del grupo b con el grupo a es como sigue: la diferencia de sus medias fue de 5.21, el IC bajo fue de -8.51, el IC alto fue de 18.94 y el valor de p ajustados por el método de Tukey fue de 0.721.

Se pueden graficar las diferencias de las medias utilizando el siguiente código: plot

```
aov(Cushings$Tetrahydrocortisone~Cushings$Type)|>
  TukeyHSD()|>
  plot()
```



Utilizando rstatix el código es:

```
aov(Tetrahydrocortisone~Type, data=Cushings)|>
  rstatix ::tukey_hsd()
# A tibble: 6 × 9
  term  group1 group2 null.value estimate conf.low conf.high p.adj
p.adj.signif
* <chr> <chr> <chr>    <dbl>    <dbl>    <dbl>    <dbl> <dbl> <chr>
1 Type  a      b         0      5.21    -8.52    18.9  0.722 ns
2 Type  a      c         0     16.8     0.652   32.9  0.0394 *
3 Type  a      u         0     11.0    -4.30    26.4  0.22  ns
4 Type  b      c         0     11.5    -3.02    26.1  0.155 ns
5 Type  b      u         0      5.84    -7.89    19.6  0.647 ns
6 Type  c      u         0     -5.70   -21.8    10.4  0.762 ns
```


Test de Dunnet

En ciertas situaciones, el interés está en **comparar varios grupos contra un grupo de referencia específico**, en lugar de comparar cada grupo con todos los demás. Este es el caso, por ejemplo, cuando se desean comparar las concentraciones de una citocina en varios grupos de pacientes con un grupo control. Otro ejemplo de este tipo de comparaciones sería si en un estudio sobre la enfermedad de Cushing, el “Tipo a” de Cushing es nuestro grupo de referencia. Nos interesaría entonces comparar este grupo con los “Tipos b, c y u” de la enfermedad.

La estadística de prueba de Dunnett para comparar la media del i -ésimo grupo de tratamiento $\bar{x}_i$ con la media del grupo de control $\bar{x}_c$ se calcula como:

$$D = \frac{\bar{x}_i - \bar{x}_c}{\sqrt{2 \times MSE \times \left(\frac{1}{n_i} + \frac{1}{n_c} \right)}}$$

Donde:

- ♦ $\bar{x}_i$ y $\bar{x}_c$ son las medias de la muestra del i -ésimo grupo de tratamiento y el grupo de control, respectivamente.
- ♦ MSE es el error cuadrático medio, generalmente obtenido de una tabla ANOVA.
- ♦ n_i y n_c son los tamaños de muestra del i -ésimo grupo de tratamiento y el grupo de control, respectivamente.

El estadístico D se compara con un valor crítico obtenido de la distribución de Dunnett, que es una distribución específica que tiene en cuenta múltiples comparaciones.

La hipótesis que se prueba en el test de Dunnet es:

- ♦ Hipótesis nula (H_0): la media del grupo de tratamiento i es igual a la media del grupo de control.
- ♦ Hipótesis alternativa (H_A): la media del grupo de tratamiento es diferente de la media del grupo de control.

Test de Dunnet en R

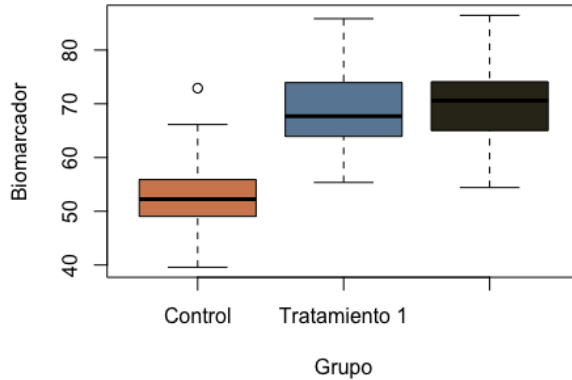
- ♦ **Ejemplo 18.6:** supongamos que es de nuestro interés comparar las concentraciones de un biomarcador en un grupo control y dos grupos con un tratamiento experimental. Para ello, vamos a crear un conjunto de datos ficticios:

```
set.seed(1234) # Sembrando la semilla
datos <- data.frame(Grupo =
                    rep(c("Control", "Tratamiento 1",
                        "Tratamiento 2"), each = 50),
                    Biomarcador = c(rnorm(50, mean=56, sd = 7),
                                    rnorm(50, mean=68, sd= 7),
                                    rnorm(50, mean=70, sd = 8)))
datos$Grupo <- as.factor(datos$Grupo)
head(datos) # visualizar los primeros 6 datos
```

	Grupo	Biomarcador
1	Control	47.55054
2	Control	57.94200
3	Control	63.59109
4	Control	39.58012
5	Control	59.00387
6	Control	59.54239

Podemos visualizar los datos:

```
boxplot(Biomarcador~Grupo, data=datos,
        col=c("#D3885E", "#6988A3", "#33301D",
              "#F08080", "#4682B4", "#FF69B4"),
        main="Concentraciones de un biomarcador"))
```



Procedemos a realizar la prueba de Levene para igualdad de varianzas y el ANOVA:

```
car::leveneTest(datos$Biomarcador~datos$Grupo)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  1.3057 0.2741
147
aov(datos$Biomarcador~datos$Grupo)|>
  summary()
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
datos\$Grupo	2	9376	4688	101.5	<2e-16 ***
Residuals	147	6792	46		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ahora procedemos a realizar la prueba de Dunnett:

```
# Instale la librería si es necesario
DescTools::DunnettTest(x=datos$Biomarcador, g=datos$Grupo)
Registered S3 method overwritten by 'DescTools':
  method          from
reorder.factor  gplots
```

Dunnett's test for comparing several treatments with a control :
95% family-wise confidence level

```
$Control
              diff      lwr.ci   upr.ci    pval
Tratamiento 1-Control 16.14808    13.11154   19.18462   2.2e-16 ***
Tratamiento 2-Control 17.33120    14.29466   20.36774   < 2e-16 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note cómo la comparación la hace con el primer nivel del factor:

```
levels(datos$Grupo)
[1] "Control"      "Tratamiento 1" "Tratamiento 2"
```

Del resultado anterior podemos ver que hay diferencias tanto en el tratamiento 1 como en el tratamiento 2.

Ventajas y limitaciones del test de Dunnet

- ♦ **Ventajas:** es más poderosa estadísticamente que aplicar pruebas *t* individuales para cada comparación cuando se tiene un grupo de control.
- ♦ **Limitaciones:** al igual que otros métodos de comparación múltiple, requiere que los datos cumplan ciertos supuestos, como la homogeneidad de varianzas.

Supuestos para el test de Dunnet

- ♦ Normalidad
- ♦ Igualdad de varianzas
- ♦ Independencia de los datos

Prueba de Scheffe

La prueba de Scheffe es un método *post hoc* utilizado en el ANOVA. Este método es especialmente útil para comparaciones no planificadas. La ventaja de la prueba de Scheffe es que corrige el nivel de significancia (α) tanto para comparaciones de medias simples como para comparaciones más complejas que involucran más de un par de medias simultáneamente.

Comparación con otros métodos

A diferencia de las pruebas LSD de Fisher (prueba que no se revisa aquí) y Tukey, la prueba de Scheffe es más flexible, pero tiene **menor poder estadístico**. Aquí hay una guía general para elegir entre Tukey y Scheffe:

- ♦ Si solo necesita realizar comparaciones por pares, opte por el método de Tukey, ya que proporciona intervalos de confianza más estrechos.
- ♦ Si tiene interés en comparar todas las combinaciones posibles de medias, tanto simples como complejas, utilice la prueba de Scheffe.

Cálculo de diferencias mínimas

Para aplicar la prueba de Scheffe, es necesario calcular las diferencias mínimas significativas entre los pares de grupos y compararlas con las diferencias absolutas observadas.

Por ejemplo:

- ♦ Media del Grupo A: 36
- ♦ Media del Grupo B: 34.5
- ♦ Media del Grupo C: 35.6
- ♦ Media del Grupo D: 36.21

COMPARACIONES	VALOR ABSOLUTO	DIFERENCIAS MÍNIMAS	¿SIGNIFICATIVA?
AB	1.5	0.32	Sí
AC	0.4	0.35	Sí
AD	0.21	0.33	No
BC	1.1	0.34	Sí
BD	1.71	0.32	Sí
CD	0.61	0.35	Sí

Suposiciones prueba de Scheffe

Es relevante mencionar que, aunque la prueba de Scheffe no requiere estrictamente que las varianzas sean homogéneas o que la distribución sea normal, es recomendable tener el mismo número de casos en cada grupo para que los resultados sean más robustos.

Ventajas y desventajas de la prueba de Scheffe

ASPECTO	DESCRIPCIÓN
Ventajas	- Bueno para análisis de datos exploratorios y para probar teorías bien desarrolladas.
	- Puede probar pares que constan de combinaciones.
	- Prueba relativamente poderosa.
	- No es necesario definir contrastes de antemano.
	- Disponible en muchos paquetes estadísticos.
	- Riesgo reducido de errores de tipo II.
Desventajas	- Requiere grupos de igual tamaño.
	- Prueba contrastes que no son de interés.
	- Más sujeto a errores de tipo 1 que otras pruebas <i>post hoc</i> .
	- Intervalos de confianza amplios y poco poder estadístico.

Prueba de Scheffe en R

- ♦ **Ejemplo 18.7:** en R podemos hacer la prueba de Scheffe utilizando la función `scheffe.test` del paquete `agricolae`. Realizaremos una prueba *post hoc* de Scheffe de la base de datos `Cushings` del paquete `MASS`.

1. Instalar y cargar librería:

```
install.packages("agricolae")
library(agricolae)## Cargar librería
```

2. Crear un modelo de ANOVA:

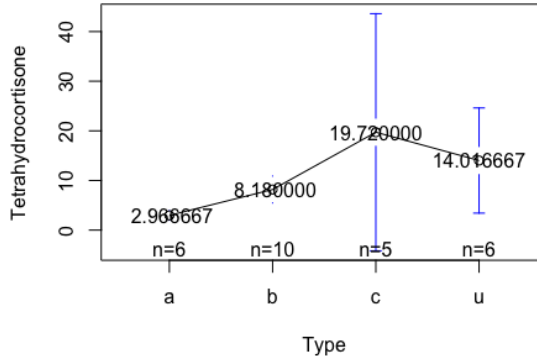
```
Anova_Cushings <- aov(Tetrahydrocortisone~Type, data=Cushings)
summary(Anova_Cushings)
```

	Df	Sum	Sq Mean	Sq	F value	Pr(>F)
Type	3	893.5	297.84		3.226	0.0412 *
Residuals	23	2123.6	92.33			

Signif. Codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3. Realizar un gráfico de medias para comprender los datos

```
library(gplots)
plotmeans(Tetrahydrocortisone~Type, data=Cushings, mean.labels = T)
Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped
Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped
Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped
Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped
```



4. Realizar la prueba *post hoc*. La función `scheffe.test` tiene los siguientes argumentos:

- `y` = modelo de ANOVA
- `trt` = variable de clasificación para el ANOVA (tratamientos)
- `main` = para especificar el título del modelo
- `group` = permite la comparación por grupos (TRUE o FALSE)
- `console` = para arrojar los resultados por consola

Es necesarios crear un objeto

```
comparaciones <- scheffe.test(y=Anova_Chusings, trt="Type", group = T, console = T)
```

Study: Anova_Chusings ~ "Type"

Scheffe Test for Tetrahydrocortisone

Mean Square Error : 92.33242

Type, means

	Tetrahydrocortisone	std	r	se	Min	Max	Q25	Q50	Q75
a	2.96667	0.9244818	6	3.922848	1.9	4.1	2.175	3.05	3.625
b	8.180000	3.7891072	10	3.038625	3.8	15.4	5.900	7.75	8.900
c	19.72000	19.2388149	5	4.297265	9.2	53.8	9.600	10.20	15.800
u	14.01667	10.0958242	6	3.922848	2.6	30.0	7.050	12.95	18.625

Alpha: 0.05 ; DF Error: 23

Critical Value of F: 3.027998

Groups according to probability of means differences and alpha level(0.05)

Means with the same letter are not significantly different.

	Tetrahydrocortisone	groups
c	19.720000	a
u	14.016667	a
b	8.180000	a
a	2.966667	a

comparaciones

\$statistics

MSerror	Df	F	Mean	CV
92.33242	23	3.027998	10.45556	91.90307

\$parameters

test	name	t	ntr	alpha
Scheffe	Type	4		0.05

\$means

Tetrahydrocortisone	std	r	se	Min	Max	Q25	Q50	Q75
a 2.966667	0.9244818	6	3.922848	1.9	4.1	2.175	3.05	3.625
b 8.180000	3.7891072	10	3.038625	3.8	15.4	5.900	7.75	8.900
c 19.720000	19.2388149	5	4.297265	9.2	53.8	9.600	10.20	15.800
u 14.016667	10.0958242	6	3.922848	2.6	30.0	7.050	12.95	18.625

\$comparison

NULL

\$groups

```

Tetrahydrocortisone    groups
c          19.720000    a
u          14.016667    a
b           8.180000    a
a           2.966667    a

attr("class")
[1] "group"

```

5. Interpretación

El test de Scheffe arroja entre otras cosas estadística descriptiva por grupos, grados de libertad e indica si las comparaciones son significativas o no sin dar un valor de p en específico. La siguiente tabla ejemplifica de una manera didáctica los resultados obtenidos para el ejemplo anterior:

GRUPO (TRATAMIENTO)	MEDIA	GRUPO COMPARACIÓN	SIGNIFICANCIA
c	19.72	a	Misma letra (no significancia)
u	14.01	a	Misma letra (no significancia)
b	8.18	a	Misma letra (no significancia)
a	2.97	a	Misma letra (no significancia)

Al tener la misma letra en el grupo de comparación todas las medias son iguales, la función asigna conforme al abecedario la letra en este grupo de comparación. En el ejemplo anterior, todas las comparaciones fueron iguales las letras, en este caso, la letra “a” son iguales.

Otro ejemplo: en este caso se utilizará la base de datos *sweetpotato* contenida en el paquete *agricolae*. En esta base hay cuatro grupos (cc, fc, ff, oo) que se encuentran en la variable virus:

```

virus  yield
1  cc    28.5
2  cc    21.7
3  cc    23.0

```

4	fc	14.9
5	fc	10.6
6	fc	13.1
7	ff	41.8
8	ff	39.2
9	ff	28.0
10	oo	38.2
11	oo	40.4
12	oo	32.1

- ◇ **Ejemplo 18.8:** el siguiente código se utilizó para realizar un ANOVA de la variable *yield* agrupado por la variable *virus*:

```
data(sweetpotato)
model<-aov(yield~virus, data=sweetpotato)
```

El siguiente código se utilizó para realizar la prueba *post hoc* de Scheffe del modelo de ANOVA anterior:

```
comparison <- scheffe.test(model,"virus", group=TRUE,console=TRUE,
                             main="Yield of sweetpotato\nDealt with different virus")
```

```
Study: Yield of sweetpotato
Dealt with different virus
```

```
Scheffe Test for yield
```

```
Mean Square Error   : 22.48917
```

```
virus, means
```

	yield	std	r	se	Min	Max	Q25	Q50	Q75
cc	24.40000	3.609709	3	2.737953	21.7	28.5	22.35	23.0	25.75
fc	12.86667	2.159475	3	2.737953	10.6	14.9	11.85	13.1	14.00

```
ff  36.33333  7.333030  3  2.737953  28.0  41.8  33.60  39.2  40.50
oo  36.90000  4.300000  3  2.737953  32.1  40.4  35.15  38.2  39.30
```

Alpha: 0.05 ; DF Error: 8

Critical Value of F: 4.066181

Minimum Significant Difference: 13.52368

Means with the same letter are not significantly different.

	yield	groups
oo	36.90000	a
ff	36.33333	a
cc	24.40000	ab
fc	2.86667	b

La siguiente tabla ayuda con la interpretación:

GRUPO	MEIDA DEL GRUPO	COMPARACIÓN	INTERPRETACIÓN	INTERPRETACIÓN
o o	36.9	a	El grupo oo es igual al ff y al cc. Todos tienen a	El grupo oo es distinto del fc (uno tiene a y el otro b)
f f	36.3	a	El grupo ff es igual al oo y al cc. Todos tienen a	El grupo ff es distinto del fc (uno tiene a y el otro b)
c c	24.4	ab	El grupo cc es igual a todos los grupos tiene a y b	
f c	12.87	b	El grupo fc es distinto al oo y ff (no tienen b)	

Test de Dunn

La prueba de Dunn es una **prueba *post hoc* no paramétrica** utilizada para identificar diferencias significativas entre las medianas de distintos grupos. Aunque sigue un principio similar al de la corrección de Bonferroni, es específica para análisis no paramétricos.

En esta prueba *post hoc* las hipótesis se establecen de la siguiente forma:

- ♦ **Hipótesis nula:** no hay diferencias significativas entre las medianas de los grupos.
- ♦ **Hipótesis alternativa:** existen diferencias significativas entre las medianas de al menos dos grupos.

Consideraciones para emplear el test de Dunn

La prueba de Dunn es especialmente útil cuando:

- ♦ Se desea comparar un subconjunto específico de todos los pares posibles de grupos.
- ♦ No se está haciendo una comparación con un grupo control.
- ♦ Se han utilizado pruebas no paramétricas, como Kruskal-Wallis para el análisis inicial.

Si se está comparando con un grupo control, la prueba de Dunnett o T3 de Dunnet sería más apropiada.

Test de Dunn en R

- ♦ **Ejemplo 18.9:** supongamos que es de nuestro interés la efectividad de 3 tratamientos para reducir el dolor medido en una escala de dolor de 0 a 100. Sabemos por estudios previos que los datos no se ajustan a una distribución normal y que debemos realizar una prueba *post hoc* para conocer en qué grupos están la diferencias.

1. Creación de *Dataframe* ficticio

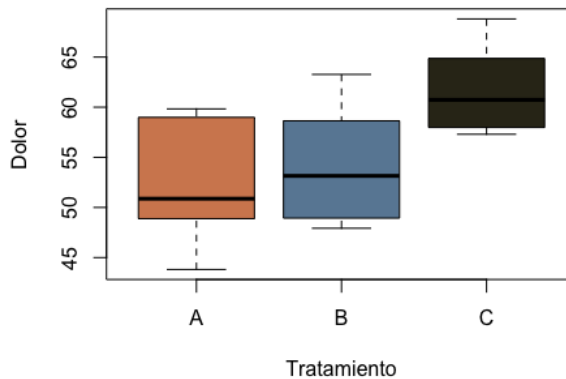
```
#Creación de un Dataframe
datos <- data.frame(Tratamiento = rep(c("A", "B", "C"), each = 10),
                    Dolor = c(runif(10, 40, 60),
                              runif(10, 45, 65),
                              runif(10, 55, 70)))
```

2. Visualizar el *Dataframe*:

```
#Visualizar los primeros 6 datos
head(datos)

  Tratamiento  Dolor
1      A      47.05997
2      A      59.61917
3      A      50.77766
4      A      48.88068
5      A      58.98734
6      A      49.04967

boxplot(Dolor~Tratamiento, data=datos,
        col=c("#D3885E", "#6988A3", "#33301D",
              main="Escala del dolor en tres distintos tratamientos"))
```



3. Realizar Kruskal-Wallis:

```
kruskal.test(Dolor ~ Tratamiento, data = datos)
```

Kruskal-Wallis rank sum test

data: Dolor by Tratamiento

Kruskal-Wallis chi-squared = 10.97, df = 2, p-value = 0.004148

4. Prueba *post hoc*:

```
library(dunn.test)
dunn.test(datos$Dolor,
          datos$Tratamiento, kw=T)
Kruskal-Wallis rank sum test
```

data: x and group

Kruskal-Wallis chi-squared = 10.9703, df = 2, p-value = 0

Comparison of x by group
(No adjustment)

Col Mean		
Row Mean	A	B
B	-0.609600	0.2711
C	-3.124203	-2.514602
	0.0009*	0.0060*

alpha = 0.05

Reject Ho if $p \leq \alpha/2$

Es necesario cambiar el parámetro kw

para indicar que la comparación viene de un ANOVA

En los resultados anteriores podemos observar que las diferencias se encuentran entre la comparación del grupo A y C () y entre B y C ().

- ♦ **Ejemplo 18.10:** también se puede realizar utilizando la librería `rstatix` aunque el método de ajuste del valor de p es distinto al de la función `dunn.test`:

```
rstatix::dunn_test(data = datos, formula = Dolor~Tratamiento)
# A tibble: 3 × 9
```

.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.signif
* <chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<chr>
1	Dolor A	B	10	10	0.610	0.542	0.542	ns
2	Dolor A	C	10	10	3.12	0.00178	0.00535	**
3	Dolor B	C	10	10	2.51	0.0119	0.0238	*

Ventajas

- ♦ **No paramétrica:** útil cuando los datos no se distribuyen normalmente o cuando las varianzas son diferentes.
- ♦ **Comparaciones por pares:** permite comparar cada par de grupos para determinar cuáles son significativamente diferentes entre sí.
- ♦ **Menos restrictiva:** menos supuestos en comparación con pruebas paramétricas como el ANOVA.
- ♦ **Adecuada para muestras pequeñas:** puede ser más fiable que otras pruebas *post hoc* cuando el tamaño de la muestra es pequeño.

Desventajas

- ♦ **Menor poder estadístico:** al ser una prueba no paramétrica, puede tener menos poder para detectar diferencias reales en comparación con pruebas paramétricas.
- ♦ **Solamente para dos grupos:** la prueba de Dunn originalmente fue diseñada para comparar solo dos grupos a la vez, aunque existen extensiones para más grupos (pruebas complejas).
- ♦ **Riesgo de error Tipo I:** al igual que otras pruebas *post hoc*, incrementa el riesgo de encontrar diferencias significativas por azar (error de Tipo I) debido a las comparaciones múltiples.

Dunnett T3

La prueba T3 de Dunnett es una extensión del test de Dunnett estándar y se utiliza para realizar comparaciones múltiples entre grupos cuando las varianzas no son homogéneas. Esta prueba de comparación a priori o múltiple utilizada

para determinar, si la media de una condición de control difiere de la de dos o más condiciones experimentales en un análisis de varianza. Se puede utilizar para **tamaños de muestra iguales y desiguales**. Se podría decir que es el equivalente **no paramétrico de la prueba de Dunnett**. Esta prueba tiene en cuenta la mayor probabilidad de cometer un error de Tipo I cuantas más comparaciones se realicen. Algunos autores recomiendan su uso para $n < 50$.

Dunnett T3 en R

♦ Ejemplo 18.11

1. Instalación del paquete:

```
install.packages("PMCMRplus") # Si es necesario instale la librería
```

2. Cargar la librería:

```
library(PMCMRplus)
# Crear datos de ejemplo con varianzas heterogéneas
groupA ← rnorm(10, mean = 50, sd = 10)
groupB ← rnorm(10, mean = 55, sd = 15)
groupC ← rnorm(10, mean = 53, sd = 20)
groupControl ← rnorm(10, mean = 52, sd = 12)

# Combinar los datos en un data frame
data ← data.frame(
  value = c(groupA, groupB, groupC, groupControl),
  group = factor(rep(c("A", "B", "C", "Control"), each = 10))
)

# Realizar ANOVA
model ← aov(value ~ group, data = data)

# Realizar prueba T3 de Dunnett
dunnettT3Test(model, var.equal=F)
```

```

Pairwise comparisons using Dunnett's T3 test for multiple com-
parisons

      with unequal variances
data: value by group
      A B C
B      1 - -
C      1 1 -
Control 1 1 1

P value adjustment method: single-step
alternative hypothesis: two.sided

```

Puede consultar más información de la función `dunnettT3Test` en: <https://rdr.io/cran/PMCMRplus/man/dunnettT3Test.html>

Ventajas Dunnett T3

- ♦ Control de error Tipo I: al igual que con la prueba de Dunnett estándar, el riesgo de errores de Tipo I se controla de manera efectiva.
- ♦ Varianzas heterogéneas: puede ser aplicada cuando las varianzas entre los grupos no son iguales.
- ♦ Comparación con un grupo control: es útil cuando se tiene un grupo de control específico al que se desean comparar otros grupos.

Desventajas Dunnett T3

- ♦ Complejidad computacional: puede ser más compleja de calcular que la prueba de Dunnett estándar.
- ♦ Poder estadístico: podría tener menos poder estadístico que otros métodos cuando las varianzas son en realidad homogéneas.
- ♦ Disponibilidad en software: no todos los paquetes estadísticos incluyen esta prueba específica.

Games-Howell

Prueba que se utiliza para determinar cuál de tres o más medias difieren entre sí cuando la relación F en un análisis de varianza es significativa. Fue desarrollado para tratar con **grupos con varianzas desiguales. Se puede utilizar con grupos de igual o diferente tamaño.** Se basa en la estadística de rango estudentizado.

La prueba de Games-Howell es una prueba *post hoc* no paramétrica que se utiliza para comparaciones múltiples después de un análisis de varianza (ANOVA) cuando los supuestos de **homogeneidad de varianzas no se cumplen.** Esta prueba calcula intervalos de confianza para cada par de medias y compara si se solapan o no para determinar la significancia estadística.

Games-Howell implementación en R

- ◇ **Ejemplo 18.12:** una forma de estimar la prueba *post hoc* de Games-Howell en R es utilizando la función `posthocTGH` del paquete `rosetta`. Para realizar la comparación *post hoc* entre los distintos tipos de Cushings y las concentraciones de tetrahydrocortisona utilice el siguiente código:

1. Instalar librería:

```
install.packages("rosetta")
```

2. Cargar librería:

```
library(rosetta)
```

3. Realizar prueba *post hoc*:

```
posthocTGH(Cushings$Tetrahydrocortisone, Cushings$Type)
```

	n	means	variances
a	6	3.0	0.85
b	10	8.2	14.36
c	5	19.7	370.13
u	6	14.0	101.93

Registered S3 methods overwritten by 'ufs':

	method	from				
	pander.crossTab	rosetta				
	pander.freq	rosetta				
	print.crossTab	rosetta				
	print.freq	rosetta				
	diff	ci.lo	ci.hi	t	df	p
b-a	5.2	1.4	9	4.1	10.7	.01
c-a	16.8	-18.2	52	1.9	4.0	.34
u-a	11.1	-4.1	26	2.7	5.1	.14
c-b	11.5	-23.2	46	1.3	4.2	.59
u-b	5.8	-9.1	21	1.4	5.9	.56
u-c	-5.7	-39.1	28	0.6	5.8	.93

La función `posthocTGH` requiere que las variables se encuentren separadas por una coma, primero se pone la variable de prueba cuantitativa y luego la variable de agrupación.

Holms

La corrección de Holm es un método de ajuste de la tasa de error de Tipo I en pruebas estadísticas múltiples. Se utiliza para controlar el error de Tipo I global en conjuntos de pruebas, ajustando el nivel de significancia de cada prueba individual. Holm es una extensión del procedimiento de Bonferroni y es menos conservador que este último.

El procedimiento de Holm es un método **secuencial** diseñado para incrementar el poder estadístico de un conjunto de pruebas mientras controla el error de Tipo I a nivel familiar (familywise Type I error). El primer paso en el procedimiento de Holm consiste en realizar todas las pruebas individuales para obtener sus respectivos valores p . Estos se ordenan en orden ascendente, desde el valor p más bajo hasta el más alto. La prueba con el valor p más bajo se somete primero a una corrección de Bonferroni. Si esta prueba no resulta ser estadísticamente significativa, el procedimiento se detiene y se concluye que

ninguna de las pruebas es significativa. De lo contrario, el proceso continúa, ajustando los niveles de significancia para las pruebas restantes. Este proceso se repite tantas veces como sea necesario.

La principal diferencia entre Bonferroni y Holms, es que en Bonferroni compara todas las combinaciones posibles; Holms solo lo hace para las comparaciones que son significativas.

Holms en R

Existen múltiples formas de utilizar este procedimiento en R, de hecho, es la prueba *post hoc* que se realiza por *default* en la mayoría las funciones. Para utilizar esta prueba *post hoc* podemos:

- ♦ Utilizar la función `pairwise.t.test()` o `pairwise.table()` cambiando el argumento `p.adjust.method` por “holm”
- ♦ Utilizar la función `t_test` de la librería `rstatix`
- ♦ Utilizar la función `ggbetweenstats` de la librería `ggstatplot`

Ventajas y desventajas de Holms

ASPECTO	DESCRIPCIÓN
Ventajas	
Menos conservador	Mientras que la corrección de Bonferroni puede ser muy conservadora, la corrección de Holm ofrece un equilibrio.
Control del error	Ayuda a controlar el error de Tipo I en pruebas múltiples.
Facilidad de uso	Es fácil de calcular y entender, y está ampliamente disponible en software estadístico.
Desventajas	
Errores Tipo II	Aumenta la probabilidad de errores de Tipo II (no rechazar una hipótesis nula falsa).
Poder estadístico	Hay métodos más modernos y menos conservadores disponibles, aunque Holm es más poderoso que Bonferroni.

Hommel

El procedimiento de Hommel es un método adaptativo y flexible que ofrece un equilibrio entre el control del error de Tipo I y el mantenimiento de la potencia estadística. Es más potente que los métodos de Bonferroni y Holm. El procedimiento de Hommel ajusta cada valor p según todas las demás comparaciones y utiliza una estrategia de búsqueda para encontrar el mayor valor p ajustado que todavía es menor que el nivel de significancia.

Ventajas

- ♦ Mayor potencia: comparado con Bonferroni y Holm, Hommel ofrece una mayor potencia estadística, lo que significa una menor probabilidad de errores de Tipo II.
- ♦ Control del error de Tipo I: similar a Bonferroni y Holm, ayuda a controlar el error de Tipo I en pruebas múltiples.

Desventajas

- ♦ Complejidad computacional: el método de Hommel puede ser más complejo de calcular que otros métodos como Bonferroni o Holm, especialmente para un gran número de pruebas.
- ♦ No es óptimo para todas las situaciones: hay métodos más modernos y menos conservadores disponibles.

Implementación de la prueba Hommel en R

- ♦ Utilizar la función `pairwise.t.test()` o `pairwise.table()` cambiando el argumento `p.adjust.method` por “hommel”
- ♦ Utilizar la función `t_test` de la librería `rstastix`
- ♦ Utilizando la función `ggbetweenstats` de la librería `ggstatplot`

Benjamini-Hochberg

El procedimiento de Benjamini-Hochberg, también conocido como corrección FDR (False Discovery Rate), es un enfoque diseñado para controlar la tasa de

descubrimientos falsos en múltiples pruebas de hipótesis. A diferencia de otros métodos como Bonferroni o Holm, que buscan controlar la tasa de error de Tipo I en toda la familia de pruebas, Benjamini-Hochberg se centra en controlar la proporción de errores Tipo I entre las hipótesis rechazadas.

¿Cómo ejecutar el procedimiento de Benjamini-Hochberg

1. Ordene los valores p individuales en orden ascendente.
2. Asigne rangos a los valores p . Por ejemplo, el más pequeño tiene un rango de 1, el segundo más pequeño un rango de 2. Calcule el valor crítico de Benjamini-Hochberg para cada valor p individual utilizando la fórmula $(i/m)Q$ donde:
 - » i = rango del valor p individual
 - » m = número total de pruebas
 - » Q = tasa de falsos descubrimientos (un porcentaje, elegido por el investigador)
3. Compare sus valores p originales con el valor crítico B-H calculado en el paso anterior. Identifique el valor p más grande que sea menor que el valor crítico.

Por ejemplo, si se realizan 8 pruebas y se elije un valor de Q de 25% solamente la comparación en el grupo A sería significativa:

VARIABLE	VALOR DE P	RANGO	$(i/m)Q$
Grupo A	0.001	1	0.03125
Grupo B	0.008	2	0.0625
Grupo C	0.039	3	0.09375
Grupo D	0.041	4	0.125
Grupo E	0.042	5	0.15625
Grupo F	0.06	6	0.1875
Grupo G	0.074	7	0.21875
Grupo H	0.205	8	0.25

El valor que se elige para q , que es la tasa de falsos descubrimientos permitidos, depende del contexto de la investigación y del nivel de rigor que se desee. No hay un valor “habitual” para q que sea universalmente aceptado en todas las disciplinas, pero un valor comúnmente utilizado es 0.05, similar al nivel de significancia α que se usa en pruebas estadísticas clásicas.

Ventajas

- ♦ **Control de FDR:** este método controla la Tasa de Descubrimiento Falso, lo que es a menudo más deseable en contextos donde se realizan muchas pruebas simultáneas.
- ♦ **Mayor potencia estadística:** Comparado con métodos más conservadores como Bonferroni, ofrece más poder estadístico, reduciendo la tasa de errores de Tipo II.
- ♦ **Aplicabilidad:** es aplicable en una amplia gama de disciplinas y tipos de datos.

Desventajas

- ♦ **No es siempre riguroso:** en ciertos escenarios puede ser menos conservador en términos de control del error Tipo I comparado con otros métodos.
- ♦ **Dependencia estructural:** su rendimiento puede verse afectado si hay una estructura de dependencia compleja entre las pruebas.

Implementación de la prueba Benjamini-Hochberg en R

- ♦ Utilizar la función `pairwise.t.test()` o `pairwise.table()` cambiando el argumento `p.adjust.method` por “BH”
- ♦ Utilizar la función `t_test` de la librería `rstatix`
- ♦ Utilizando la función `ggbetweenstats` de la librería `ggstatplot`

Nemenyi Test

La prueba de Nemenyi es la misma que la prueba de Tukey HSD, excepto que probamos la diferencia entre rangos de los grupos. La corrección Nemenyi estima un valor de q para la distribución de Tukey:

$$q = \frac{|R_i^- - R_j^-|}{s.e.}$$

Donde:

$$s.e. = \sqrt{\frac{k(n+1)}{12}}$$

Donde k = el número de grupos y n = el tamaño total de la muestra, donde los tamaños son todos iguales. El estadístico q tiene una distribución q de rango studentizado y se compara con el valor de tablas para obtener los valores de p (igual que Tukey).

Nemenyi implementación en R

- ♦ **Ejemplo 18.13:** se puede utilizar la función `NemenyiTest` del paquete `DescTools`. Siguiendo con el ejemplo de la base de datos `Chushings` tendríamos lo siguiente:

1. Instalar librería:

```
install.packages("DescTools")
```

2. Cargar librería:

```
library(DescTools)
```

3. Realizar KW:

```
kruskal.test(Cushings$Tetrahydrocortisone, Cushings$Type)
```

Kruskal-Wallis rank sum test

data: Cushings\$Tetrahydrocortisone and Cushings\$Type

Kruskal-Wallis chi-squared = 13.509, df = 3, p-value = 0.003655

4. Realizar la prueba de Nemenyi:

```
NemenyiTest(Cushings$Tetrahydrocortisone, Cushings$Type)
```

Nemenyi's test of multiple comparisons for independent samples (tukey)

	mean.rank.diff	pval
b-a	9.466667	0.0958 .
c-a	16.416667	0.0036 **
u-a	12.916667	0.0249 *
c-b	6.950000	0.3793
u-b	3.450000	0.8345
u-c	-3.500000	0.8858

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Comentarios finales

- ♦ Las comparaciones múltiples generalmente se realizan solo después de que una prueba F revela significación. En cualquier caso, se necesita ANOVA para calcular MSE.
- ♦ Muchos paquetes estadísticos realizan la prueba de Tukey u otros procedimientos de comparaciones múltiples, y calculan el valor p para cada comparación, que se puede utilizar para decidir qué grupos son significativamente diferentes de los demás.
- ♦ Es posible que la prueba F sea significativa, pero ninguna de las comparaciones por pares es significativa. Por el contrario, la prueba F puede no mostrar significación, pero la comparación para un par específico aún puede ser significativa. Esto sucede porque ambos requieren un patrón gaussiano, pero los datos prácticos rara vez siguen un patrón gaussiano exacto. Las pruebas F y Tukey se comportan de manera diferente para la desviación del patrón gaussiano. El problema puede surgir con mayor frecuencia para n pequeña.

- ♦ Si se presenta el fenómeno descrito en el punto anterior, se recomienda utilizar otra prueba *post hoc*.
- ♦ Los procedimientos anteriores son válidos solo para comparaciones planificadas previamente. A veces, la idea de probar la significación estadística entre dos o más grupos surge después de ver los datos observados (esto a veces se denomina espionaje de datos).

Ejercicios

En este capítulo no se muestran las resoluciones de los ejercicios, sin embargo, puede resolverlos utilizando los ejemplos aquí descritos.

- ♦ **Ejercicio 18.1:** utilice el *Dataframe* Cushings de la librería MASS y realice una prueba *post hoc* para Pregnanetriol por tipo (Type) de Cushings. Resuelva este ejercicio utilizando: a. `pairs.t.test` b. `t_student` de librería `rstatix` c. Realice los gráficos pertinentes d. Utilice la librería `ggstatsplot`.
- ♦ **Ejercicio 18.2:** realice una prueba *post hoc* de Tukey para Pregnanetriol por tipo (Type) de Cushings. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.3:** realice una prueba *post hoc* de Dunnet para Pregnanetriol por tipo (Type) de Cushings. Considere al Cushing tipo “a” como grupo de referencia. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.4:** asuma que se cumplen todos los supuestos para realizar la prueba de Scheffe y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Interprete sus resultados. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.5:** asuma que se cumplen todos los supuestos para realizar la prueba de Dunn y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Interprete sus resultados. interprete sus resultados. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.6:** asuma que se cumplen todos los supuestos para realizar la prueba de T3 de Dunnet y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Tome como grupo control al grupo de Cushing “a”. Interprete sus resultados. Utilice el *Dataframe* Cushings de la librería MASS.

- ♦ **Ejercicio 18.7:** asuma que se cumplen todos los supuestos para realizar la prueba de Games-Howell y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Interprete sus resultado. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.8:** asuma que se cumplen todos los supuestos para realizar la prueba de Holms y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Realice la prueba por los tres métodos descritos en el compendio. Interprete sus resultado. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.9:** asuma que se cumplen todos los supuestos para realizar la prueba de Hommel y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Realice la prueba por los tres métodos descritos en el compendio. Utilice el *Dataframe* Cushings de la librería MASS.
- ♦ **Ejercicio 18.10:** asuma que se cumplen todos los supuestos para realizar la prueba de BH y realice esta comparación *post hoc* para Pregnanetriol por tipo (Type). Realice la prueba por los tres métodos descritos en el compendio. Utilice el *Dataframe* Cushings de la librería MASS.

Ejercicio 18.11: la función `t_test` de la librería `rstastix` contiene las siguientes pruebas *post hoc*: “holm”, “hochberg”, “hommel”, “bonferroni”, “BH”, “BY” y “fdr”. Realice una pequeña investigación de cada una de las pruebas que incluye:

- » Fundamento
- » Ventajas
- » Desventajas
- » Si necesita varianzas iguales
- » Si necesita una distribución normal
- » Si necesita grupos con n igual

Referencias

- Antonello, J. M., Clark, R. L. y Heyse, J. F. (1993). Application of the tukey trend test procedure to assess developmental and reproductive toxicity. I. Measurement data. *Fundamental and applied toxicology: official journal of the Society of Toxicology*, 21(1), 52-58. <https://doi.org/10.1006/faat.1993.1071>
- Armstrong, R. A. (2014). When to use the Bonferroni correction. *Ophthalmic & physiological optics: the journal of the British College of Ophthalmic Opticians (Optometrists)*, 34(5), 502-508. <https://doi.org/10.1111/opo.12131>
- Atil, H. y Unver, Y. (2001). Multiple comparisons. *Journal of Biological Sciences*, 1(8), 723-727. <https://doi.org/10.3923/jbs.2001.723.727>
- Benjamini, Y. y Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1), 289-300. <http://www.jstor.org/stable/2346101>
- Binus, S., Koch, A., Petschelt, A. y Berthold, C. (2013). Restoration of endodontically treated teeth with major hard tissue loss—bond strength of conventionally and adhesively luted fiber-reinforced composite posts. *Dental traumatology: official publication of International Association for Dental Traumatology*, 29(5), 339-354. <https://doi.org/10.1111/edt.12013>
- Brechenmacher, T., Xu, J., Dmitrienko, A. y Tamhane, A. C. (2011). A mixture gatekeeping procedure based on the Hommel test for clinical trial applications. *Journal of biopharmaceutical statistics*, 21(4), 748-767. <https://doi.org/10.1080/10543406.2011.551334>
- Curran-Everett, D. y Milgrom, H. (2013). post hoc data analysis: benefits and limitations. *Current opinion in allergy and clinical immunology*, 13(3), 223-224. <https://doi.org/10.1097/ACI.0b013e3283609831>
- De Mendiburu, F. (2023). *Agricolae: Statistical Procedures for Agricultural Research*. Rpackage version 1.3-7. <https://CRAN.R-project.org/package=agricolae>
- Dinno, A. (2015). Nonparametric pairwise multiple comparisons in independent groups using Dunn's test. *The Stata Journal*, 15(1), 292-300. <https://doi.org/10.1177/1536867X1501500117>
- Eichstaedt, K. E., Kovatch, K. y Maroof, D. A. (2013). A less conservative method to adjust for familywise error rate in neuropsychological research: The Holm's sequential Bonferroni procedure. *NeuroRehabilitation*, 32(3), 693-696. <https://doi.org/10.3233/NRE-130893>

- Elliott, H. L. (1996). post hoc analysis: use and dangers in perspective. *Journal of hypertension. Supplement: official journal of the International Society of Hypertension*, 14(2), S21-S25. <https://doi.org/10.1097/00004872-199609002-00006>
- Games, P. A. y Howell, J. F. (1976). Pairwise multiple comparison procedures with unequal N's and/or variances: A Monte Carlo study. *Journal of Educational Statistics*, 1(2), 113-125. <https://doi.org/10.3102/10769986001002113>
- Green, G. y Diggle, P. (2007). On the Operational Characteristics of the Benjamini and Hochberg False Discovery Rate Procedure. *Statistical Applications in Genetics and Molecular Biology*, 6(1). <https://doi.org/10.2202/1544-6115.1302>
- Haynes, W. (2013). Benjamini–Hochberg method. En W. Dubitzky, O. Wolkenhauer, K. H. Cho y H. Yokota (Eds.), *Encyclopedia of Systems Biology*. Springer. https://doi.org/10.1007/978-1-4419-9863-7_1215
- Holm, K. y Christman, N. J. (1985). post hoc tests following analysis of variance. *Research in nursing & health*, 8(2), 207-210. <https://doi.org/10.1002/nur.4770080215>
- Indrayan, A. y Malhotra, R. K. (2017). Inference from means. En *Medical Biostatistics* (4ª ed., pp. 377-418). Chapman & Hall/CRC Biostatistics Series.
- Kassambara, A. (2023). *rstatix: Pipe-Friendly Framework for Basic Statistical Tests*. R package version 0.7.2. <https://CRAN.R-project.org/package=rstatix>.
- Kim, H. Y. (2015). Statistical notes for clinical researchers: post hoc multiple comparisons. *Restorative dentistry & endodontics*, 40(2), 172-176. <https://doi.org/10.5395/rde.2015.40.2.172>
- Kucuk, U., Eyuboglu, M., Kucuk, H. O. y Degirmencioglu, G. (2016). Importance of using proper post hoc test with ANOVA. *International journal of cardiology*, 209, 346. <https://doi.org/10.1016/j.ijcard.2015.11.061>
- Lee, S. y Lee, D. K. (2018). What is the proper way to apply the multiple comparison test? *Korean Journal of Anesthesiology*, 71(5), 353-360. <https://doi.org/10.4097/kja.d.18.00242>
- Liu, Y. y Chen, W. (2012). A SAS macro for testing differences among three or more independent groups using Kruskal-Wallis and Nemenyi tests. *Journal of Huazhong University of Science and Technology [Medical Sciences]*, 32(1), 130-134. <https://doi.org/10.1007/s11596-012-0023-9>
- Mata, D. A. y Milner, D. A., Jr (2021). Statistical Methods in Experimental Pathology: A Review and Primer. *The American journal of pathology*, 191(5), 784-794. <https://doi.org/10.1016/j.ajpath.2021.02.009>

- McHugh, M. L. (2011). Multiple comparison analysis testing in ANOVA. *Biochemia medica*, 21(3), 203-209. <https://doi.org/10.11613/bm.2011.029>
- McHugh, M. L. (2011). Multiple comparison analysis testing in ANOVA. *Biochemia medica*, 21(3), 203-209. <https://doi.org/10.11613/bm.2011.029>
- Peters, G. y Verboon, P. (2023). rosetta: Parallel Use of Statistical Packages in Teaching. R package version 0.3.12. <https://CRAN.R-project.org/package=rosetta>
- Pohlert, T. (2023). PMCMRplus: Calculate Pairwise Multiple Comparisons of Mean Rank Sums Extended. R package version 1.9.10. <https://CRAN.R-project.org/package=PMCMRplus>
- Puga, T. B., Schafer, J., Thiel, G., Scigliano, N., Ruan, T., Toledo, A., Agbedanu, P. N. y Treffer, K. (2024). COVID-19 National Football League (NFL) Injury Analysis: Follow-Up Study. *JMIRx med*, 5, e45688. <https://doi.org/10.2196/45688>
- Ramsey, P. H. y Ramsey, P. P. (2009). Power and Type I errors for pairwise comparisons of means in the unequal variances case. *The British journal of mathematical and statistical psychology*, 62(Pt 2), 263-281. <https://doi.org/10.1348/000711008X291542>
- Sauder, D. C. y DeMars, C. E. (2019). An updated recommendation for multiple comparisons. *Advances in Methods and Practices in Psychological Science*, 2(1), 26-44. <https://doi.org/10.1177/2515245918808784>
- Sauder, D. y DeMars, C. E. (2018, September 19). Multiple comparisons. Open Science Framework. https://osf.io/6xa45?view_only=ec6b0641cad546bfbf2c3c13be9156f6
- Scheffé, H. (1953). A method for judging all contrasts in the analysis of variance. *Biometrika*, 40(1-2), 87-110. <https://doi.org/10.1093/biomet/40.1-2.87>
- Signorell, A. (2024). DescTools: Tools for Descriptive Statistics. R package version 0.99.54. <https://CRAN.R-project.org/package=DescTools>
- Srinivas, T. R., Ho, B., Kang, J. y Kaplan, B. (2015). post hoc analyses: after the facts. *Transplantation*, 99(1), 17-20. <https://doi.org/10.1097/TP.0000000000000581>
- Upton, G. y Cook, I. (2014). *A Dictionary of Statistics* (3rd ed.). Oxford University Press.
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics with S*. Fourth Edition. Springer.

Pruebas de hipótesis para la comparación de proporciones

Ana Gabriela Flores Vargas
Edsaúl Emilio Pérez Guerrero

Un grupo de investigadores desea conocer si saltarse el desayuno se asocia con la obesidad. Para ello, plantean un estudio en el que se incluyó dos grupos de individuos: uno integrado por individuos que si se saltaban el desayuno y otro grupo en el que los individuos desayunaban todos los días. Se identificó cuántas personas con sobrepeso había en cada uno de los grupos y se realizó la prueba estadística pertinente.

En este estudio podemos identificar que los investigadores están interesados en conocer la relación de dos variables cualitativas, la presencia o ausencia de ayuno y la presencia o ausencia de sobrepeso. Para este tipo de situaciones, se deben de emplear pruebas estadísticas que nos permiten comparar una variable cualitativa en dos o más grupos.

En este capítulo nos ocuparemos en resolver estas situaciones. Para ello, revisaremos primero la construcción de tablas de contingencia, posteriormente revisaremos la prueba de χ^2 y prueba exacta de Fisher, para finalizar con la revisión del test de McNemar.

Tablas de contingencia

En Estadística y en especial en Epidemiología, existe una herramienta que nos permite conocer resumir la relación entre dos variables categóricas. Estas herramientas se conocen como **tablas de contingencia** o tablas cruzadas. Estas

tablas permiten observar cómo se distribuyen las frecuencias de diferentes combinaciones de categorías de las variables involucradas. En la [tabla 19.1](#) se representa una tabla de contingencia clásica. En ella podemos ver que:

- ♦ Las filas representan las categorías de una variable.
- ♦ Las columnas representan las categorías de otra variable.
- ♦ Las celdas contienen las frecuencias absolutas (número de observaciones) de cada combinación de categorías de las variables.

Se acostumbra a que las casillas se nombren de la “a” a la “d”. Es importante que respete esta nomenclatura para la estimación de medidas epidemiológicas ([Capítulo 20](#)). Aunque el ejemplo más simple involucra dos variables binarias (tablas utilizadas en epidemiología), las tablas de contingencia se pueden expandir para incluir variables con múltiples categorías.

TABLA 19.1. TABLA DE CONTINGENCIA

	OBESIDAD	NO OBESIDAD	TOTAL
Ayuno	a	b	a + b
Desayuno	c	d	c + d
Total	a + c	b + d	a + b + c + d

Usted ya ha utilizado las funciones de R que nos permiten construir tablas de contingencia `table` y `prop.table` (ver “[Estadística descriptiva para variables cualitativas](#)”). Sin embargo, vamos a recordarlas realizando un ejemplo.

- ♦ **Ejemplo 19.1:** imagine que está interesado en construir una tabla de contingencia para identificar en cuántos de los recién nacidos sus madres fumaban. Utilice la base de datos `birthwt` de la librería `MASS` para ello:

```
# Importar base de datos
library(MASS)
data("birthwt")
# Corregir factores
birthwt$smoke ← factor(birthwt$smoke,
  labels = c("No fumó",
    "Fumó"))
```

```
birthwt$low <- factor(birthwt$low,
                      labels = c("Normo peso",
                                "Bajo peso"))

# Construcción de tabla
table(birthwt$smoke,
      birthwt$low)
```

	Normo peso	Bajo peso
No fumó	86	29
Fumó	44	30

Si es de su interés tener proporciones en lugar de frecuencias, puede utilizar la función `prop.table()`, que, si recuerda, es una función que se alimenta de tablas:

```
# Construcción de tabla
table(birthwt$smoke,
      birthwt$low)|>
prop.table(2)|>
# Utilice 2 para proporciones por columnas
round(2)
```

	Normo peso	Bajo peso
No fumó	0.66	0.49
Fumó	0.34	0.51

Las tablas de contingencia son especialmente útiles para la realización de la prueba de X^2 y para representar los datos de dos variables cualitativas.

Prueba de X^2

La prueba de X^2 es una técnica estadística que se utiliza para la comparación de variables cualitativas en dos o más grupos. Es un método no paramétrico basado en la distribución X^2 que mide la independencia de dos métodos de clasificación.

Esta prueba permite analizar datos categóricos, por ejemplo:

- ♦ Número de éxitos que ha tenido una intervención.
- ♦ Porcentaje de pacientes que presentan una característica.
- ♦ Proporción de resultados favorables obtenidos de grupo de pacientes con tratamientos distintos.

Para obtener el estadístico de prueba se emplea la siguiente fórmula:

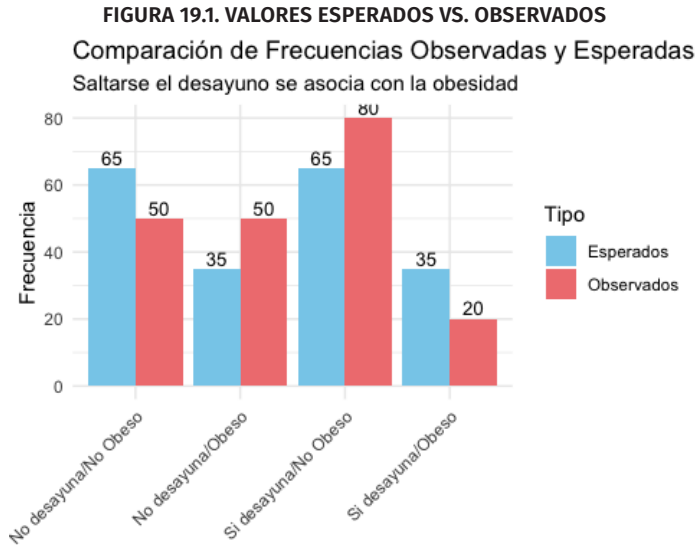
$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^r \left[\frac{(O_{ij} - E_{ij})^2}{E_{ij}} \right]$$

Donde: O_{ij} es la frecuencia observada y E_{ij} es la frecuencia esperada para la celda .

En la fórmula anterior hay dos aspectos especialmente útiles para comprender el fundamento de la prueba X^2 , los valores o frecuencias esperadas y observadas. Las **frecuencias observadas** son el número de individuos en la muestra que caen dentro de las diversas categorías de la variable de interés. Mientras que las **frecuencias esperadas** son el número de individuos en la muestra que se esperarías observar si alguna hipótesis nula respecto a la variable es verdadera. Entre más grande sea la diferencia entre los valores esperados y los observados menos probabilidad hay que los resultados sean poco compatibles con la H_0 .

La [figura 19.1](#) muestra un gráfico de barras con los valores observados y esperados hipotéticos para el ejemplo del inicio del capítulo. Como podrá observar, en algunas categorías, como la de “No desayuna/Obeso”, las frecuencias observadas y esperadas son bastante diferentes, mientras que en otras, como “Sí desayuna/No Obeso”, son muy cercanas. La notable diferencia entre las frecuencias observadas y esperadas para “No desayuna/Obeso” sugiere una asociación significativa entre saltarse el desayuno y la obesidad, ya que el valor de X^2 es más grande.

- ◇ Si las diferencias entre los valores observados y esperados son pequeñas, sugiere que los datos son compatibles con H_0 .
- ◇ Si las diferencias son grandes es menos probable que sean compatibles con la H_0 .



Una forma muy resumida de realizar manualmente una prueba de X^2 es siguiendo los pasos mostrados enseguida:

1. Tomar muestras aleatorias de las poblaciones de interés de grupos independientes.
2. Realizar tablas de contingencia.
3. Calcular las frecuencias esperadas para cada celda.
4. Comparar las frecuencias esperadas y observadas.
5. Comparar el valor calculado de X^2 (mi moneda de cambio) con el valor tabulado de X^2 (la moneda de cambio de la distribución).
6. Determinar el valor de p .

Vamos realizar paso a paso la prueba de X^2 .

En el estudio sobre saltarse el desayuno y la obesidad, los investigadores encontraron las descritas en la [tabla 19.2](#). El objetivo de este estudio fue evaluar si el ayuno se asocia con la obesidad. Las hipótesis para este estudio son:

- ♦ H_0 : la obesidad y el ayuno son independientes (no existe relación entre las variables).
- ♦ H_A : la obesidad y el ayuno son dependientes (existe relación entre las variables).

Al ser la prueba de X^2 una prueba que mide la independencia de dos métodos de clasificación, las hipótesis deben de plantearse en este sentido (dependencia y no dependencia) y no en el sentido de porcentajes y frecuencias.

TABLA 19.2. FRECUENCIAS SOBRE EL ESTUDIO DEL AYUNO Y LA OBESIDAD

	OBESIDAD	NO OBESIDAD	TOTAL
Ayuno	16 (a)	134 (b)	150
Desayuno	6 (c)	119 (d)	125
Total	22	253	275

Ahora, el siguiente paso sería estimar los valores esperados. Para ello, utilizamos la siguiente fórmula:

$$E_{ij} = \frac{(n_i)(n_j)}{n}$$

Donde n_i representa el total de la columna de la celda de interés y n_j representa el total de la fila de interés. Por ejemplo, para la celda “a”, el valor de n_i es 22 y el valor de n_j es 150. En esta fórmula n es el total de la muestra. La [figura 19.2](#) muestra cómo seleccionar los valores de n_j y n_i para la celda “a”. Note cómo la celda “a” es donde se cruzan los valores de los recuadros marcados en azul y naranja.

FIGURA 19.2. OBTENCIÓN DE VALORS DE n_i Y n_j

Tabla 20.2: Frecuencias sobre el estudio del ayuno y la obesidad

	Obesidad	No obesidad	Total
Ayuno	16 (a)	134 (b)	150 (n_j)
Desayuno	6 (c)	119 (d)	125 (n_j)
Total	22 (n_i)	253 (n_i)	275 (n)

Sustituyendo en la fórmula para la casilla los valores serían:

$$a = \frac{(22)(150)}{275} 12$$

Y para el resto de las celdas:

$$b = \frac{(253)(150)}{275} 138$$

$$c = \frac{(22)(125)}{275} 10$$

$$d = \frac{(253)(125)}{275} 115$$

Con el cálculo de los valores esperados, ahora debemos de estimar el valor del estadístico de X^2 (mi moneda de cambio). Sustituyendo en la fórmula del estadístico obtenemos lo siguiente:

$$X^2 = \frac{(16-12)^2}{12} + \frac{(134-138)^2}{138} + \frac{(6-10)^2}{10} + \frac{(119-115)^2}{115} = 3.2$$

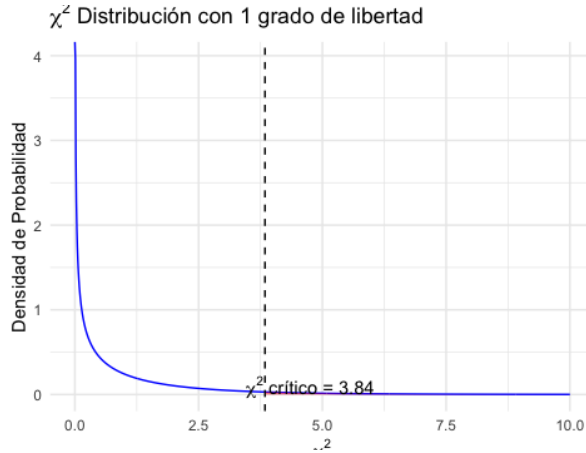
El siguiente paso es estimar el valor de referencia de la distribución (la moneda de cambio de la distribución), para ello empleamos la función `qchisq()`:

```
qchisq(p = 0.05, df=1, lower.tail = F)
[1] 3.841459
```

Como revisamos en “[Distribución chi-cuadrada \(\$X^2\$ \)](#)”, la distribución X^2 en realidad es una familia de distribuciones que dependen de los grados de libertad. Para calcular los grados de libertad se utiliza la fórmula: $gl = (r - 1)(c - 1)$, donde gl son los grados de libertad, r es el número de renglones o filas y c es el número de columnas. Para una tabla de dos por dos los grados de libertad son 1, por ello se seleccionó $df=1$.

En la distribución X^2 la región de rechazo se encuentra a la derecha de la distribución, en este caso a la derecha del 3.84 (ver [figura 19.3](#)). Recordemos que esta distribución son valores de z elevados al cuadrado.

FIGURA 19.3. ZONA DE RECHAZO PARA UNA DISTRIBUCIÓN χ^2



Con estos resultados, no podemos rechazar la H_0 porque 3.12 es menor a 3.84. Finalmente, el valor de p para estos datos sería:

```
pchisq(q = 3.12, df = 1, lower.tail = F)
[1] 0.07733678
```

Vamos a resolver este ejercicio utilizando R.

- ♦ **Ejemplo 19.2:** primero vamos a crear un *Dataframe* con los datos del estudio del ayuno y la obesidad. Puede copiar y pegar el código:

```
# Crear un Dataframe con los datos observados de la tabla
ayuno_obesidad <- data.frame(
  Ayuno = rep(c("Ayuno", "Desayuno"), # Nombres de las columnas
             times = c(150, 125)), # Frecuencias de las columnas
  Obesidad = c(rep("Obeso", 16), # Frecuencias de las filas
               rep("No Obeso", 134), # Frecuencias de las filas
               rep("Obeso", 6), # Frecuencias de las filas
               rep("No Obeso", 119))) # Frecuencias de las filas
```

Ahora vamos a utilizar la función `chisq.test()` para realizar nuestra prueba de hipótesis:

```
table(ayuno_obesidad$Ayuno, ayuno_obesidad$Obesidad)|>
  chisq.test(correct = F)
```

Pearson's Chi-squared test

```
data: table(ayuno_obesidad$Ayuno, ayuno_obesidad$Obesidad)
X-squared = 3.1884, df = 1, p-value = 0.07416
```

La función también permite obtener resultados de la siguiente forma:

```
chisq.test(ayuno_obesidad$Ayuno,
           ayuno_obesidad$Obesidad,
           correct = F)
```

Pearson's Chi-squared test

```
data: ayuno_obesidad$Ayuno and ayuno_obesidad$Obesidad
X-squared = 3.1884, df = 1, p-value = 0.07416
```

En el código anterior, uno de los argumentos con el que se alimentó la función `chisq.test()` fue `correct = F`. Este argumento permite indicarle a la función que no realice la corrección por continuidad. Dedicaremos una sección a explicar este punto (ver “[Correcciones por continuidad en la distribución \$X^2\$](#) ”).

En R para poder obtener los valores esperados y observados debemos de guardar los resultados de la prueba de X^2 en un objeto y luego acceder a ese objeto utilizando el signo `$`.

♦ Ejemplo 19.3:

```
chisq.test(ayuno_obesidad$Ayuno,
           ayuno_obesidad$Obesidad,
           correct = F)→x2
# Para obtener los valores esperados
x2$expected
```


ayuno_obesidad\$Obesidad		
ayuno_obesidad\$Ayuno	No Obeso	Obeso
Ayuno	138	12
Desayuno	115	10

Correcciones por continuidad en la distribución X^2

Como lo explicamos en secciones anteriores, la prueba de X^2 permite evaluar datos que, en el mejor de los casos, son discretos (frecuencias observadas) y compararlos con datos de una distribución continua (frecuencias esperadas). Esta discrepancia puede introducir un sesgo, especialmente cuando los valores esperados son pequeños. Cuando las frecuencias esperadas en una tabla de contingencia son pequeñas (generalmente se considera un umbral de 5, aunque esto no es una ley), la distribución X^2 puede no ser una buena aproximación. En palabras más sencillas, se podría decir que estamos comparando peras con manzanas, pero cuanto menos información tengamos de la forma de las manzanas, más nos estaremos equivocando y es probable que cometamos un error por continuidad, si es que este no se corrige.

La corrección más utilizada para tratar de minimizar el sesgo por falta de continuidad es la corrección de Yates. La corrección de Yates se aplica restando 0.5 a la diferencia entre las frecuencias observadas y esperadas antes de elevar al cuadrado y dividir por la frecuencia esperada. La fórmula corregida para el estadístico X^2 con corrección de Yates es:

$$X^2_{\text{corregido}} = \sum \frac{(|O_{ij} - E_{ij}| - 0.5)^2}{E_{ij}}$$

Corregir o no corregir el estadístico X^2 puede ser un dilema, se sugiere que se realice una corrección de Yates cuando se utilizan tablas de contingencia de 2×2 y cuando se esperan valores pequeños. Sin embargo, a menos que su muestra sea lo suficientemente grande, le recomendamos siempre hacer la corrección de Yates ya que reduce considerablemente el error tipo I.

En R, cuando utilice la función `chisq.test()` cambie a `correct = T` o simplemente omita el argumento.

Además de la corrección existen otras correcciones:

- ♦ Corrección de Williams.
- ♦ Corrección de Haldane-Anscombe.
- ♦ Corrección de Efron.
- ♦ Método exacto de Fisher.

Aquí solo nos enfocaremos en el último método.

Prueba exacta de Fisher

Este método se nombró así en honor a sir Ronald Fisher. La prueba exacta de Fisher es un procedimiento estadístico de prueba de hipótesis que permite evaluar la asociación entre dos variables categóricas (al igual que X^2). Este método es especialmente útil en tamaño de muestra pequeños, por lo que es la opción a utilizar cuando los valores esperados en la prueba de X^2 son pequeños.

En general el test de Fisher es muy robusto, sin embargo, necesita cumplir los siguientes supuestos para ser utilizado:

- ♦ Variables categóricas
- ♦ Observaciones independientes
- ♦ Datos mutuamente excluyentes (en relación al anterior)

A diferencia de otras pruebas estadísticas el resultado de la prueba exacta de Fisher es el valor de p :

$$\frac{(a+c)!(a+b)!(b+d)!(c+d)!}{n!a!b!c!d!}$$

En R utilice la función `fisher.test()` para su aplicación. Esta función se alimenta de tablas o de dos variables categóricas o factores:

```
table(ayuno_obesidad$Ayuno, ayuno_obesidad$Obesidad)>
fisher.test()
```

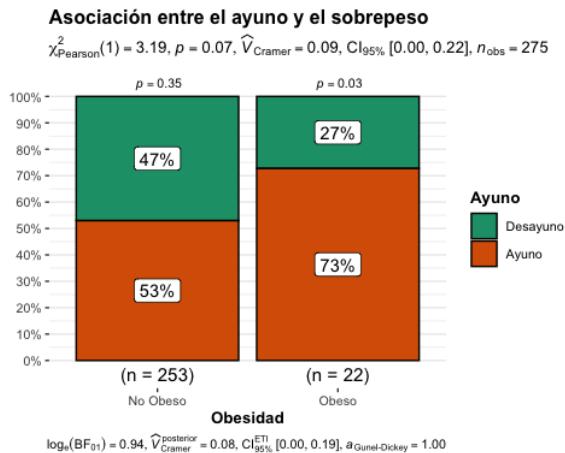
Fisher's Exact Test for Count Data

```
data: table(ayuno_obesidad$Ayuno, ayuno_obesidad$Obesidad)
p-value = 0.07939
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.1313403 1.1856429
sample estimates:
odds ratio
 0.4234786
```

χ^2 en ggstatsplot

Utilizando los datos del [ejemplo 19.2](#) vamos a utilizar ggstatsplot para realizar la prueba de χ^2 :

```
ayuno_obesidad|>
  ggstatsplot::ggbarstats( x=Ayuno,
                           y=Obesidad,
                           type = "parametric",
                           title = "Asociación entre el ayuno y el sobrepeso",
                           subtitle = "Utilizando ggstatplot")
```



La gráfica anterior es sumamente informativa. Entre otras cosas, podemos ver el valor de X^2 y su respectivo valor de p . Además, se muestran los porcentajes de cada una de las categorías. Note cómo los porcentajes están estimados por columnas (obeso y no obeso). Esto es lo correcto, ya que cuanto más diferentes sean las barras naranjas en las columnas (obeso y no obeso), mayor será el valor de X^2 , ya que los valores esperados serán muy diferentes de los observados.

Además, en esta gráfica se muestra el valor de la V de Cramer, la cual es una medida para el tamaño del efecto. El tamaño del efecto de V de Cramer se calcula a partir del valor de X^2 y ajusta este valor para el tamaño de la tabla y el tamaño de la muestra, proporcionando una medida de asociación que varía entre 0 y 1. Entre más cercano a 1 más fuerte será el tamaño del efecto.

Test de McNemar

Hasta ahora, hemos revisado pruebas que nos permiten relacionar dos variables cualitativas en grupos independientes. Pero ¿qué sucede si mis datos provienen de grupos pareados? De entrada no podemos utilizar la prueba de X^2 , ya que se basa en evaluar la dependencia de dos variables y nuestros grupos son dependientes. Nuestra opción entonces es el test de McNemar.

Para poder emplear esta prueba se debe de cumplir lo siguiente:

- ♦ Los participantes deben estar emparejados o emparejados/relacionados
 - ♦ Variables cualitativas
 - ♦ Categorías mutuamente excluyentes
- ♦ **Ejemplo 19.4:** supongamos que estamos interesados en evaluar la efectividad de una nueva terapia para dejar de fumar. Tenemos un grupo de 100 pacientes que participaron en el estudio. Cada paciente se evalúa antes y después de la terapia para ver si dejaron de fumar.

La H_0 para este estudio es:

- ♦ La proporción de individuos que dejó de fumar es similar antes y después de la terapia.

Los datos de este estudio se organizaron en una tabla de 2X2:

	DESPUÉS: DEJÓ DE FUMAR	DESPUÉS: NO DEJÓ DE FUMAR	TOTAL
Antes: fumaba	30	20	50
Antes: no fumaba	10	40	50
Total	40	60	100

Además, estos datos también fueron capturados en un *Dataframe*. Utilice el siguiente código para generar base de datos:

```
fumar_habitos <- data.frame(
  Antes = rep(c("Fumaba", "No Fumaba"), times = c(50, 50)),
  Despues = c(rep("Dejó de Fumar", 30),
               rep("No Dejó de Fumar", 20),
               rep("Dejó de Fumar", 10),
               rep("No Dejó de Fumar", 40))
)
```

Ahora vamos a utilizar la función `mcnemar.test()` para obtener los resultados directamente desde R:

```
table(fumar_habitos$Antes, fumar_habitos$Despues)|>
  mcnemar.test()

McNemar's Chi-squared test with continuity correction

data: table(fumar_habitos$Antes, fumar_habitos$Despues)
McNemar's chi-squared = 2.7, df = 1, p-value = 0.1003
```

Con estos datos no es posible rechazar la H_0 , por lo tanto, podemos concluir que la proporción de individuos antes y después de la terapia es similar. Estos resultados muestran, además, las monedas de cambio utilizadas en la prueba y los grados de libertad para estimarlas.

Ejercicios

En este capítulo no se describen las resoluciones, pero puede utilizar los ejemplos de este capítulo y anteriores para resolverlos.

- ♦ **Ejercicio 19.1:** un grupo de investigadores desea conocer si el fumar en el primer trimestre del embarazo se asocia al bajo peso de los recién nacidos. Estos datos fueron capturados en la base de datos “`birthwt`”. Utilice α de 0.05.
 - a) Construya un gráfico de barras con el paquete base de R y `ggplot2`
 - b) Reporte los valores esperados por casilla
 - c) Utilice también `ggstatplot`
 - d) Concluya
 - e) ¿Es necesario utilizar la corrección de Yates?

- ♦ **Ejercicio 19.2:** un grupo de investigadores desea conocer si la raza es independiente al bajo peso al nacer. Utilice la base `birthwt`, genere una tabla de contingencia y un gráfico de barras con proporciones. Además, compruebe su resultado utilizando la librería `ggstatsplot`.
- ♦ **Ejercicio 19.3:** ¿el bajo peso al nacer se asocia a la hipertensión previa al embarazo? Utilice la base `birthwt`, genere una tabla de contingencia y un gráfico de barras con proporciones. Además, compruebe su resultado utilizando la librería `ggstatsplot`.
- ♦ **Ejercicio 19.4:** ¿el antecedente de tabaquismo es independiente de la raza?. Utilice la base `birthwt`, genere una tabla de contingencia y un gráfico de barras con proporciones. Además, compruebe su resultado utilizando la librería `ggstatsplot`.
- ♦ **Ejercicio 19.5:** resuelva las preguntas anteriores, esta vez utilizando la prueba exacta de Fisher.
- ♦ **Ejercicio 19.6:** utilizando la base `Melanoma` de la librería MASS contiene los datos 205 pacientes originarios de Dinamarca que fueron diagnosticados con melanoma entre los años 1962 a 1977. Si quiere conocer el significado de cada una de las variables, después de cargar la librería MASS utilice el siguiente comando `help(“Melanoma”)`. Esta base de datos se utilizó anteriormente en el [capítulo 11](#).

- a) Realice la recodificación de las variables cualitativas a factores. Realice este paso solo si es necesario.
 - b) ¿La edad será un factor que condicione el desenlace en los pacientes? Realice un corte de la variable edad en los 50 años.
 - c) ¿Los hombres tendrán una mayor mortalidad comparados con las mujeres?
 - d) ¿La presencia o ausencia de úlceras será un factor que influye en los desenlaces de los pacientes?
 - e) Repita los puntos del b al d, pero esta vez compare a los pacientes que murieron por melanoma vs. el resto de los pacientes (agrupe alive y dead from other causes).
 - f) Repita la resolución de estos ejercicios utilizando la librería `ggstatsplot`.
- ♦ **Ejercicio 19.7:** suponga que tiene como tarea evaluar la efectividad de un programa de ejercicio en la reducción de la hipertensión. Estos datos fueron capturados en un *Dataframe* que podrá replicar con el siguiente código:

```
# Crear el data frame con los datos observados de la tabla
hipertension <- data.frame(
  Antes = rep(c("Hipertenso",
               "No Hipertenso"),
             times = c(50, 50)),
  Despues = c(rep("Hipertenso", 40),
              rep("No Hipertenso", 10),
              rep("Hipertenso", 5),
              rep("No Hipertenso", 45))
)
```

¿Podría concluir que el programa fue efectivo?

Referencias

- Amiri, S. y Modarres, R. (2017). Comparison of tests of contingency tables. *Journal of biopharmaceutical statistics*, 27(5), 784-796. <https://doi.org/10.1080/10543406.2016.1269786>
- Brace, J. C. y Savalei, V. (2017). Type I error rates and power of several versions of scaled chi-square difference tests in investigations of measurement invariance. *Psychological methods*, 22(3), 467-485. <https://doi.org/10.1037/met0000097>
- Dawson, P. (1991). The Yates' correction factor in chi-squared analyses. *The British journal of psychiatry: the journal of mental science*, 159, 292. <https://doi.org/10.1192/bjp.159.2.292a>
- De Muth, J. E. (2009). Overview of biostatistics used in clinical research. *American journal of health-system pharmacy: AJHP: official journal of the American Society of Health-System Pharmacists*, 66(1), 70-81. <https://doi.org/10.2146/ajhp070006>
- Fagerland, M. W., Lydersen, S. y Laake, P. (2013). The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC medical research methodology*, 13, 91. <https://doi.org/10.1186/1471-2288-13-91>
- Faizi, N., & Alvi, Y. (2023). Categorical variables. En N. Faizi y Y. Alvi (Eds.), *Bio-statistics Manual for Health Research* (pp. 127-148). Academic Press. <https://doi.org/10.1016/B978-0-443-18550-2.00001-3>
- Feuer, E. J. y Kessler, L. G. (1989). Test statistic and sample size for a two-sample McNemar test. *Biometrics*, 45(2), 629-636.
- Foldnes, N. y Olsson, U. H. (2015). Correcting Too Much or Too Little? The Performance of Three Chi-Square Corrections. *Multivariate behavioral research*, 50(5), 533-543. <https://doi.org/10.1080/00273171.2015.1036964>
- Heumann, C., Schomaker, M. y Shalabh (2016). Hypothesis Testing. En *Introduction to Statistics and Data Analysis With Exercises, Solutions and Applications in R* (pp. 209-247). Springer.
- Hirji, K. F., Tan, S. y Elashoff, R. M. (1991). A quasi--exact test for comparing two binomial proportions. *Statistics in Medicine*, 10(7), 1137-1153. <https://doi.org/10.1002/sim.4780100713>
- Indrayan, A. y Malhotra, R. K. (2017). *Inference from proportions* (4ª ed., pp. 307-342). Chapman & Hall/CRC Biostatistics Series.

- Jung, S. H., Kang, S. H. y Ahn, C. (2003). Chi-square test for R x C contingency tables with clustered data. *Journal of biopharmaceutical statistics*, 13(2), 241-251. <https://doi.org/10.1081/BIP-120019269>
- Kim, S. y Lee, W. (2017). Does McNemar's test compare the sensitivities and specificities of two diagnostic tests? *Statistical methods in medical research*, 26(1), 142-154. <https://doi.org/10.1177/0962280214541852>
- Lee, S. W. (2022). Methods for testing statistical differences between groups in medical research: statistical standard and guideline of life cycle committee. *Life Cycle*, 2. <https://doi.org/10.54724/lc.2022.e1>
- Ma, L. y Mao, J. (2018). Fisher exact scanning for dependency. *Journal of the American Statistical Association*, 114(525), 245-258. <https://doi.org/10.1080/01621459.2017.1397522>
- MacFarland, T. W. y Yates, J. M. (2021). Correlation, association, regression, likelihood, and prediction. En *Using R for Biostatistics* (pp. 427-584). Springer.
- McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica*, 23(2), 143-149. <https://doi.org/10.11613/bm.2013.018>
- Moscarelli, M. (2023). Hypothesis testing. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 113-135).
- Paek, I. (2010). Conservativeness in rejection of the null hypothesis when using the continuity correction in the mh chi-square test in dif applications. *Applied Psychological Measurement*, 34(7), 539-548. <https://doi.org/10.1177/0146621610378288>
- Sahai, H. y Khurshid, A. (1995). On analysis of epidemiological data involving a 2 x 2 contingency table: an overview of Fisher's exact test and Yates' correction for continuity. *Journal of biopharmaceutical statistics*, 5(1), 43-70. <https://doi.org/10.1080/10543409508835098>
- Satorra, A. y Bentler, P. M. (2000). A scaled difference chi-square test statistic for moment structure analysis. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.199064>
- Sauro, J. y Lewis, J. R. (2012). Is there a statistical difference between designs? En J. Sauro y J. R. Lewis (Eds.), *Quantifying the User Experience* (pp. 63-103). Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-384968-7.00005-9>
- Serra, N., Rea, T., Di Carlo, P. y Sergi, C. (2019). Continuity correction of Pearson's chi-square test in 2x2 Contingency Tables: A mini-review on recent development. *10.2427/13059*.
- Serra, N. (2018). A significant minimization of Pearson's chi square statistics in 2x2 contingency tables: preliminary results for small samples. *15*. 1-5. *10.2427/12949*.

- Shen, C., Panda, S. y Vogelstein, J. T. (2022). The Chi-Square Test of Distance Correlation. *Journal of computational and graphical statistics: a joint publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America*, 31(1), 254-262. <https://doi.org/10.1080/10618600.2021.1938585>
- Wu, Y. (2022). Weighted McNemar's test for the comparison of two screening tests in the presence of verification bias. *Statistics in medicine*, 41(16), 3149-3163. <https://doi.org/10.1002/sim.9409>

Medidas epidemiológicas

Daniel Osmar Suárez Rico

La estimación de factores de riesgo, incidencias y prevalencias es algo inherente a ciencias de la salud. Si bien, éstas se aplican más en la Salud Pública, aquí haremos una revisión breve de ellas con objetivo de que el lector tenga conocimientos básicos de estas.

Para este capítulo necesitaremos cargar las siguientes librerías:

```
library(survival) # Para análisis de supervivencia  
library(ggplot2) # Para gráficos  
library(survminer) # Para gráficos de supervivencia  
library(epiR) # Para medidas epidemiológicas
```

Definiciones clave

Las medidas epidemiológicas son herramientas estadísticas utilizadas para describir la frecuencia y la asociación de enfermedades y otros eventos de salud en poblaciones específicas. Estas medidas se utilizan en Ciencias de la Salud para evaluar la magnitud de los problemas de salud, identificar factores de riesgo y tomar decisiones basadas en evidencia para la prevención y el control de enfermedades. Las principales medidas epidemiológicas que se considerarán en este capítulo son:

1. Medidas de asociación:

- » **Riesgo relativo (RR):** razón de la incidencia de una enfermedad entre los expuestos a un factor de riesgo y la incidencia entre los no expuestos.
- » **Odds ratio (OR):** razón de las probabilidades de ocurrencia del evento entre los expuestos y no expuestos, utilizada comúnmente en estudios de casos y controles.
- » **Hazard ratio (HR):** medida de la tasa de ocurrencia de un evento en un grupo expuesto en comparación a un grupo no expuesto a lo largo del tiempo.

2. Medidas de frecuencia:

- » **Prevalencia:** proporción de individuos en una población que presentan un proceso clínico en un momento determinado.
- » **Incidencia:** frecuencia de aparición de nuevos casos de una enfermedad en una población durante un periodo específico. Puede expresarse como incidencia acumulada o tasa de incidencia.

3. Medidas de impacto:

- » **Riesgo atribuible:** diferencia en la tasa de incidencia de la enfermedad entre los expuestos y no expuestos, que puede atribuirse a la exposición.

Las medidas epidemiológicas nos permiten a los investigadores:

- ♦ Cuantificar la carga de enfermedades en poblaciones específicas.
- ♦ Evaluar la eficacia de intervenciones y programas de salud.
- ♦ Identificar y cuantificar factores de riesgo asociados con enfermedades.
- ♦ Informar políticas y estrategias de salud pública para mejorar la salud de las poblaciones.

Medidas de frecuencia

Prevalencia

La prevalencia (P) es la proporción de individuos en una población que presentan un proceso clínico en un momento determinado. Se puede calcular mediante el muestreo representativo de una población definida en un momento específico.

Existen dos tipos principales de prevalencia:

- ♦ **Prevalencia puntual:** indica la proporción de individuos en una población que tienen una enfermedad o condición en un momento específico en el tiempo y generalmente se expresa como porcentaje. Se calcula de la siguiente manera:

$$P = \left(\frac{\text{Número de casos en un momento específico}}{\text{Población total}} \right) \times 100$$

Por ejemplo, si una población de 1000 personas, 50 tienen una enfermedad particular en un día determinado, la prevalencia puntual sería:

$$P = \left(\frac{50}{1,000} \right) \times 100 = 5$$

Esta medida nos es útil para evaluar la carga de una enfermedad en una población en un momento dado y así se podrán planificar los recursos de salud necesarios para el tratamiento y prevención.

- ♦ **Prevalencia de periodo:** es una medida epidemiológica que refleja la proporción de individuos en una población que han tenido una enfermedad o condición en algún momento durante un periodo de tiempo específico, la prevalencia de periodo abarca un intervalo de tiempo. Se calcula de la siguiente manera:

$$P = \left(\frac{\text{Número de casos existentes en cualquier momento durante el periodo}}{\text{Población total durante el periodo}} \right) \times 100$$

Para ilustrarlo, supongamos que, en una población de 100 personas, 70 han tenido una enfermedad en algún momento durante el último año. La prevalencia de periodo para ese año sería:

$$P = \left(\frac{70}{1,000} \right) \times 100 = 7$$

Prevalencia en R

- ◇ **Ejemplo 20.1:** vamos a generar una serie de datos ficticios para estimar la prevalencia puntual:

```
# Instalación y carga de la librería epiR
install.packages("epiR") # Instale si es necesario
library(epiR)
library(epiR)

# Fijar la semilla para reproducibilidad
set.seed(123)

# Definir el tamaño de la muestra
n ← 1000

# Generar identificadores únicos para los individuos
id ← 1:n

# Generar edades aleatorias entre 20 y 80 años
age ← sample(20:80, n, replace = TRUE)

# Generar géneros aleatorios (Male/Female)
gender ← sample(c("Male", "Female"), n, replace = TRUE)

# Generar la presencia de enfermedad (0 = No, 1 = Sí) con una pre-
valencia del 10%
disease ← sample(c(0, 1), n, replace = TRUE, prob = c(0.9, 0.1))

# Definir la fecha de inicio y fin de la observación
start_date ← as.Date("2020-01-01")
end_date ← as.Date("2023-01-01")

# Generar tiempos aleatorios de riesgo (en días) hasta un máximo de
3 años
```

```
time_at_risk <- sample(0:1095, n, replace = TRUE)

# Calcular la fecha del primer caso (si aplica)
event_date <- start_date + time_at_risk
event_date[disease == 0] <- NA # Solo asignar fecha a los casos con
enfermedad

# Crear un Dataframe con los datos generados
data <- data.frame(id, age, gender, disease, start_date, event_date,
time_at_risk)
```

```
# Mostrar las primeras filas del Dataframe
```

```
head(data)
```

	id	age	gender	disease	start_date	event_date	time_at_risk
1	1	50	Male	0	2020-01-01	<NA>	880
2	2	34	Male	0	2020-01-01	<NA>	50
3	3	70	Female	0	2020-01-01	<NA>	67
4	4	33	Female	1	2020-01-01	2021-09-11	619
5	5	22	Female	1	2020-01-01	2022-10-30	1033
6	6	61	Female	0	2020-01-01	<NA>	959

Ahora podemos estimar la prevalencia:

```
# Calcular la prevalencia como el número de casos divididos por el
total de individuos
```

```
prevalence <- sum(data$disease) / n
```

```
# Mostrar la prevalencia
```

```
prevalence
```

```
[1] 0.106
```

Incidencia

La incidencia refleja la dinámica de ocurrencia de nuevos casos presentes en cualquier momento durante un periodo específico. Existen dos tipos principales de medidas de incidencia:

- ♦ **Incidencia acumulada (IA):** la incidencia acumulada es una medida epidemiológica que se utiliza para describir la frecuencia con la que ocurren nuevos casos de una enfermedad en una población específica durante un periodo de tiempo determinado.

La incidencia acumulada se calcula dividiendo el número de casos nuevos de la enfermedad que han ocurrido durante un periodo de estudio entre el número de personas que estaban en riesgo de desarrollar la enfermedad al inicio del periodo de estudio. La fórmula para utilizar sería:

$$IA = \left(\frac{\text{Número de casos nuevos}}{\text{Número de personas en riesgo al inicio del periodo}} \right) \times 100$$

Por ejemplo, suponiendo que estamos estudiando la incidencia acumulada de una enfermedad en una comunidad de 1000 personas durante solamente un año. Si al final del año se han registrado 50 nuevos casos de la enfermedad, la incidencia acumulada sería:

$$IA = \left(\frac{50}{1000} \right) \times 100 = 5$$

Esto significa que la incidencia acumulada de la enfermedad en esa comunidad durante ese año es del 5%. Esto es útil para comparar la frecuencia de nuevas enfermedades entre diferentes poblaciones o periodos, evaluar la efectividad de intervenciones preventivas y para identificar patrones y tendencias en la aparición de enfermedades.

- ♦ **Tasa de incidencia (TI):** la tasa de incidencia es otra medida epidemiológica clave que se utiliza para cuantificar la frecuencia con la que ocurren nuevos casos de una enfermedad en una población específica durante un periodo de tiempo determinado.

Calculándose como el número de casos nuevos de una enfermedad que ocurren en una población en riesgo durante un periodo específico, dividido por la suma del tiempo en que cada individuo estuvo en riesgo durante el periodo de estudio:

$$TI = \left(\frac{\text{Número de casos nuevos}}{\text{Suma del tiempo en riesgo de todos los individuos}} \right)$$

Por ejemplo, suponga que se está estudiando la incidencia de una enfermedad en una comunidad y se tienen los siguientes datos para un periodo de un año:

- ♦ 5 personas estuvieron en riesgo durante todo el año (5 personas-año)
- ♦ 3 personas estuvieron en riesgo durante 6 meses cada una (1.5 personas-año)
- ♦ 2 personas estuvieron en riesgo durante 3 meses cada una (0.5 personas-año)

La suma del tiempo en riesgo sería:

$$5 + 1.5 + 0.5 = 7 \text{ personas} - \text{año}$$

Si se registraron 10 casos nuevos de la enfermedad durante este año, la tasa de incidencia sería:

$$TI = \left(\frac{10}{8} \right) = 1.43 \text{ casos por persona} - \text{año}$$

La tasa de incidencia es útil para evaluar la rapidez con la que una enfermedad está ocurriendo en una población, comparar la frecuencia de nuevas enfermedades entre diferentes poblaciones o periodos, ajustar por el tiempo que cada individuo estuvo en riesgo, lo cual es especialmente útil en estudios donde las personas entran y salen del periodo de riesgo en diferentes momentos.

- ♦ **Ejemplo 20.2:** utilizando los datos generados en el [ejemplo 20.1](#) vamos a estimar la incidencia:

```
# Calcular la incidencia acumulada usando epiR
# La función epi.conf calcula intervalos de confianza para proporciones
```

```
incidence <- epi.conf(c(sum(data$disease),
                        n - sum(data$disease)), method = "exact")

# Mostrar la incidencia acumulada con su intervalo de confianza
incidence
  est se    lower    upper
1 500 394 -4506.245 5506.245
# Calcular la tasa de incidencia (casos por persona-año)
# Convertir el tiempo de riesgo total de días a años
time_at_risk_years <- sum(data$time_at_risk) / 365.25

# Calcular la tasa de incidencia como el número
# de casos dividido por el tiempo de riesgo en años
incidence_rate <- sum(data$disease) / time_at_risk_years

# Mostrar la tasa de incidencia
incidence_rate
[1] 0.06921645
```

Medidas de asociación

Riesgo relativo (RR)

El riesgo relativo (RR) es una medida de la probabilidad de que ocurra un evento (como una enfermedad) en un grupo expuesto en comparación con un grupo no expuesto. Se calcula como la razón entre la incidencia de la enfermedad en el grupo expuesto y la incidencia en el grupo no expuesto. Un RR mayor a 1 indica una asociación positiva entre la exposición y la enfermedad, mientras que un RR menor a 1 sugiere una asociación protectora:

$$RR = \frac{\text{Incidencia en el grupo expuesto}}{\text{Incidencia en el grupo no expuesto}}$$

El riesgo relativo se puede interpretar con relación a 1:

- ♦ **RR = 1:** no hay asociación entre la exposición y la enfermedad.
- ♦ **RR > 1:** la exposición está asociada con un aumento en el riesgo de enfermedad.
- ♦ **RR < 1:** la exposición está asociada con una disminución en el riesgo de enfermedad.

Para poder calcular el RR, necesitamos datos de un estudio de cohortes, donde se sigue a un grupo de individuos a lo largo del tiempo para observar la aparición de una enfermedad.

Supongamos un estudio en el que se evalúa la relación entre fumar (exposición) y desarrollar cáncer de pulmón (enfermedad).

	DESARROLLARON CÁNCER DE PULMÓN	NO DESARROLLARON CÁNCER DE PULMÓN	TOTAL
Fumadores	A	B	A+B
No fumadores	C	D	C+D

Donde:

- ♦ A = número de fumadores que desarrollaron cáncer de pulmón.
- ♦ B = número de fumadores que no desarrollaron cáncer de pulmón.
- ♦ C = número de no fumadores que desarrollaron cáncer de pulmón.
- ♦ D = número de no fumadores que no desarrollaron cáncer de pulmón.

La incidencia en el grupo expuesto (fumadores) es:

$$\text{Incidencia en expuestos} = \frac{A}{A+B}$$

La incidencia en el grupo no expuesto (no fumadores) es:

$$\text{Incidencia en no expuestos} = \frac{C}{C+D}$$

Por lo tanto, el RR se calcula como:

$$RR = \frac{\frac{A}{A+B}}{\frac{C}{C+D}} = \frac{A \cdot (C+D)}{C \cdot (A+B)}$$

Por ejemplo, suponga que está realizando un estudio sobre el efecto de una vacuna en la prevención de influenza. Los datos recolectados son los siguientes:

	DESARROLLARON INFLUENZA	NO DESARROLLARON INFLUENZA	TOTAL
Vacunados	20	1980	2 000
No vacunados	30	970	1 000

$$\text{Incidencia en vacunados} = \frac{20}{20+1980} = 0.01$$

$$\text{Incidencia en no vacunados} = \frac{30}{30+970} = 0.03$$

$$RR = \frac{0.01}{0.03} = 0.33$$

Un RR de 0.33 indica que las personas que recibieron la vacuna tienen 67% menos de riesgo de desarrollar la enfermedad en comparación con las que no recibieron la vacuna.

Intervalos de confianza para riesgos relativos

Los intervalos de confianza (IC) para los riesgos relativos (RR) nos funcionan para estimar la precisión del RR y para inferir si hay una asociación estadísticamente significativa entre la exposición y la enfermedad. A continuación, se explica cómo estimar los intervalos de confianza para los RR utilizando un enfoque basado en la aproximación normal.

- Calcular el riesgo relativo (RR):** primero, necesitamos calcular el RR utilizando los datos de incidencia en los grupos expuestos y no expuestos. Supongamos que tenemos los siguientes datos:

	ENFERMOS	NO ENFERMOS	TOTAL
Expuestos	a	b	a + b
No expuestos	c	d	c + d

El RR ya sabemos calcularlo como:

$$RR = \frac{\frac{a}{a+b}}{\frac{c}{c+d}} = \frac{a \cdot (c+d)}{c \cdot (a+b)}$$

2. **Calcular el error estándar del logaritmo del RR:** una vez teniendo el RR, calculamos el error estándar del logaritmo natural de RR ($\log(RR)$):

$$SE = (\log(RR)) = \sqrt{\frac{1}{a} + \frac{1}{c} + \frac{1}{a+b} - \frac{1}{c+b}}$$

3. **Calcular los límites del intervalo de confianza:** ya con el error estándar, lo utilizaremos para calcular los límites del intervalo de confianza (IC) para $\log(RR)$. Esto generalmente se usa con un nivel de confianza del 95%, lo que corresponde a un valor de Z de 1.96.

$$\text{Límite inferior del IC para } \log(RR) = \log(RR) - 1.96 \times SE(\log(RR))$$

$$\text{Límite superior del IC para } \log(RR) = \log(RR) + 1.96 \times SE(\log(RR))$$

4. **Transformar los límites del IC a escala original:** finalmente convertimos los límites del IC del $\log(RR)$ a la escala original del RR tomando el antilogaritmo (exponencial):

$$\text{Límite inferior del IC para } RR = e^{\log(RR) - 1.96 \times SE(\log(RR))}$$

$$\text{Límite superior del IC para } RR = e^{\log(RR) + 1.96 \times SE(\log(RR))}$$

Por ejemplo, suponga que en un estudio se está evaluando la eficacia de un nuevo medicamento para la hipertensión. Los datos recolectados son los siguientes:

	DESARROLLARON HIPERTENSIÓN	NO DESARROLLARON HIPERTENSIÓN	TOTAL
Vacunados	20	180	200
No vacunados	50	150	200

Calculamos RR:

$$RR = \frac{20 \times (50 + 150)}{50 \times (20 + 180)} = 0.4$$

Calculamos el error estándar del logaritmo de RR:

$$SE = (\log(RR)) = \sqrt{\frac{1}{20} + \frac{1}{50} + \frac{1}{200} - \frac{1}{200}} = 0.2449$$

Calculamos los límites del intervalo de confianza para $\log(RR)$:

$$\text{Límite inferior del IC para } \log(RR) = \log(0.4) - 1.96 \times 0.2449 = -1.3956$$

$$\text{Límite superior del IC para } \log(RR) = \log(0.4) + 1.96 \times 0.2449 = -0.4373$$

Transformamos los límites del IC a escala original:

$$\text{Límite inferior del IC para } RR = e^{-1.3956} = 0.247$$

$$\text{Límite superior del IC para } RR = e^{-0.4373} = 0.646$$

Estos resultados nos demuestran que con un RR de 0.40, significa que las personas que tomaron el medicamento tienen 60% menos de riesgo de desarrollar hipertensión en comparación con las personas que no tomaron el medicamento. El intervalo de confianza del 95% para el RR es de aproximadamente 0.25 a 0.65, lo que indica que estamos 95% seguros de que el verdadero RR se encuentra dentro de este rango.

Odds ratio (OR)

El odds ratio (OR) es otra medida de asociación ampliamente utilizada en estudios epidemiológicos, especialmente en estudios de casos y controles. Es una medida de la fuerza de la asociación entre una exposición y una enfermedad. Compara las probabilidades (odds) de exposición entre los casos (individuos con la enfermedad) y los controles (individuos sin la enfermedad):

$$OR = \left(\frac{\text{Odds de exposición en los casos}}{\text{Odds de exposición en los controles}} \right)$$

El odds ratio se puede interpretar con relación a 1:

- ♦ **OR = 1:** no hay asociación entre la exposición y la enfermedad.
- ♦ **OR > 1:** la exposición está asociada con un aumento en la probabilidad de enfermedad.
- ♦ **OR < 1:** la exposición está asociada con una disminución en la probabilidad de enfermedad.

Para poder calcular el OR, necesitamos datos de un estudio de casos y controles, donde los datos se organizan en una tabla de contingencia:

	EXPUESTOS	NO EXPUESTOS	TOTAL
Casos	A	B	A+B
Controles	C	D	C+D

Donde:

- ♦ A = número de casos expuestos.
- ♦ B = número de casos no expuestos.
- ♦ C = número de controles expuestos.
- ♦ D = número de controles no expuestos.

La odds de exposición en los casos es:

$$\text{Odds de exposición en casos} = \frac{A}{B}$$

La odds de exposición en los controles es:

$$\text{Odds de exposición en controles} = \frac{C}{D}$$

Por lo tanto, el OR se calcula como:

$$\text{OR} = \frac{\frac{A}{B}}{\frac{C}{D}} = \frac{A \times D}{C \times B}$$

Por ejemplo, en un estudio de casos y controles se evalúa la relación entre la exposición a un medicamento y la aparición de hipertensión, arrojando los siguientes datos:

	EXPUESTOS AL MEDICAMENTO	NO EXPUESTOS AL MEDICAMENTO	TOTAL
Casos	20	80	100
Controles	50	150	200

$$\text{Odds de exposición en casos} = \frac{20}{80} = 0.25$$

$$\text{Odds de exposición en controles} = \frac{50}{150} = 0.3333$$

$$\text{OR} = \frac{0.25}{0.3333} = 0.75$$

Un OR de 0.75 indica que los individuos expuestos al medicamento tienen 25% menos de probabilidades de desarrollar hipertensión en comparación con los no expuestos al medicamento. Esto sugiere que, en este estudio, la exposición al medicamento está asociada con una menor probabilidad de desarrollar hipertensión.

Intervalos de confianza para odds ratio

Los intervalos de confianza del odds ratio (OR) proporcionan una medida de la precisión de la estimación del OR y permiten evaluar la significancia estadística de la asociación entre una exposición y una enfermedad. Si el intervalo de confianza no incluye el valor 1, se puede concluir que existe una asociación estadísticamente significativa. Además, los intervalos de confianza ayudan a entender la variabilidad inherente en la estimación del OR, proporcionando un rango dentro del cual es probable que se encuentre el verdadero OR en la población. Esto es crucial para interpretar la solidez y la confiabilidad de los resultados.

Para estimar los intervalos de confianza (IC) para los OR, se utiliza una aproximación basada en la distribución normal del logaritmo del OR.

- 1. Calcular el OR:** primero, necesitamos calcular el OR utilizando los datos de un estudio de casos y controles. Supongamos que tenemos los siguientes datos:

	EXPUESTOS	NO EXPUESTOS	TOTAL
Casos	a	b	a+b
Controles	c	d	c+d

Donde:

$$OR = \frac{A \times D}{C \times B}$$

- 2. Calcular el error estándar del logaritmo del OR:** el siguiente paso es calcular el error estándar del logaritmo natural del OR como $\log(OR)$:

$$SE = (\log(RR)) = \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$$

- 3. Calcular los límites del intervalo de confianza:** utilizamos el error estándar para calcular los límites del intervalo de confianza para $\log(OR)$. Generalmente, se usa un nivel de confianza del 95%, lo que corresponde a un valor de Z de 1.96:

$$\text{Límite inferior del IC para } \log(OR) = \log(OR) - 1.96 \times SE(\log(OR))$$

$$\text{Límite superior del IC para } \log(OR) = \log(OR) + 1.96 \times SE(\log(OR))$$

4. **Transformar los límites del IC a escala original:** finalmente convertimos los límites del IC del $\log(OR)$ a la escala original del OR tomando el antilogaritmo (exponencial):

$$\text{Límite inferior del IC para } OR = e^{\log(OR) - 1.96 \times SE(\log(OR))}$$

$$\text{Límite superior del IC para } OR = e^{\log(OR) + 1.96 \times SE(\log(OR))}$$

Suponiendo que estamos realizando un estudio de casos y controles para evaluar la relación entre el consumo de un medicamento y la incidencia de osteoartritis, donde:

	EXPUESTOS AL MEDICAMENTO	NO EXPUESTOS AL MEDICAMENTO	TOTAL
Casos	20	80	100
Controles (osteoartritis)	50	150	200

Calculamos el OR:

$$OR = \frac{20 \times 150}{50 \times 80} = 0.75$$

Calculamos el error estándar del logaritmo del OR:

$$SE = (SE(\log(OR))) = \sqrt{\frac{1}{20} + \frac{1}{80} + \frac{1}{50} + \frac{1}{150}} = 0.2987$$

Calcular los límites del intervalo de confianza para $\log(OR)$:

$$\text{Límite inferior del IC para } \log(OR) = \log(0.75) - 1.96 \times 0.2987 = -0.8732$$

$$\text{Límite superior del IC para } \log(OR) = \log(0.75) + 1.96 \times 0.2987 = -0.2978$$

Transformar los límites del IC a la escala original:

$$\text{Límite inferior del IC para } OR = e^{-0.8732} = 0.417$$

$$\text{Límite superior del IC para } OR = e^{-0.2978} = 1.346$$

Estos resultados nos demuestran que con un OR de 0.75, significa que las personas expuestas al medicamento tienen 25% menos de probabilidades de desarrollar osteoartritis en comparación con las personas no expuestas al medicamento. El intervalo de confianza del 95% para el OR es de aproximadamente 0.42 a 1.35. Dado que el intervalo de confianza incluye el valor 1, no podemos concluir que la asociación es estadísticamente significativa.

Pruebas de hipótesis para estimar los valores de

Las pruebas de hipótesis son fundamentales para determinar la significancia estadística de los resultados en estudios epidemiológicos. En el contexto de OR y RR, la prueba de hipótesis se utiliza para evaluar si la asociación observada entre una exposición y una enfermedad es significativa.

1. Formulación de hipótesis:

- » **Hipótesis nula (H_0):** no hay asociación entre la exposición y la enfermedad ($OR = 1$ o $RR = 1$).
- » **Hipótesis alternativa (H_1):** hay una asociación entre la exposición y la enfermedad ($OR \neq 1$ o $RR \neq 1$).

2. Calcular el estadístico de prueba: el estadístico de prueba para OR y RR se basa en el logaritmo del OR y RR, ya que estos logaritmos son aproximadamente normales.

Para el odds ratio (OR):

$$Z = \frac{\log(OR)}{SE = (\log(OR))}$$

Para el riesgo relativo (RR):

$$Z = \frac{\log(RR)}{SE = (\log(RR))}$$

Donde SE es el error estándar del logaritmo de OR o RR.

3. Calcular el error estándar (SE):

Para el OR:

$$SE = (\log(OR)) = \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$$

Para el RR:

$$SE = (\log(RR)) = \sqrt{\frac{1}{a} + \frac{1}{c} + \frac{1}{a+b} + \frac{1}{c+d}}$$

4. **Calcular el valor de p :** el valor de p se calcula utilizando la distribución normal estándar (Z). El valor de p es la probabilidad de observar un valor del estadístico de prueba tan extremo o más extremo que el observado, bajo la hipótesis nula.

Para un test bilateral:

$$P = 2 \times (1 - \Phi(|Z|))$$

Por ejemplo, suponga que está evaluando la relación entre el consumo de un medicamento y la incidencia de asma. Los datos recolectados son los siguientes:

	EXPUESTOS AL MEDICAMENTO	NO EXPUESTOS AL MEDICAMENTO	TOTAL
Casos (asma)	10	90	100
Controles	50	150	200

Calculamos el OR:

$$OR = \frac{10 \times 150}{90 \times 50} = 0.33$$

Calculamos el error estándar del logaritmo del OR:

$$SE = (\log(OR)) = \sqrt{\frac{1}{10} + \frac{1}{90} + \frac{1}{50} + \frac{1}{150}} = 0.3712$$

Calculamos el estadístico de prueba (Z):

$$Z = \frac{\log(0.33)}{0.3712} = -2.99$$

Calculamos el valor de p :

$$P = 2 \times (1 - \Phi(|-2.99|)) \approx 2 \times (1 - 0.9986) = 0.0028$$

El valor de p es 0.0028, lo que significa que hay suficiente evidencia para rechazar la hipótesis nula a un nivel de significancia del 5% (α). Esto indica que hay una asociación significativa entre el consumo del medicamento y una menor incidencia de asma.

- ♦ **Ejemplo 20.3:** suponga que está realizando un estudio epidemiológico para investigar la relación entre el tabaquismo y el desarrollo de una enfermedad pulmonar crónica (EPC), para ello en su estudio incluye 1000 individuos. Su objetivo es estimar el RR y el OR. Este ejemplo solo tiene como objetivo la estimación de las medidas, no su interpretación. Tome en cuenta que no se está considerando el diseño epidemiológico del estudio aquí, por lo que los resultados no son representativos:

```
# Fijar la semilla para reproducibilidad
set.seed(123)

# Definir el tamaño de la muestra
n ← 1000

# Generar identificadores únicos para los individuos
id ← 1:n

# Generar estado de fumador (0 = No fumador, 1 = Fumador) con una prevalencia del 30%
smoker ← sample(c(0, 1), n, replace = TRUE, prob = c(0.7, 0.3))

# Generar presencia de enfermedad pulmonar crónica (0 = No, 1 = Sí)
# Asumimos que el 20% de los fumadores desarrollan EPC y el 5% de los no fumadores
disease ← ifelse(smoker == 1, sample(c(0, 1),
                                     sum(smoker == 1),
                                     replace = TRUE, prob = c(0.8, 0.2)),
                 sample(c(0, 1), sum(smoker == 0),
                       replace = TRUE, prob = c(0.95, 0.05)))

# Crear un Dataframe con los datos generados
data ← data.frame(id, smoker, disease)

# Mostrar las primeras filas del Dataframe
head(data)
```

id	smoker	disease
----	--------	---------

1	1	0	0
2	2	1	0
3	3	0	0
4	4	1	1
5	5	1	1
6	6	0	0

Creamos una tabla de contingencia:

```
# Crear una tabla de contingencia con los datos
table_data <- table(data$smoker, data$disease)
colnames(table_data) <- c("No enfermo", "Enfermo")
rownames(table_data) <- c("No fumador", "Fumador")
table_data
```

	No enfermo	Enfermo
No fumador	667	38
Fumador	235	60

Ahora ya podemos estimar el RR y OR. Nota: no se está considerando el diseño del estudio.

```
# Calcular el Riesgo Relativo (RR) usando epiR
# epi.2by2 crea una tabla 2x2 y calcula varias medidas epidemiológicas
rr_result <- epi.2by2(table_data, method = "cohort.count", conf.level = 0.95)
```

```
# Mostrar los resultados del Riesgo Relativo
rr_result$massoc.summary
```

var	est	lower	upper
1 Inc risk ratio	1.187657	1.118166	1.261466
2 Inc odds ratio	4.481523	2.907438	6.907817
3 Attrib inc risk *	14.948912	10.062496	19.835328
4 Attrib fraction in exposed (%)	15.800574	10.567829	20.727147
5 Attrib inc risk in population *	10.538983	5.589833	15.488133

```
6 Attrib fraction in population (%)      11.684017      7.634753      15.555762
# Calcular la Razón de Probabilidades (OR) usando epiR
# Aquí usamos la misma tabla, pero con el método "case.control"
or_result <- epi.2by2(table_data, method = "case.control", conf.level = 0.95)

# Mostrar los resultados de la Razón de Probabilidades
or_result$massoc.summary
```

var	est	lower	upper
1 Exposure odds ratio (W)	4.481523	2.907438	6.907817
2 Attrib fraction (est) in exposed (%)	77.644142	64.879009	85.919328
3 Attrib fraction (est) in population (%)	57.446415	48.432423	64.884766

Hazard ratio (HR)

El hazard ratio (HR) es otra medida de asociación utilizada en estudios epidemiológicos y clínicos. Es una medida de la tasa de ocurrencia de un evento (muerte, recaída de una enfermedad) en un grupo en comparación con otro, durante un periodo específico. Es especialmente útil en estudios de supervivencia, donde el tiempo hasta el evento es una variable crucial. El HR se estima generalmente usando modelos de regresión de riesgos proporcionales de Cox, y se interpreta como la razón de las tasas de riesgo entre dos grupos.

El HR se puede interpretar con relación a 1:

- ♦ **HR = 1:** no hay diferencia en el riesgo de ocurrencia del evento entre los dos grupos.
- ♦ **HR > 1:** el grupo en el numerador tiene un mayor riesgo de ocurrencia del evento en comparación con el grupo en el denominador.
- ♦ **HR < 1:** el grupo en el numerador tiene un menor riesgo de ocurrencia del evento en comparación con el grupo en el denominador.

El cálculo del HR involucra métodos estadísticos avanzados, como los modelos de regresión de Cox. El HR se define como la razón de las tasas de riesgo entre dos grupos en un estudio, típicamente un grupo tratado y un grupo de control. Este describe la tasa de ocurrencia del evento en el grupo tratado comparada con la tasa en el grupo de control, durante un periodo específico.

Modelo de regresión de Cox

El modelo de Cox es una herramienta poderosa y flexible para el análisis de datos de supervivencia. Este modelo permite estimar el HR y sus intervalos de confianza, proporcionando información crucial sobre la eficacia de un tratamiento.

Principales características del modelo de Cox:

- ♦ **Riesgos proporcionales:** asume que el HR es constante a lo largo del tiempo.
- ♦ **Variables tiempo-dependientes:** puede incluir covariables que cambian con el tiempo.
- ♦ **Datos censurados:** maneja eficientemente los datos censurados (pacientes que no experimentan el evento durante el periodo de estudio).

Imaginemos un estudio clínico que evalúa un nuevo tratamiento para el cáncer de pulmón. El objetivo es comparar el tiempo hasta la recaída entre el grupo tratado y el grupo de control.

Datos del estudio:

- ♦ **Grupo tratado:** 50 pacientes
- ♦ **Grupo de control:** 50 pacientes
- ♦ **Seguimiento:** 24 meses
- ♦ **Eventos (recaída):** 50 en el grupo tratado, 50 en el grupo de control

Aplicación del modelo de Cox: el modelo de Cox se ajusta a los datos de supervivencia para estimar el HR. Supongamos que el HR estimado es 0.6.

Interpretación: un HR de 0.6 indica que los pacientes en el grupo tratado tienen un 40% menos de riesgo de recaída en cualquier momento dado en comparación con los pacientes en el grupo de control.

PACIENTE	GRUPO	TIEMPO (MESES)	EVENTO (1= SÍ, 0= NO)
1	Tratado	3	1
2	Tratado	8	1
3	Tratado	12	0

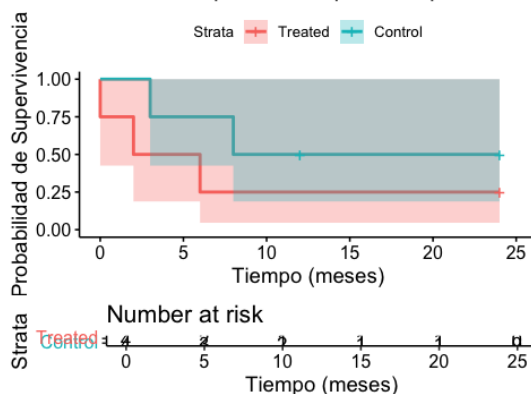
PACIENTE	GRUPO	TIEMPO (MESES)	EVENTO (1= SÍ, 0= NO)
...	...	...	...
50	Tratado	24	0
51	Control	2	1
52	Control	6	1
...	...	...	...
99	Control	0	1
100	Control	24	0

Ventajas del modelo de Cox:

- ❖ **Flexibilidad:** puede manejar múltiples covariables y datos censurados.
- ❖ **No paramétrico:** no requiere suposiciones sobre la distribución del tiempo hasta el evento.
- ❖ **Interpretación clara:** el HR proporciona una medida intuitiva de la eficacia del tratamiento.

Para entender el efecto del tratamiento sobre la supervivencia, calculamos el HR utilizando un modelo de regresión de Cox. Este modelo nos proporciona una medida cuantitativa del efecto del tratamiento sobre el riesgo de evento (recaída, en este caso).

FIGURA 20.1. CURVA DE KAPLAN-MEIER
Curva de Kaplan-Meier para Grupos Tratado



En la [figura 20.1](#) se muestra una curva de Kaplan-Meier comparando la supervivencia de dos grupos de pacientes: los tratados con un nuevo fármaco y el grupo de control. La línea azul representa al grupo tratado y la línea naranja al grupo de control. Las sombras alrededor de las líneas representan los intervalos de confianza al 95%.

Interpretación de la curva:

- ♦ **Eje Y (vertical):** probabilidad de supervivencia.
- ♦ **Eje X (horizontal):** tiempo en meses.
- ♦ **Líneas:** las líneas muestran la proporción de pacientes que sobreviven en cada punto del tiempo.
- ♦ **Sombras:** los intervalos de confianza al 95% para cada grupo.

En la curva de Kaplan-Meier presentada, se puede observar que la probabilidad de supervivencia disminuye con el tiempo en ambos grupos. Sin embargo, el grupo tratado muestra una supervivencia ligeramente mejor en comparación con el grupo de control, lo que sugiere un beneficio potencial del tratamiento.

Cálculo del hazard ratio (HR): para cuantificar esta diferencia en supervivencia, utilizamos el modelo de regresión de riesgos proporcionales de Cox. El HR es una medida que compara las tasas de eventos entre dos grupos a lo largo del tiempo. Un HR menor a 1 indica un beneficio del tratamiento, mientras que un HR mayor a 1 sugiere un mayor riesgo.

Por ejemplo, si el HR calculado es 0.6, esto significa que los pacientes tratados tienen 40% menos de riesgo de experimentar el evento (por ejemplo, recaída) en comparación con los pacientes en el grupo de control.

La combinación de la curva de Kaplan-Meier y el modelo de regresión de Cox proporciona una herramienta robusta para analizar y comparar la supervivencia entre diferentes grupos de tratamiento, permitiendo una interpretación detallada de la eficacia de los tratamientos a lo largo del tiempo.

- ♦ **Ejemplo 20.4:** considere un estudio de cohorte que evalúa el efecto del tratamiento A frente al tratamiento B en la supervivencia de pacientes con una determinada enfermedad. Queremos estimar el HR para comparar los riesgos de muerte entre los dos grupos de tratamiento.

1. Prepare los datos:

```
# Instalación
install.packages("survival")

# Carga de la librería survival
library(survival)

# Fijar la semilla para reproducibilidad
set.seed(123)

# Definir el tamaño de la muestra
n ← 500

# Generar identificadores únicos para los individuos
id ← 1:n

# Asignar aleatoriamente los tratamientos (0 = Tratamiento A, 1 =
Tratamiento B)
treatment ← sample(c(0, 1), n, replace = TRUE)

# Generar tiempos de seguimiento (en días)
# Usamos una distribución exponencial para los tiempos de supervivencia
follow_up ← rexp(n, rate = ifelse(treatment == 0, 0.01, 0.015)) # Tasa
de eventos diferente según el tratamiento

# Generar eventos de muerte (1 = Muerte, 0 = Censura)
status ← sample(c(0, 1), n, replace = TRUE, prob = c(0.6, 0.4)) # 40%
de eventos

# Crear un Dataframe con los datos generados
data ← data.frame(id, treatment, follow_up, status)

# Mostrar las primeras filas del Dataframe
head(data)
```

	id	treatment	follow_up	status
1	1	0	110.75715	0
2	2	0	115.89130	0
3	3	0	84.15477	1
4	4	1	157.26721	0
5	5	0	115.49643	0
6	6	1	120.69366	1

2. Ajuste el modelo Cox:

```
# Ajustar el modelo de Cox usando la función coxph
cox_model <- coxph(Surv(follow_up, status) ~ treatment, data = data)

# Mostrar el resumen del modelo
summary(cox_model)

Call:
coxph(formula = Surv(follow_up, status) ~ treatment, data = data)

n= 500, number of events= 205

              coef exp(coef) se(coef)      z Pr(>|z|)
treatment 0.3728      1.4518  0.1420 2.625  0.00866 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
treatment      1.452      0.6888      1.099      1.918

Concordance= 0.552 (se = 0.02 )
Likelihood ratio test= 6.81 on 1 df,  p=0.009
Wald test               = 6.89 on 1 df,  p=0.009
Score (logrank) test = 6.97 on 1 df,  p=0.008
```

3. Interpretación: el HR se encuentra en el resumen del modelo, en la sección correspondiente a la variable *treatment*. Un HR mayor que 1 indica que el tratamiento B (codificado como 1) está asociado con un mayor riesgo de muerte en comparación con el tratamiento A (codificado como 0).

Riesgo atribuible

El riesgo atribuible (RA) es una medida de impacto utilizada en epidemiología para cuantificar la cantidad de riesgo que puede atribuirse a un factor específico. Esta medida es útil para entender la magnitud del problema de salud en una población expuesta a un factor de riesgo en comparación con una población no expuesta.

Riesgo atribuible en los expuestos (RAe)

El RAe cuantifica la proporción de la incidencia de una enfermedad en un grupo de individuos expuestos a un factor de riesgo que puede atribuirse específicamente a esa exposición. Es útil para entender cuánto de la incidencia de la enfermedad podría evitarse si se eliminara la exposición al factor de riesgo en la población expuesta.

Se puede calcular de dos maneras, dependiendo de los datos disponibles:

1. RAe a partir de incidencia:

$$RAe = \frac{I_e - I_o}{I_e}$$

Donde:

- » (I_e) : incidencia de la enfermedad en el grupo expuesto.
- » (I_o) : incidencia de la enfermedad en el grupo no expuesto.

2. RAe a partir de riesgo relativo:

$$RAe = \frac{RR - 1}{RR}$$

Donde:

- » (RR): riesgo relativo, que es la razón de la incidencia en el grupo expuesto respecto a la incidencia en el grupo no expuesto.

El RAe se puede interpretar con relación a o:

- ♦ **RAe = 0**: indica que no hay diferencia en el riesgo de la enfermedad entre los expuestos y los no expuestos.
- ♦ **RAe > 0**: indica que una proporción de la incidencia de la enfermedad en los expuestos puede atribuirse a la exposición.
- ♦ **RAe < 0**: indica una protección de la exposición contra la enfermedad (esto es raro y generalmente indica un error en los datos o en la interpretación).

Por ejemplo, suponga que en un estudio encuentra que la incidencia de cáncer de pulmón en fumadores I_e es de 0.30 (30 casos por 100 personas) y en no fumadores I_o es de 0.05 (5 casos por 100 personas).

Usando incidencia:

$$RAe = \frac{0.3 - 0.05}{0.3} = 0.833$$

Esto indica que el 83.3% de los casos de cáncer de pulmón en fumadores pueden atribuirse al hábito de fumar.

Usando riesgo relativo:

Primero calculamos el (RR):

$$RA = \frac{I_e}{I_o} = \frac{0.3}{0.05} = 6$$

Luego el (RAe):

$$RAe = \frac{6 - 1}{6} = 0.833$$

Esto también indica que el 83.33% de los casos de cáncer de pulmón en fumadores pueden atribuirse al hábito de fumar.

El RAe proporciona información valiosa para la salud pública, ya que permite identificar y cuantificar el impacto de la eliminación de ciertos factores de riesgo en la prevención de enfermedades.

Riesgo atribuible poblacional

El riesgo atribuible poblacional (RAP) es una medida epidemiológica que estima la proporción de la incidencia de una enfermedad en la población total que puede atribuirse a una exposición específica. Esta medida es útil para evaluar el impacto potencial en la salud pública de eliminar un factor de riesgo en la población general.

El RAP se puede calcular de varias maneras, dependiendo de los datos disponibles:

1. RAP a partir de la incidencia en la población:

$$RAP = \frac{I_p - I_o}{I_p}$$

Donde:

- » (I_p) : incidencia de la enfermedad en la población total.
- » (I_o) : incidencia de la enfermedad en el grupo no expuesto.

2. RAP a partir del RR y la prevalencia de la exposición ((P_e)):

$$RAP = \frac{P_e \times (RR - 1)}{P_e \times (RR - 1) + 1}$$

Donde:

- » (RR) : riesgo relativo, que es la razón de la incidencia en el grupo expuesto respecto a la incidencia en el grupo no expuesto.
- » (P_e) : prevalencia de la exposición en la población.

El RAP se puede interpretar con relación a o:

- ♦ **RAP = o**: indica que no hay diferencia en el riesgo de la enfermedad en la población debido a la exposición.
- ♦ **RAP > o**: indica que una proporción de la incidencia de la enfermedad en la población total puede atribuirse a la exposición.
- ♦ **RAP < o**: indica una protección de la exposición contra la enfermedad en la población (generalmente indica un error en los datos o en la interpretación).

Por ejemplo, suponga que en un estudio encuentra que la incidencia de cáncer de pulmón en la población total I_p es de 0.10 (10 casos por 100 personas), la incidencia en no fumadores (I_o) es de 0.05 (5 casos por 100 personas), la prevalencia de fumadores en la población P_e es de 0.30 (30%) y el riesgo relativo RR es de 6.

Usando la incidencia en la población:

$$RAP = \frac{0.1 - 0.05}{0.01} = 0.5$$

Esto indica que el 50% de los casos de cáncer de pulmón en la población total pueden atribuirse al hábito de fumar.

Usando el riesgo relativo (RR) y la prevalencia de la exposición (P_e):

$$RAP = \frac{0.3 \times (6 - 1)}{0.3 \times (6 - 1) + 1} = 0.6$$

Esto indica que el 60% de los casos de cáncer de pulmón en la población total pueden atribuirse al hábito de fumar.

El RAP proporciona una estimación del impacto potencial en la salud pública de eliminar un factor de riesgo en la población general, lo que es fundamental para la planificación y la implementación de intervenciones de salud pública.

Ejercicios

En este capítulo no se muestran las resoluciones de los ejercicios, pero puede utilizar los ejemplos de este capítulo y anteriores para resolverlos.

- ♦ **Ejercicio 20.1:** un estudio epidemiológico se llevó a cabo para investigar la prevalencia y la incidencia de una enfermedad crónica en una población de 1000 individuos. Se registró si cada persona tiene la enfermedad y el tiempo de seguimiento:

1. Genere un conjunto de datos con el código de enseguida.
2. Calcule la prevalencia de la enfermedad en la población.
3. Calcule la incidencia acumulada y la tasa de incidencia de la enfermedad en la población.

```
# Definir el tamaño de la muestra
n ← 1000

# Generar identificadores únicos para los individuos
id ← 1:n

# Generar edades aleatorias entre 20 y 80 años
age ← sample(20:80, n, replace = TRUE)

# Generar géneros aleatorios (Male/Female)
gender ← sample(c("Male", "Female"), n, replace = TRUE)

# Generar la presencia de enfermedad (0 = No, 1 = Sí) con una pre-
valencia del 10%
disease ← sample(c(0, 1), n, replace = TRUE, prob = c(0.9, 0.1))

# Definir la fecha de inicio y fin de la observación
start_date ← as.Date("2020-01-01")
end_date ← as.Date("2023-01-01")

# Generar tiempos aleatorios de riesgo (en días) hasta un máximo de 3 años
time_at_risk ← sample(0:1095, n, replace = TRUE)

# Calcular la fecha del primer caso (si aplica)
event_date ← start_date + time_at_risk
event_date[disease == 0] ← NA # Solo asignar fecha a los casos con
enfermedad

# Crear un Dataframe con los datos generados
data ← data.frame(id, age, gender, disease, start_date, event_date,
time_at_risk)

# Mostrar las primeras filas del Dataframe
head(data)
```


- ♦ **Ejemplo 20.5:** un estudio se llevó a cabo para investigar la relación entre el tabaquismo y el desarrollo de una enfermedad pulmonar crónica (EPC). Se recogieron datos de 1000 individuos, registrando si son fumadores y si han desarrollado EPC:
- » Genere un conjunto de datos simulado que incluya información sobre el tabaquismo y la EPC.
 - » Genere una tabla de contingencia para los datos de fumadores y no fumadores frente a la EPC.
 - » Calcule el riesgo relativo para la EPC en fumadores comparado con no fumadores.
 - » Calcule el OR para la EPC en fumadores comparado con no fumadores.

Nota: no se toma en cuenta el contexto del estudio.

```
# Definir el tamaño de la muestra
n ← 1000

# Generar identificadores únicos para los individuos
id ← 1:n

# Generar estado de fumador (0 = No fumador, 1 = Fumador) con una
prevalencia del 30%
smoker ← sample(c(0, 1), n, replace = TRUE, prob = c(0.7, 0.3))

# Generar presencia de enfermedad pulmonar crónica (0 = No, 1 = Sí)
disease ← ifelse(smoker == 1, sample(c(0, 1), sum(smoker == 1), re-
place = TRUE, prob = c(0.8, 0.2)),
  sample(c(0, 1), sum(smoker == 0), replace = TRUE, prob = c(0.95, 0.05)))

# Crear un Dataframe con los datos generados
data ← data.frame(id, smoker, disease)

# Mostrar las primeras filas del Dataframe
head(data)
```

En este código tuvimos que utilizar `ifelse` para asignar la enfermedad a los fumadores y no fumadores. Esta función en palabras sencillas, evalúa una condición y devuelve un valor si la condición es verdadera y otro valor si es falsa. En este caso, si la persona es fumadora, la probabilidad de tener la enfermedad es del 20%, mientras que, si no es fumadora, la probabilidad es del 5%.

El uso completo de `if`, `else` y `ifelse` se escapan de los fines de este libro, sin embargo, le recomendamos revisar: [If, Else, Ifelse](https://www.datacamp.com/tutorial/if-else-function-r) y <https://www.datacamp.com/tutorial/if-else-function-r> para más información.

- ♦ **Ejercicio 20.2:** un estudio de cohorte se llevó a cabo para evaluar el efecto del tratamiento A frente al tratamiento B en la supervivencia de pacientes con una determinada enfermedad. Queremos estimar el hazard ratio (HR) para comparar los riesgos de muerte entre los dos grupos de tratamiento.
 - » Genere un conjunto de datos simulado que incluya información sobre los tratamientos y los tiempos de seguimiento.
 - » Ajuste un modelo de Cox para comparar los riesgos de muerte entre los dos grupos de tratamiento.
 - » Interprete sus resultados.

```
# Definir el tamaño de la muestra
n ← 500

# Generar identificadores únicos para los individuos
id ← 1:n

# Asignar aleatoriamente los tratamientos (0 = Tratamiento A, 1 =
# Tratamiento B)
treatment ← sample(c(0, 1), n, replace = TRUE)

# Generar tiempos de seguimiento (en días)
follow_up ← rexp(n, rate = ifelse(treatment == 0, 0.01, 0.015))

# Generar eventos de muerte (1 = Muerte, 0 = Censura)
status ← sample(c(0, 1), n, replace = TRUE, prob = c(0.6, 0.4))
```

```
# Crear un Dataframe con los datos generados
data ← data.frame(id, treatment, follow_up, status)
```

```
# Mostrar las primeras filas del Dataframe
head(data)
```

	id	treatment	follow_up	status
1	1	1	15.214005	0
2	2	0	8.995285	0
3	3	0	152.390240	1
4	4	1	92.026377	0
5	5	1	11.007216	0
6	6	0	42.910834	1

Referencias

- Abin Varghese, T., Shaheen, N., y Hussein, M. (2022). An efficient confidence interval estimation for prevalence calculated from misclassified data. *Biostatistics & Epidemiology*, 7.
- Adam, M., Denwood, M., y Christiansen, L. E. (2023). Estimating true prevalence through questionnaire data. *Journal of Medical Virology*, 95.
- Ahlbom, A. (2020). Epidemiology is about disease in populations. *European Journal of Epidemiology*, 35, 1111-1113.
- Alex, M., Mahjoub, Y., Larry, C. S., y Pringsheim, T. (2022). Prevalence and Incidence of Huntington's Disease: An Updated Systematic Review and Meta-Analysis. *Movement Disorders*, 37, 2327-2335.
- Armigón Pallás, J. M., y Jiménez Villa, J. (2000). *Métodos de investigación clínica y epidemiológica*. Elsevier S.A.
- Atramentova, L., y Utevska, O. (2021). Evaluation of population frequency for age-dependent multifactorial disorders. *Faktori eksperimental'noi evolucii organizmiv*.
- Bayer, D., Fay, M., y Graubard, B. (2022). Confidence intervals for prevalence estimates from complex surveys with imperfect assays. *Statistics in Medicine*, 42, 1822-1867.
- Cheon-Sig, L., Stephen, A. S., y Sarjinder, S. (2021). Estimation of odds ratio, attributable risk, relative risk, correlation coefficient and other parameters using randomized response techniques. *Behaviormetrika*, 48, 371-392.

- Di Maso, M., Bravi, F., Polesel, J., Negri, E., Decarli, A., Serraino, D. , ... y La Vecchia, C. (2020). Attributable fraction for multiple risk factors: Methods, interpretations, and examples. *Statistical Methods in Medical Research*, 29(3), 854-865.
- Diallo, G., y Bordea, G. (2021). Public Health and Epidemiology Informatics: Recent Research Trends. *Yearbook of Medical Informatics*, 30, 280-282.
- Didelez, V., y Stensrud, M. (2021). On the logic of collapsibility for causal effect measures. *Biometrical Journal*, 64, 235-242.
- Friis, R., y Sellers, T. (1996). *Epidemiology for public health practice* (4th ed.). Jones & Bartlett Learning.
- Fujiwara, T. (2020). Pitfalls and Tips for Statistical Methods in Epidemiology: A New Series of Special Articles Has Started. *Journal of Epidemiology*, 30, 151-152.
- Kleinbaum, D. G., Kupper, L. L., y Morgenstern, H. (1982). *Epidemiology research: Principles and quantitative methods*. Van Nostrand Reinhold Company.
- Last, J. M., e International Epidemiological Association. (2001). *A Dictionary of Epidemiology*. University Press.
- Lininger, M., Root, H., Camplain, R., y Barger, S. (2022). Describing the appropriate use and interpretation of odds and risk ratios. *Research in Sports Medicine*, 1-7.
- Martínez-Camblor, P., Todd, A. M., y O'Malley, A. J. (2021). A robust hazard ratio for general modeling of survival-times. *International Journal of Biostatistics*, 18, 537-551.
- Merlo, J., Chaix, B., Ohlsson, H., Beckman, A., Johnell, K., Hjerpe, P., ... y Larsson, K. (2006). A brief conceptual tutorial of multilevel analysis in social epidemiology: using measures of clustering in multilevel logistic regression to investigate contextual phenomena. *Journal of Epidemiology & Community Health*, 60, 290-297.
- Migliavaca, C., Cinara, S., Colpani, V., Barker, T., Ziegelmann, P., Munn, Z., y Stein, C. (2022). Meta-analysis of prevalence: I2 statistic and how to deal with heterogeneity. *Research Synthesis Methods*, 13, 363-367.
- Mirón Canelo, J. A., y Andrés-Sánchez, S. (2008). Medidas de frecuencia, asociación e impacto en investigación aplicada. *Medicina y Seguridad del Trabajo*, 54, 93-102.
- Pocock, S., Collier, T., Dandreo, K. J., De Stavola, B. D., Goldman, M., Kalish, L., y ... & McCormack, V. (2004). Issues in the reporting of epidemiological studies: a survey of recent practice. *BMJ*, 329, 883.
- Pocock, S., Kleinbaum, D., Kupper, L., y Morgenstern, H. (1983). *Epidemiologic Research: Principles and Quantitative Methods*. Biometrics, 39, 822.

- Schmidt, C. O., y Kohlmann, T. (2008). Risk quantification in epidemiologic studies. *International Journal of Public Health*, 53, 118-119.
- Selvin, S. (1991). *Statistical Analysis of Epidemiologic Data* (2nd ed.). Oxford University Press.
- Seong Ho, P., y Kyunghwa, H. (2022). How to Clearly and Accurately Report Odds Ratio and Hazard Ratio in Diagnostic Research Studies? *Korean Journal of Radiology*, 23, 777-784.
- Spruance, S. L., Reid, J. E., Grace, M., y Samore, M. (2004). Hazard ratio in clinical trials. *Antimicrobial Agents and Chemotherapy*, 48(8), 2787-2792.
- Tsao, C. W., Aday, A. W., Almarzooq, Z. I., Anderson, C. A. M., Arora, P., Avery, C. L., ... y American Heart Association Council on Epidemiology and Prevention Statistics Committee and Stroke Statistics Subcommittee. (2023). Heart Disease and Stroke Statistics—2023 Update: A Report From the American Heart Association. *Circulation*, 147.
- Walter, S. (2000). Choice of effect measure for epidemiological data. *Journal of Clinical Epidemiology*, 53(9), 931-939.
- Wei, Y., Miao, Y., Xu, C., y Jiang, T. (2021). Text Data Mining of Risk Behaviors During the Outbreak of Infectious Disease in Epidemiological Case Reports. *Proceedings of the 6th International Conference on Intelligent Computing and Signal Processing*, 250-253.
- Yuanyuan, L., Wen, W., Yangxin, H., y Henian, C. (2023). How big is a big hazard ratio in clinical trials? *International Journal of Clinical Trials*.

Correlación

Edsaúl Emilio Pérez Guerrero

En este capítulo se describirán las pruebas de correlación. Sin embargo, al inicio del capítulo veremos algunas de las diferencias que existen entre correlación y regresión, ya que son dos pruebas que se suelen confundir, tanto en su fundamento como en su aplicación.

La correlación y la regresión lineal simple son métodos estadísticos que estudian la relación existente entre dos variables numéricas. Partiremos de un ejemplo.

Un grupo de investigadores desean conocer a relación existe entre la densidad mineral y el porcentaje de grasa corporal. Una vista de 15 de los 265 datos del *Dataframe* original se muestra en la [tabla 21.1](#).

**TABLA 21.1. RELACIÓN ENTRE LA DENSIDAD MINERAL
Y EL PORCENTAJE DE GRASA DE 15 PACIENTES**

DENSITY	FAT
1.0708	12.3
1.0853	6.1
1.0414	25.3
1.0751	10.4
1.0340	28.7
1.0502	20.9
1.0549	19.2
1.0704	12.4

DENSITY	FAT
1.0900	4.1
1.0722	11.7
1.0830	7.1
1.0812	7.8
1.0513	20.8
1.0505	21.2
1.0484	22.1

Por definición, la correlación mide la asociación de dos variables numéricas y, en cierta medida, mide cuánto cambia una variable con respecto al cambio de la otra. Aunque esta relación nunca será causal.

Mientras que la regresión se usa para explicar o modelar la relación entre una variable continua Y, llamada variable respuesta o variable dependiente, y una variable X, llamada variable independiente.

Otras diferencias importantes entre la correlación y la regresión son:

1. La correlación se usa para medir el grado de asociación entre dos variables sin implicar causalidad, mientras que la regresión lineal se utiliza para generar un modelo matemático que permita predecir el valor de una variable a partir de la otra.
2. A diferencia de la regresión lineal, la correlación no tiene en cuenta la asignación de las variables a X e Y; únicamente mide la relación entre dos variables y no es necesario que una de ellas sea dependiente o independiente. De hecho, en la correlación no hay una variable dependiente ni independiente. En cambio, en la regresión lineal sí hay una variable dependiente (Y) que dependerá de los cambios que suceden en una variable independiente (X).
3. A nivel experimental, la correlación se suele emplear cuando ninguna de las variables se ha controlado; simplemente se han medido ambas y se desea saber si están relacionadas. En el caso de estudios de regresión lineal, es más común que una de las variables se controle (tiempo, concentración de reactivo, temperatura...) y se mida la otra.
4. Es común que los estudios de correlación precedan a los de regresión lineal para analizar la relación entre las variables y, si se comprueba que están correlacionadas, se procede a generar el modelo de regresión lineal. La

correlación es más sencilla en términos estadísticos comparada con la regresión. Es importante tener en cuenta que la regresión lineal no implica causalidad, sino que proporciona una manera de predecir la relación entre las variables.

En resumen, la correlación es una medida estadística que indica el grado de asociación entre dos variables, pero no implica causalidad. En cambio, la regresión lineal es un modelo matemático que puede utilizarse para predecir la relación entre dos variables y puede proporcionar información sobre la causalidad. La correlación se utiliza para medir la fuerza de la relación entre dos variables, mientras que la regresión lineal se utiliza para modelar y predecir la relación entre ellas.

Pruebas de correlación

La correlación es la medida que se emplea en estadística para conocer que tan asociadas están dos variables, algunas de sus características son:

- ♦ Su finalidad es examinar la dirección y la magnitud de la asociación entre dos variables cuantitativas.
- ♦ La correlación es una medida que va de -1 a 1, donde -1 indica una correlación negativa perfecta, 0 indica que no hay correlación y 1 indica una correlación positiva perfecta.
- ♦ La correlación no implica causalidad. Aunque dos variables puedan estar correlacionadas, no significa que una variable modifique la otra.
- ♦ La correlación se puede calcular utilizando diferentes métodos, siendo el coeficiente de correlación de Pearson el más común, sobre todo cuando los datos tienen una relación lineal.
- ♦ La correlación es simétrica, es decir, que la correlación entre A y B es igual a la correlación entre B y A.
- ♦ Permite conocer si al aumentar una variable, aumenta la otra o disminuye.
- ♦ Si la relación entre dos variables es perfecta e inversa, $r = -1$, mientras que, si es lineal y directa, $r = 1$.
- ♦ Cuando las dos variables no están correlacionadas, $r = 0$.

Nunca olvide que la correlación no implica causalidad. La [figura 21.1](#) muestra irónicamente cómo la correlación no implica causalidad.

FIGURA 21.1. LA VENTA DE HELADOS SE CORRELACIONA CON EL NÚMERO DE ATAQUES DE TIBURÓN. LA CORRELACIÓN NO ES CAUSALIDAD



Existen varios criterios para establecer si la relación es buena o no, nosotros emplearemos la propuesta por la universidad de [Emory](#) que se relaciona con el tamaño del efecto de Cohen:

- ♦ 0.1: efecto pequeño (relación pequeña)
- ♦ 0.3: efecto moderado (relación moderada)
- ♦ 0.5: efecto fuerte (relación fuerte)

Pruebas de hipótesis en correlación y coeficientes de correlación

Existen tres formas de estimar la correlación mediante los coeficientes de correlación de Pearson, Spearman y Kendall. Se describen a continuación las principales características de los primeros dos, ya que son los más utilizados en la práctica.

Coefficiente de correlación Pearson

La fórmula para estimar el coeficiente de correlación de Pearson es:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

En esta fórmula, r_{xy} representa el coeficiente de correlación entre las variables x e y , x_i e y_i son los valores de las variables x e y en la i -ésima observación, mientras que $\bar{x}$ y $\bar{y}$ son las medias de x e y , respectivamente.

Existe una relación muy importante entre el coeficiente de correlación de Pearson y la covarianza. Para fines prácticos, la covarianza mide cuánta variación hay entre una variable y una segunda. Es decir, la covarianza se refiere a cuánto varía una variable si una segunda variable varía de la misma forma. Por ejemplo, si dos variables tienen la misma variación una con respecto a la otra, la covarianza sería de uno. Un ejemplo de esto, aunque quizá carece de sentido, es medir el peso de un paciente en kg y en libras. Estas mediciones tendrían una covarianza de 1.

Por lo tanto, la fórmula para la estimación del coeficiente de correlación de Pearson se puede simplificar como:

$$r_{xy} = \frac{cov(x,y)}{s_x s_y}$$

Donde: r_{xy} representa el coeficiente de correlación entre las variables x e y , $cov(x, y)$ es la covarianza entre x e y , mientras que s_x y s_y son las desviaciones estándar de x e y , respectivamente.

Prueba de hipótesis en el coeficiente de correlación Pearson

Nuestra intención es tratar de demostrar que el coeficiente de correlación es distinto de cero. Por lo tanto:

- ♦ $H_o: r = 0$
- ♦ $H_A: r \neq 0$

Para poder concluir si rechazamos o aceptamos nuestra hipótesis, utilizamos la distribución t y la siguiente fórmula:

$$r_{xy} = \frac{r_{xy} \sqrt{n-2}}{\sqrt{1-r_{xy}^2}}$$

Donde: t es el valor calculado para la prueba de significancia del coeficiente de correlación de Pearson, r_{xy} es el coeficiente de correlación de Pearson y n es el tamaño de la muestra.

Los supuestos que se deben cumplir para utilizar el coeficiente de correlación de Pearson son:

- ♦ Las variables se observan sobre una muestra aleatoria de individuos (cada individuo debe tener un par de valores).
- ♦ Existe una asociación lineal entre las dos variables.
- ♦ Para una prueba de hipótesis válida y cálculo de intervalos de confianza, ambas variables deben tener una distribución aproximadamente normal.
- ♦ Ausencia de valores atípicos en el conjunto de datos.

Coeficiente de correlación Spearman

Mientras que el coeficiente de correlación de Pearson asume linealidad, el coeficiente de correlación de Spearman asume una relación monotónica (las variables tienden a moverse en la misma dirección relativa, pero no necesariamente a un ritmo constante) entre dos variables. El coeficiente de correlación de Spearman se utiliza como una alternativa no paramétrica al coeficiente de correlación de Pearson. Se aplica en casos en los que los valores son ordinales o en aquellos casos en los que los valores son continuos, pero no cumplen con la condición de normalidad. La ventaja de utilizar el coeficiente de Spearman es que trabaja con rangos, lo que lo hace menos sensible a valores extremos que el coeficiente de Pearson.

La estimación del coeficiente de correlación de Spearman p o r_s se basa en la siguiente ecuación:

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

En esta fórmula, p es el coeficiente de correlación de Spearman, n es el tamaño de la muestra y d_i es la diferencia en los rangos de los valores i en cada variable.

La hipótesis nula para este coeficiente es: $H_0: p = 0$ que se puede comprobar mediante el estadístico t con la siguiente fórmula:

$$t = \frac{r_s}{\sqrt{\frac{1 - r_s^2}{n - 2}}}$$

Donde: t es el valor calculado para la prueba de significancia del coeficiente de correlación de Spearman, r_s es el coeficiente de correlación de Spearman y n es el tamaño de la muestra.

Para utilizar el coeficiente de correlación de Spearman, es necesario que:

1. Las variables se observen en una muestra aleatoria de individuos y que haya una asociación monótona entre ellas.
2. Que la variable utilizada sea al menos de tipo ordinal.

No olvide que la asociación monótona indica que las variables tienden a moverse en la misma dirección relativa, pero no necesariamente a un ritmo constante. Es importante destacar que todas las correlaciones lineales son monótonas, pero no todas las asociaciones monótonas son lineales, ya que también puede haber asociaciones monótonas no lineales. Simplemente un número de positivo indica que ambas variables incrementan o disminuyen en el mismo sentido, mientras que un coeficiente negativo indica que una aumenta y la otra disminuye.

La [tabla 21.2](#) muestra una comparación entre el coeficiente de correlación de Spearman y Pearson:

TABLA 21.2. COMPARACIÓN ENTRE EL COEFICIENTE DE CORRELACIÓN DE PEARSON Y SPEARMAN

TIPO DE VARIABLE	COEFICIENTE DE CORRELACIÓN DE PEARSON	COEFICIENTE DE CORRELACIÓN DE SPEARMAN
Variables	Variables continuas que tengan una relación lineal y una distribución normal.	Variables continuas que no cumplen con el supuesto de normalidad o variables ordinales.
Supuestos	1. Muestra aleatoria. 2. Asociación lineal. 3. Distribución normal. 4. Ausencia de valores atípicos.	No tiene supuestos específicos sobre la distribución o forma funcional de las variables.
Fórmula	$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$	$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$

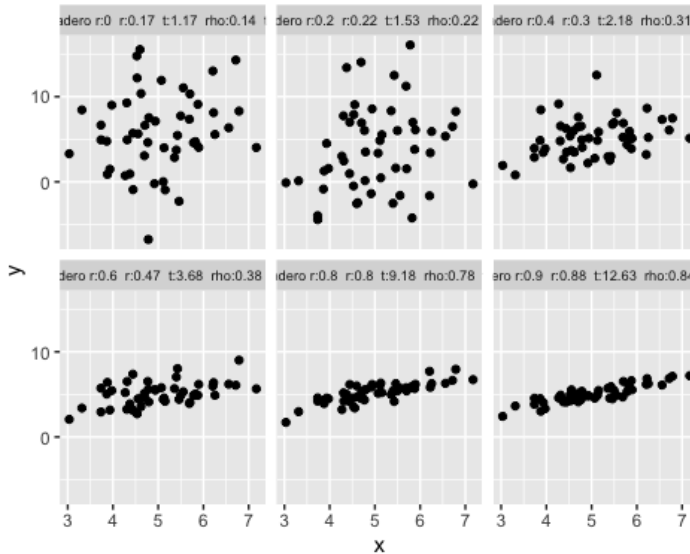
La correlación y el problema visual

Establecer la fuerza de correlación es muy difícil de visualizar mediante gráficas.

La [figura 21.2](#) trata de ejemplificar esta situación. En esta figura, se muestran varios gráficos de dispersión con diferentes valores de correlación r . Cada gráfico tiene puntos que representan la relación entre dos variables x e y . En la parte

superior de cada gráfico se indica el “Verdadero r ” (correlación verdadera) y la “ r ” observada. La correlación verdadera es el valor subyacente de la relación lineal, mientras que la observada es la calculada a partir de los datos mostrados. También se muestra el valor t y ρ , que indican otras formas de evaluar la relación entre las variables. A pesar de tener diferentes valores de t , visualmente algunos gráficos pueden parecer similares debido a la distribución y forma de los datos. Por ejemplo, aunque los puntos en algunos gráficos se dispersan ampliamente, el valor de R puede ser similar si la tendencia general es la misma.

FIGURA 21.2. GRÁFICOS DE DISPERSIÓN CON DIFERENTES VALORES DE CORRELACIÓN



Pruebas de correlación en R

Existen varios métodos para medir la correlación entre dos variables, los más utilizados son:

1. **Pearson:** para realizarlo en R se puede utilizar la función: `cor.test(x, y, method = "spearman")`, donde x y y son dos vectores numéricos.
2. **Spearman:** para realizarlo en R se puede utilizar la función: `cor.test(x, y, method = "Spearman")`, donde x y y son dos vectores numéricos. Es la alternativa no paramétrica para las pruebas de correlación.

3. **Kendall:** prueba útil para correlacionar variables ordinales. Para realizarlo en R se puede utilizar la función: `cor.test(x,y,method = "kendall")`, donde `x` y `y` son dos vectores numéricos.

Ejercicios para pruebas de correlación

Para realizar los siguientes ejemplos utilice la base de datos “bodyfat”, la cual contiene mediciones antropométricas de un grupo de pacientes. Es una versión modificada de la base de datos BF de la librería `bbreg`. Puede encontrar más información del *Dataframe* [aquí](#).

Puede descargar la base [aquí](#). Utilice el menú de RStudio para la importación de la base de datos, tome en cuenta que es un archivo `.txt`. En “[Archivos .txt y .csv](#)” se puede consultar más detalle para la importación.

Se muestran los primero 6 datos de la base `bodyfat`:

`head(bodyfat)`

	Density	Fat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle
1	1.0708	12.3	23	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9
2	1.0853	6.1	22	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4
3	1.0414	25.3	22	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0
4	1.0751	10.4	26	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8
5	1.0340	28.7	24	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0
6	1.0502	20.9	24	210.25	74.75	39.0	104.5	94.4	107.8	66.0	42.0	25.6
	Biceps		Forearm		Wrist							
1	32.0		27.4		17.1							
2	30.5		28.9		18.2							
3	28.8		25.2		16.6							
4	32.4		29.4		18.2							
5	32.2		27.7		17.7							
6	35.7		30.6		18.8							

Para fines de estos ejercicios adjunte la base datos utilizando la función `attach()`:

`attach(bodyfat)`

Supongamos que queremos conocer la correlación de las variables `Weight` y `Height`. Determine los coeficientes de correlación: Pearson, Spearman y Kendall.

Para Pearson:

```
cor(Weight, Height)
[1] 0.3082785
cor.test(Weight, Height)
```

Pearson's product-moment correlation

```
data: Weight and Height
t = 5.1239, df = 250, p-value = 5.99e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1920207 0.4160038
sample estimates:
      cor
0.3082785
```

Para Spearman:

```
cor(Weight, Height, method = "spearman")
[1] 0.5153362
cor.test(Weight, Height, method = "spearman")
Warning in cor.test.default(Weight, Height, method = "spearman"): Cannot
compute exact p-value with ties
```

Spearman's rank correlation rho

```
data: Weight and Height
S = 1292660, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
```

```

      rho
0.5153362
Para Kendall:
cor(Weight, Height, method = "kendall")
[1] 0.3657961
cor.test(Weight, Height, method = "kendall")

```

Kendall's rank correlation tau

```

data: Weight and Height
z = 8.5352, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.3657961

```

Ahora determine la correlación que existe entre el peso Weight y la densidad mineral ósea Density. Además, cree gráficos de dispersión y agregue los valores de p de la prueba de Pearson:

```

cor(Weight, Density)
[1] -0.5940619
cor.test(Weight, Density)

```

Pearson's product-moment correlation

```

data: Weight and Density
t = -11.677, df = 250, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6685561 -0.5077639
sample estimates:
      cor
-0.5940619

```


Para Spearman:

```
cor(Weight, Density, method = "spearman")
[1] -0.6014411
cor.test(Weight, Density, method = "spearman")
Warning in cor.test.default(Weight, Density, method = "spearman"): Cannot
compute exact p-value with ties
```

Spearman's rank correlation rho

```
data: Weight and Density
S = 4271245, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
-0.6014411
```

Para Kendall:

```
cor(Weight, Density, method = "kendall")
[1] -0.4297207
cor.test(Weight, Density, method = "kendall")
```

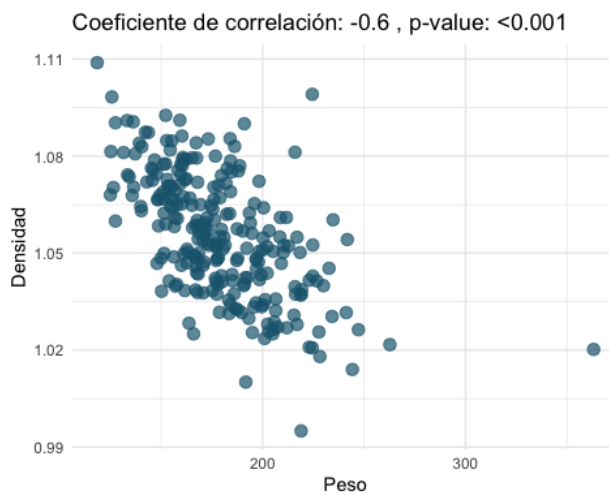
Kendall's rank correlation tau

```
data: Weight and Density
z = -10.145, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
-0.4297207
```

Para la gráfica de correlación puede utilizar ggplot:

```
library(ggplot2)
# Incluye el valor del coeficiente de correlación y el valor de p en
# el título o como anotación en el gráfico
coef_corr <- -0.60
p_value <- "<0.001"

# Crear el gráfico
bodyfat |>
  ggplot(mapping = aes(x = Weight, y = Density)) +
  geom_point(color = "#19647E", size = 3, alpha = 0.7) + # Cambiar el
# color de los puntos
  labs(
    title = paste("Coeficiente de correlación:", coef_corr, ", p-value:", p_value),
    x = "Peso",
    y = "Densidad"
  ) +
  theme_minimal() # Aplicar un tema minimalista
```



Matrices de correlación y gráficos de dispersión

Una matriz de correlación es una tabla o gráfica que muestra los coeficientes de correlación entre múltiples variables. Cada celda en la matriz contiene el coeficiente de correlación entre al menos un par de variables. Hay varias maneras de hacerlo. Vamos a explorar algunas de las opciones que hay en R.

Matrices de correlación utilizando Psych

En R existen muchas funciones que permiten la creación de gráficos y matrices de correlación. En este compendio revisaremos la función: `pairs.panels` de la librería (`psych`).

Primero instale la librería:

```
install.packages("psych")
```

Una vez instalada, llame a la librería:

```
library(psych)
```

```
Attaching package: 'psych'
```

```
The following objects are masked from 'package:ggplot2':
```

```
%+%, alpha
```

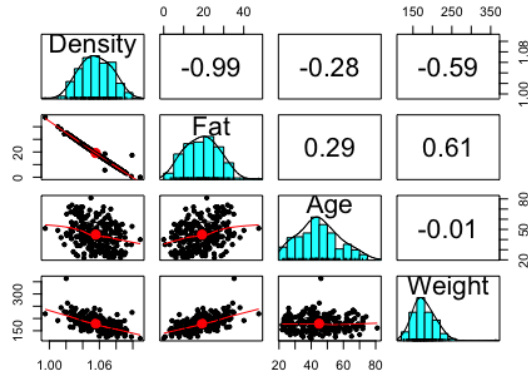
Ahora podemos utilizar la función `pairs.panels`, que se aplica a todo el *Data-frame*, por lo que es importante que su selección solo tenga variables numéricas:

```
pairs.panels(bodyfat)
```

Si utilizó el anterior, su resultado fue un gráfico enorme. Es imposible visualizar correctamente este gráfico, por lo que podemos hacer pedir que se haga primero de algunas variables y luego de otras. Para este ejemplo se harán las primeras 4 y luego las últimas 4 variables. Esto con fines de que los gráficos se puedan mostrar adecuadamente. Usted puede hacerlo de la cantidad de variables que desee.

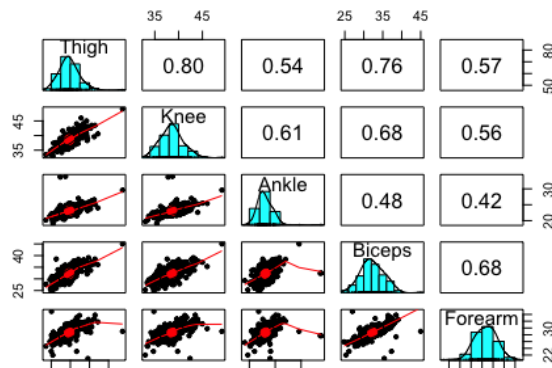
```
pairs.panels(bodyfat[, 1:4], main="Matriz de correlación de las primeras 4 variables")
```

Matriz de correlación de las primeras 4 variables



```
pairs.panels(bodyfat[, 10:14], main="Matriz de correlación de las últimas 4 variables")
```

Matriz de correlación de las últimas 4 variables



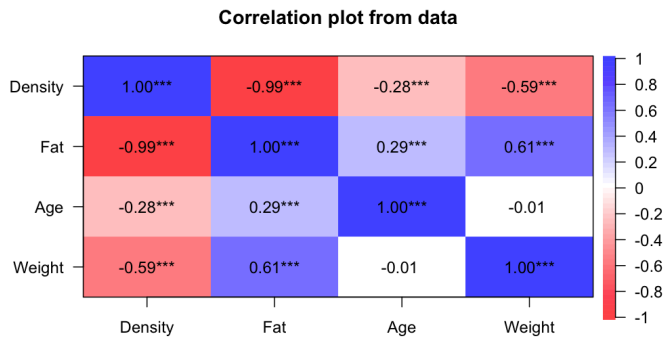
Estos gráficos muestran el coeficiente de correlación, una gráfica de distribución una nube de puntos de la relación de las dos variables. Por ejemplo, el coeficiente de correlación entre Fat y Density es de 0.99 en la primera gráfica. El coeficiente de correlación entre Height y Weight es 0.31.

La librería `psych` contiene una función que permite crear otra matriz de correlación e incluir la significancia estadística en ellas.

```
psych::cor.plot(bodyfat[,1:4], stars = T, cex = 1)
```

No se muestra la gráfica debido a los problemas de dimensiones. Pero usted puede ver algo así:

FIGURA 21.3. EJEMPLO DE MATRIZ DE CORRELACIÓN UTILIZANDO COR.PLOT



Esta gráfica muestra los valores del coeficiente de correlación, en azul para los positivos y en rojo para los negativos. Además, marca la significancia estadística con asteriscos.

Matrices de correlación utilizando Corrr

Ahora vamos a utilizar la librería `corrr` para realizar nuestra matriz de correlación:

```
install.packages("corrr") # Si es necesario
```

Utilice el siguiente código para obtener una matriz de correlación con el método de Spearman de los primeros 4 datos de la base `Bodyfat`. En su Rstudio usted verá los números de colores, se marcan en color las correlaciones negativas y en blanco las positivas. Además, hay un par de coeficientes con un guion por debajo, por ejemplo, en el coeficiente de 0.0127. Estos son los coeficientes que después de realizar la prueba de hipótesis resultaron no significativos:

```
corrr::correlate(bodyfat[,1:4])
```

Correlation computed with

- Method: 'pearson'
- Missing treated using: 'pairwise.complete.obs'

A tibble: 4 × 5

	term	Density	Fat	Age	Weight
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Density	NA	-0.988	-0.278	-0.594
2	Fat	-0.988	NA	0.291	0.612
3	Age	-0.278	0.291	NA	-0.0127
4	Weight	-0.594	0.612	-0.0127	NA

Matrices de correlación utilizando ggstatplot

ggstatplot también permite crear matrices de correlación:

```
library(ggstatsplot)
```

```
ggcorrmat(
```

```
  data = bodyfat[, 8:14], ## Data Frame
```

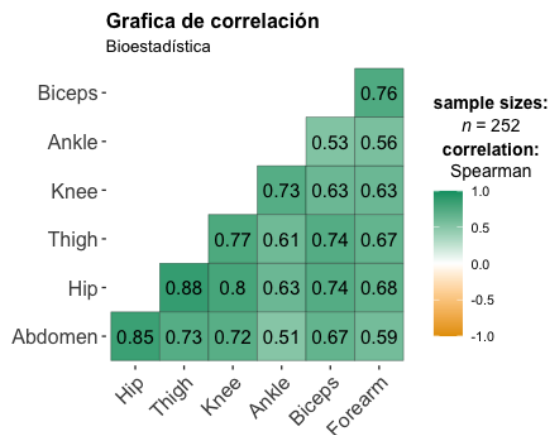
```
  type = "np", ## Paramétrica o no paramétrica
```

```
  matrix.type = "lower", ## Estructura de la gráfica
```

```
  title = "Grafica de correlación", ## Título
```

```
  subtitle = "Bioestadística" ## Subtítulo
```

```
)
```



X = non-significant at $p < 0.05$ (Adjustment: Holm)

Gráficos scatterplot

Se pueden realizar gráficos de dispersión y agregar valores y una línea de tendencia de regresión. El siguiente código muestra la correlación entre la densidad mineral y el porcentaje de grasa:

```
cor.test(Density, Fat)
```

Pearson's product-moment correlation

data: Density and Fat

t = -100.22, df = 250, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.9904570 -0.9843641

sample estimates:

cor

-0.9877824

```
plot(Density, Fat, main = "Densidad mineral vs Porcentaje de grasa",
     xlab="Densidad mineral", ylab="Porcentaje de grasa", col="cyan4", pch=20)
```

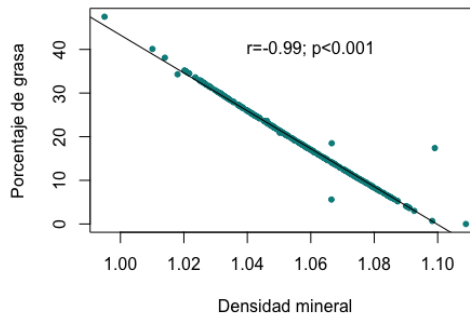
```
# Segunda capa
```

```
text(x=1.06, y=40, label="r=-0.99; p<0.001")
```

```
# Tercera capa
```

```
abline(lm(Fat~Density))
```

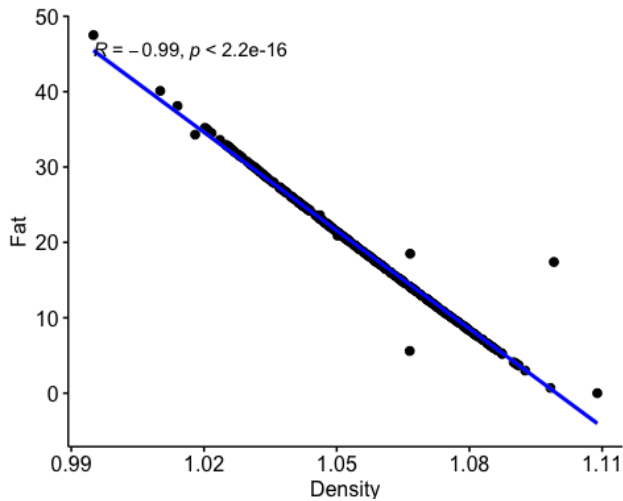
Densidad mineral vs Porcentaje de grasa



La librería `ggpubr` contiene la función `stat_cor` que permite crear la gráfica anterior y además añadir intervalos de confianza, lo cual es mucho más informático:

```
install.packages("ggpubr")
library(ggpubr)

ggscatter(bodyfat, x = "Density", y = "Fat",
  add = "reg.line", # Añadir línea de tendencia
  add.params = list(color = "blue", fill = "lightgray"), # Colores
  conf.int = TRUE # Añadir intervalos de confianza
)+ stat_cor(method = "pearson") # Añadir coeficiente de correlación
```



Ejercicios

En este capítulo no se muestran la resolución de los ejercicios, pero puede utilizar los ejemplos del capítulo para responderlos.

- ♦ **Ejercicio 21.1:** realice una gráfica de dispersión para identificar la correlación entre el porcentaje de grasa y el resto de las variables de la base `Bodyfat`.

- ♦ **Ejercicio 21.2:** realice al menos tres matrices de correlación distintas para evaluar los variables de la base de datos Bodyfat.
- ♦ **Ejercicio 21.3:** utilice la función `ggscatterstats` de la librería `ggstatplot` para realizar al menos 5 gráficos de dispersión de la base Bodyfat.

Referencias

- Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18(3), 91-93. <https://doi.org/10.1016/j.tjem.2018.08.001>
- Chang, W. (2020). *R Graphics Cookbook: Practical Recipes for Visualizing Data*. O'Reilly.
- Gatsonis, C. y Sampson, A. R. (1989). Multiple correlation: exact power and sample size calculations. *Psychological bulletin*, 106(3), 516-524. <https://doi.org/10.1037/0033-2909.106.3.516>
- Indrayan, A. y Malhotra, R. K. (2017). Relationships: Quantitative Outcome. En *Medical Biostatistics* (4ª ed., pp. 423-465). Chapman & Hall/CRC Biostatistics Series.
- Janse, R. J., Hoekstra, T., Jager, K. J., Zoccali, C., Tripepi, G., Dekker, F. W. y Van Diepen, M. (2021). Conducting correlation analysis: important limitations and pitfalls. *Clinical kidney journal*, 14(11), 2332-2337. <https://doi.org/10.1093/ckj/sfab085>
- Janse, R. J., Hoekstra, T., Jager, K. J., Zoccali, C., Tripepi, G., Dekker, F. W. y Van Diepen, M. (2021). Conducting correlation analysis: important limitations and pitfalls. *Clinical kidney journal*, 14(11), 2332-2337. <https://doi.org/10.1093/ckj/sfab085>
- Kassambara, A. (2023). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.6.0. <https://CRAN.R-project.org/package=ggpubr>
- Moscarelli, M. (2023). Correlation. En *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 93-11). Springer Cham. <https://doi.org/10.1007/978-3-031-33073-5>
- Mukaka, M. M. (2012). Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi medical journal: the journal of Medical Association of Malawi*, 24(3), 69-71.
- Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach. *Journal of Open Source Software*, 6(61), 3167. doi:10.21105/joss.03167
- R Core Team (2024). *R: A language and environment for statistical computing*.

- Revelle, W. (2024). psych: Procedures for Psychological, Psychometric, and Personality Research. Northwestern University, Evanston, Illinois. R package version 2.4.3. <https://CRAN.R-project.org/package=psych>
- Schober, P., Boer, C. y Schwarte, L. A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia and analgesia*, 126(5), 1763-1768. <https://doi.org/10.1213/ANE.0000000000002864>
- Venables, W. N. y Ripley, B. D. (2002). *Modern Applied Statistics*.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag.
- Wickham, H. (2019). *Advanced R* (2nd ed.). Chapman & Hall/CRC The R Serie.
- Zou, K. H., Tuncali, K. y Silverman, S. G. (2003). Correlation and simple linear regression. *Radiology*, 227(3), 617-622. <https://doi.org/10.1148/radiol.2273011499>

Regresión lineal

Edsaúl Emilio Pérez Guerrero

Es bien sabido que el consumo de sal tiene un efecto sobre el aumento de las cifras de tensión arterial (Bhattacharya *et al.*, 2022; Takase *et al.*, 2015). Cuando usted gráfica los datos de consumo de sal y las cifras de tensión arterial de un grupo de pacientes, puede observar una relación lineal y positiva entre estas dos variables: conforme aumenta el consumo de sal, también aumentan las cifras de la tensión arterial. Además, es posible establecer una relación matemática entre estas dos variables para describir cómo cambian las cifras de tensión arterial conforme se incrementa el consumo de sal. Incluso, con la cantidad y calidad de datos correctas, es posible estimar en un paciente determinado cuáles podrían ser sus cifras de tensión arterial, si se documenta su consumo de sal.

Las técnicas de regresión (como la que uso en el ejemplo de párrafo anterior) son una de las técnicas estadísticas mas maravillosas y utilizadas para modelar la relación entre una variable dependiente (x) y una o más variables independientes (y). En Ciencias de la Salud, Biología, Química y otras disciplinas, sus usos son diversos, por ejemplo:

- ♦ Predicción
- ♦ Identificación de factores de riesgos
- ♦ Para aislar el efecto de una sola variable
- ♦ Para identificar el efecto de varias variables sobre una dependiente
- ♦ Identificar la independencia de dos variables
- ♦ Análisis de supervivencia
- ♦ Modelos de costo-efectividad

La regresión lineal, la técnica de regresión más sencilla y que se revisará a detalle en este capítulo, se divide en dos tipos principales: la regresión lineal simple, que involucra un solo predictor (una sola variable independiente), y la regresión lineal múltiple, que involucra múltiples predictores (más de dos variables dependientes). Ambos métodos permiten no solo predecir valores de la variable dependiente, sino también cuantificar la fuerza y la naturaleza de las relaciones entre las variables. Además, en el caso de la regresión lineal múltiple es posible aislar el efecto de una sola variable o identificar el efecto de varias variables sobre nuestra variable.

A lo largo de este capítulo, revisaremos los fundamentos teóricos de la regresión lineal simple y múltiple, sus aplicaciones prácticas en el campo de la salud y cómo llevar a cabo estos análisis utilizando R. Aprenderemos a evaluar los supuestos necesarios para asegurar la validez de nuestros modelos y a interpretar correctamente los resultados obtenidos. Comenzaremos con la regresión lineal simple y después revisaremos lo relacionado con la regresión lineal múltiple.

Regresión lineal simple

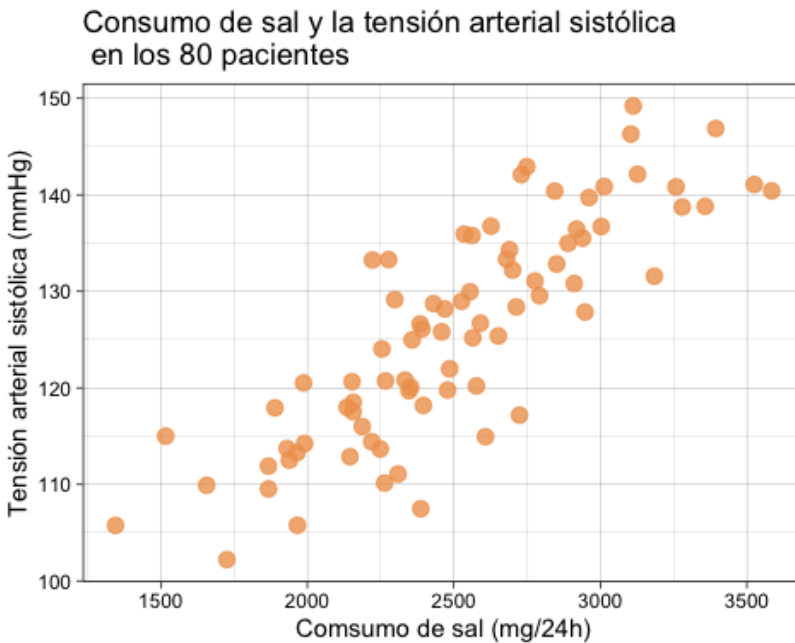
El análisis de regresión lineal simple es una técnica que se utiliza para explicar o modelar la relación entre dos variables numéricas. En esta técnica se debe diferenciar e identificar la variable dependiente (x) y la variable independiente o predictora (y). Retomaremos el ejemplo del consumo de sal sobre las cifras de tensión arterial descrito al inicio del capítulo para explicar los aspectos teóricos de la regresión lineal simple, pero trabajando con datos específicamente diseñados para este capítulo.

Un grupo de investigadores tiene como objetivo evaluar el efecto del consumo de sal diario, medido en mg/24h, sobre las cifras de tensión arterial sistólica (mmHg). Para ello, incluyen en un estudio a 80 pacientes, en quienes evalúan el consumo de sal durante 10 días y lo relacionan con la tensión arterial sistólica medida en reposo de 20 minutos a las 5 de la tarde todos los días. Estimaron los promedios de consumo de sal y de tensión arterial de los 10 días y capturaron los datos en el *Dataframe* `sal_tension_arterial.csv`.

De acuerdo a lo descrito en el estudio, dado que se busca la relación de dos variables numéricas, hay claramente una variable dependiente (presión sistólica) y una independiente (consumo de sal), y siguiendo el [diagrama para la selección de pruebas de hipótesis](#) se sugiere utilizar técnicas de regresión. Como son dos variables, la técnica de regresión es la lineal.

La [figura 22.1](#) muestra un scatterplot del consumo de sal y la tensión arterial sistólica en los 80 pacientes.

FIGURA 22.1. CONSUMO DE SAL Y LA TENSIÓN ARTERIAL SISTÓLICA EN LOS 80 PACIENTES

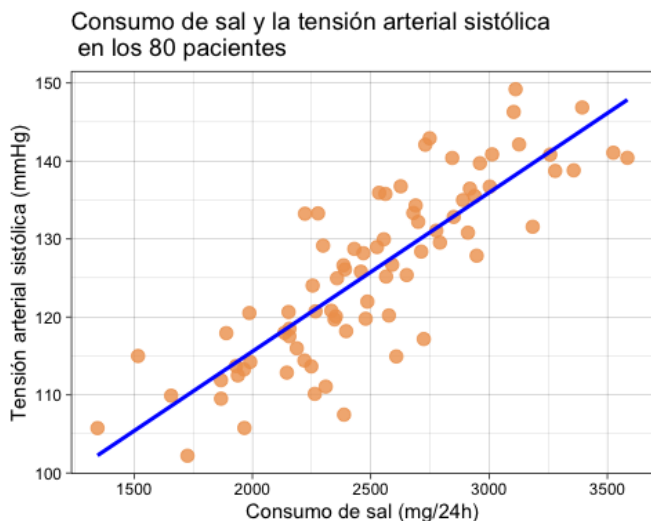


Básicamente en lo que consiste una regresión es encontrar una ecuación o modelo que describa el comportamiento de los datos. El modelo más sencillo es el que está representado por la ecuación de la recta:

$$y = \beta_0 + \beta_1 x_1$$

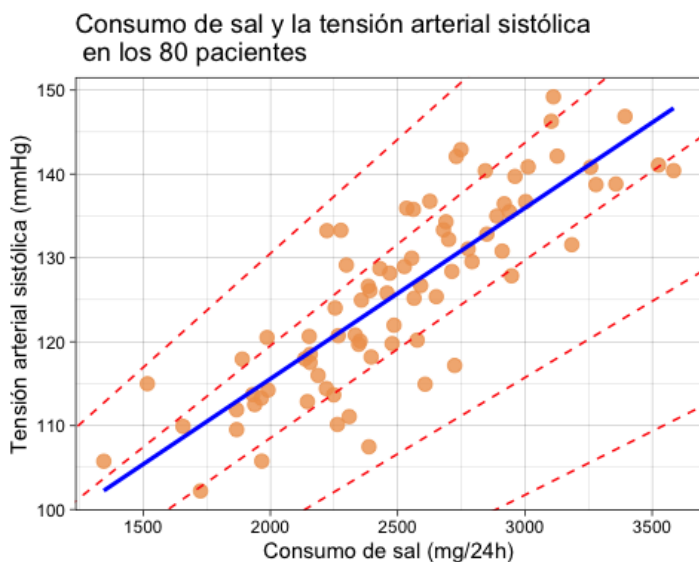
Si graficamos este modelo matemático, ahora lucirá como la [figura 22.2](#).

FIGURA 22.2. CONSUMO DE SAL Y LA TENSIÓN ARTERIAL CON LÍNEA DE REGRESIÓN



Ahora la interrogante es: ¿por qué esa línea y no las otras tantas opciones que se muestran en la [figura 22.3](#)?

FIGURA 22.3. CONSUMO DE SAL Y LA TENSIÓN ARTERIAL CON LÍNEA DE REGRESIÓN

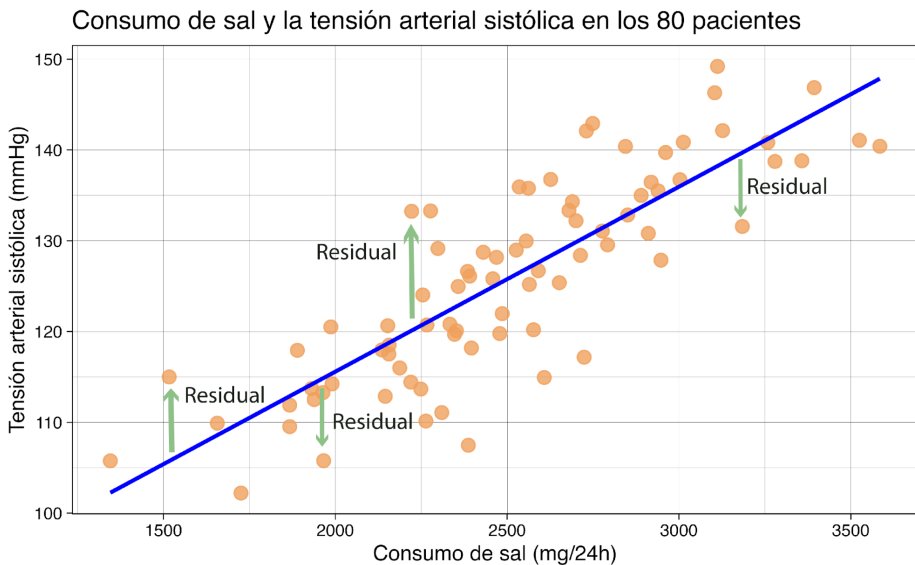


Para poder estimar la línea que mejor describe un grupo de datos se utilizan varios métodos, como el método de los mínimos cuadrados, la máxima verosimilitud, el método de los mínimos cuadrados ponderados, el método de bootstrap, entre otros. Sin embargo, el más utilizado y el que emplea R por defecto es el método de los mínimos cuadrados, que explicaremos a continuación.

Método de los mínimos cuadrados

Antes de comenzar a explicar en detalle en qué consiste el método de los mínimos cuadrados, debemos definir qué es un residual (ver [figura 22.4](#)). Este concepto se refiere a la diferencia que existe entre el valor estimado (línea azul) y el valor real (punto naranja). Precisamente, el método de los mínimos cuadrados minimiza la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo. En otras palabras, este método estima el modelo que produce los residuales más pequeños elevados al cuadrado.

FIGURA 22.4. RESIDUALES EN REGRESIÓN



El método de los mínimos cuadrados se base en la siguiente fórmula:

$$\beta = (X^T X)^{-1} X^T y$$

Donde:

- ♦ β es el vector de coeficientes estimados.
- ♦ X es la matriz de diseño que contiene las variables independientes.
- ♦ y es el vector de valores de la variable dependiente.
- ♦ X^T es la transpuesta de la matriz X .
- ♦ $(X^T X)^{-1}$ es la inversa de la matriz $X^T X$.

Componentes de la regresión

La [figura 22.5](#) muestra los principales componentes de una regresión de forma gráfica. Estos componentes son:

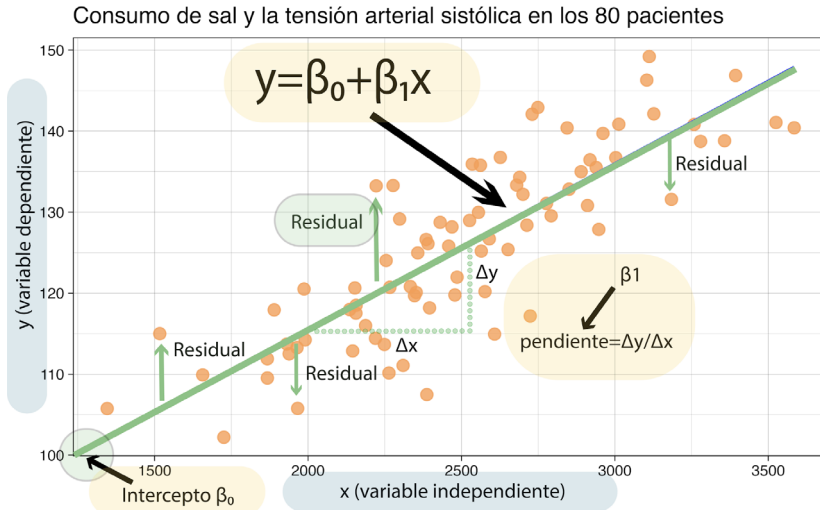
- ♦ x (variable independiente)
- ♦ y (variable dependiente)
- ♦ β_1 (pendiente): que indica el cambio en y por unidad de cambio en x
- ♦ β_0 (intercepto): que refleja dónde la línea de la recta cruza el eje de las y
- ♦ $y = \beta_0 + \beta_1 x$ (ecuación de la recta): modelo matemático que define el comportamiento lineal de mis datos
- ♦ Residual: diferencia entre el valor real y el valor predicho

La estimación de la pendiente se puede estimar de forma aproximada mediante $\Delta y / \Delta x$, sin embargo, su estimación exacta es un poco más compleja de lo que parece y se calcula utilizando la siguiente fórmula:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Por su parte, para estimar el valor del intercepto se utiliza la siguiente fórmula:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

FIGURA 22.5. COMPONENTES DE LA REGRESIÓN LINEAL

Ahora que ya conocemos algunos de los aspectos teóricos, vamos realizar una regresión lineal simple en R para identificar el efecto del consumo de sal (mg/24h) sobre la tensión arterial sistólica (mmHg).

Regresión lineal simple R

- ♦ **Ejemplo 22.1:** para este ejemplo vamos a utilizar la base de datos [sal_arte-rial.csv](#) contiene datos del estudio de los 80 pacientes en los que se estudio el consumo de sal y tensión arterial. Esta base tiene 6 variables, pero de momento nos enfocaremos en dos: Consumo_Sal que contiene el promedio de sal (mg/24h) y Tensión_Arterial que contiene las cifras de tensión arterial (mmHg).

Antes de realizar una regresión le aconsejamos se pregunte lo siguiente:

- ♦ ¿Para cumplir con los objetivos de mi estudio requiero realizar una regresión?
- ♦ ¿Cuál es el objetivo de mi regresión?
- ♦ ¿Mis variables son adecuadas para una regresión? ¿Son dos variables numéricas con distribución normal? (la normalidad de los datos es recomendable, pero no necesaria)
- ♦ ¿Hay indicios en mis análisis previos que requieran el uso de una regresión?

Si no puede responder estas preguntas, no haga una regresión.

Las respuestas a las dos primeras preguntas ya las hemos respondido. También sabemos que ambas variables, de acuerdo a lo descrito, son dos variables numéricas. Haría falta responder las características de la distribución de estas variables. Vamos a poner manos a la obra.

Importe la base:

```
df ← read.csv("Bases/sal_tension_arterial.csv")
```

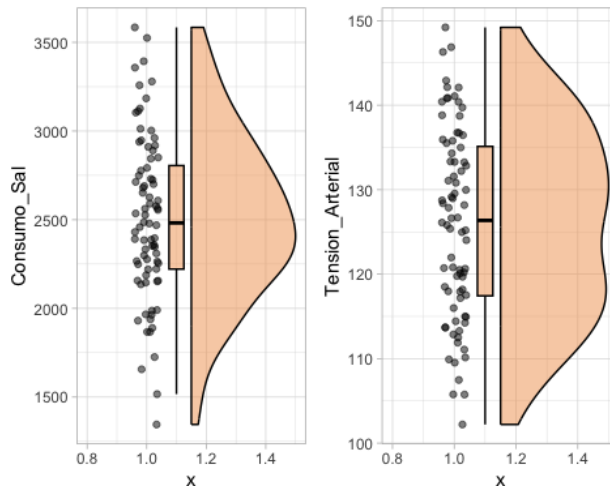
Note cómo para simplificar las cosas llamamos a la base df. Si su base se guardó en su carpeta de “Bases” no tendría problema en utilizar el código anterior.

Vamos a realizar un gráfico de raincloud para conocer la distribución de mis variables:

```
library(ggplot2)
library(ggrain) # Par el raincloud
library(gridExtra) # Para la arreglo de gráficas
df|>
  ggplot(mapping = aes(x=1, y=Consumo_Sal))+
  geom_rain(fill="#F0A05B", alpha=0.5, col="black" )+
  theme_light()→ g1 #Guardamos la gráfica en g1

df|>
  ggplot(mapping = aes(x=1, y=Tension_Arterial))+
  geom_rain(fill="#F0A05B", alpha=0.5, col="black" )+
  theme_light()→ g2 #Guardamos la gráfica en g1

grid.arrange(g1,g2, ncol=2) # Arreglo en dos columnas
```



Las gráficas muestran que los datos siguen una distribución normal. Aun así siempre puede comprobarlos con alguna prueba estadística u otra gráficas.

En realidad, en la regresión lineal simple, no es estrictamente necesario que las variables (dependiente e independiente) sigan una distribución normal. El supuesto de normalidad se aplica principalmente a los residuos del modelo (como veremos más adelante), no a las variables en sí. Sin embargo, es altamente recomendable que sigan una distribución normal.

Ahora, algo indispensable para realizar una regresión es demostrar linealidad de las variables estudiadas. En la [figura 22.1](#) claramente podemos el consumo de sal y la tensión arterial presentan una relación lineal, esta relación también la puede evaluar con estimando el coeficiente de correlación de Pearson. El código utilizado para la construcción de la [figura 22.1](#) fue:

```
library(ggplot2)

df <- read.csv(file = "Bases/sal_tension_arterial.csv")

df|>
  ggplot(mapping = aes(x=Consumo_Sal, y=Tension_Arterial))+
  geom_point(col= "#F0A05B", alpha = 0.8, size = 3)+
```

```
theme_linedraw()+labs(x="Consumo de sal (mg/24h)", y= "Tensión arterial sistólica (mmHg)", title = "Consumo de sal y la tensión arterial sistólica en los 80 pacientes")
```

Para hacer una regresión lineal simple en R utilice la función `lm()`, la cual se alimenta de dos vectores numéricos que deben ser alimentados en forma de formula con el signo `~`. A este modelo de regresión después le debemos aplicar un `summary()` para poder visualizarlo correctamente. La estructura general es `lm(y ~ x, data = datos)`:

```
lm(Tension_Arterial~Consumo_Sal, data = df)|>
summary()
```

Call:

```
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.989	-3.424	-0.313	3.531	13.142

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	74.839105	3.861251	19.38	<2e-16 ***
Consumo_Sal	0.020370	0.001513	13.46	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom

Multiple R-squared: 0.699, Adjusted R-squared: 0.6952

F-statistic: 181.1 on 1 and 78 DF, p-value: < 2.2e-16

Lo primero que observamos en estos resultados es la ecuación de la recta: `lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)`. La interpretación de estos resultados requiere de la revisión de otros conceptos y de una

descripción más detallada. De momento identifique el valor de 0.020370, el cual nos dice que por cada mg/24H que se aumente el consumo de sal, la tensión arterial aumenta en promedio 0.020370 mmHg. Interesante, ¿no lo cree?

Normalidad de los residuales

Lo primero que vamos a explicar de nuestros resultados son los residuales que se resaltan en la [figura 22.6](#).

FIGURA 22.6. RESULTADOS DE LA REGRESIÓN. RESIDUALES

```
Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)
```

Residuals:				
Min	1Q	Median	3Q	Max
-15.989	-3.424	-0.313	3.531	13.142

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal   0.020370   0.001513   13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16

```

En una regresión, los residuales deben seguir una distribución normal. En la [figura 22.6](#) vemos que los residuales pueden variar desde -15.98 hasta 13.412. Evaluar la magnitud de los residuales solo tomando en cuenta los valores mínimos y máximos puede ser un error, ya que estos están en relación con las unidades de las variables.

Lo importante es que la mediana sea cercana a 0, y que el tercer cuartil y el primer cuartil se ubiquen a la misma distancia de la mediana. En este ejemplo, ambos están aproximadamente a 3 unidades. También debemos visualizar que la distancia entre el valor máximo y el tercer cuartil sea más o menos la misma que la distancia entre el valor mínimo y el primer cuartil. Según los resultados mostrados, los residuales tienen una distribución normal.

Es importante que los residuales presenten esta forma porque garantiza, en cierta medida, que la regresión se “equivoca” de la misma manera a lo largo de todos los valores de la variable. De lo contrario, los residuales sin una distribución normal reflejan una regresión que se equivoca de manera diferente con unos valores (por ejemplo, valores pequeños) y de otra forma con otros valores (por ejemplo, valores más altos).

Si lo requiere, siempre puede evaluar la normalidad de los residuales con una prueba de hipótesis o gráficas. Solo tendría que guardar su regresión en un objeto y luego acceder a los residuales, como lo ejemplifica el siguiente código:

```
modelo <- lm(Tension_Arterial~Consumo_Sal, data = df)
nortest::lillie.test(modelo$residuals)
```

Lilliefors (Kolmogorov-Smirnov) normality test

```
data:  modelo$residuals
D = 0.061958, p-value = 0.6297
```

Coefficientes β

En una regresión lineal simple hay dos coeficientes que son muy útiles para la interpretación correcta de los resultados. En la [figura 22.5](#) podemos ver que en la gráfica hay un intercepto y una pendiente, estos dos valores son coeficientes β que son representados como β_0 y β_1 , respectivamente.

La [figura 22.7](#) resalta los coeficientes β_0 y β_1 y algunas medidas asociadas a ellos.

FIGURA 22.7. RESULTADOS DE LA REGRESIÓN. COEFICIENTES

```
Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-15.989  -3.424  -0.313   3.531  13.142

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal   0.020370   0.001513   13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16
```

El intercepto (β_0) es valor en el que la línea de la regresión cruza el eje de las . En otras palabras, sería el valor de la presión arterial cuando el consumo de sal es cero, en este caso el intercepto es de 74.839105 mmHg, valor en naranja en la [figura 22.8](#). Aunque este valor no siempre tiene una interpretación práctica (ya que un consumo de sal de 0 podría no ser realista), es necesario para la ecuación de la recta. Posteriormente, también podemos ver el valor del error estándar, marcado en verde en la [figura 22.8](#). Este valor de 3.86, es útil para conocer la precisión en la estimación de β_0 , un valor grande indica poca precisión y un valor pequeño buena precisión. Sin embargo, siempre hay que tener en cuenta las unidades de las variables: no sería lo mismo un error estándar de 3.861251 kg que un error estándar de 3.861251 gramos.

En azul, en la [figura 22.8](#) se muestran los valores de t para una prueba de hipótesis con $H_0: \beta_0 = 0$ para el valor del intercepto.

Los valores de t se estiman mediante:

$$t = \frac{\hat{\beta}_i}{SE(\hat{\beta}_i)}$$

Donde:

- ♦ $\hat{\beta}_i$ es el valor estimado del coeficiente
- ♦ SE es el error estándar estimado para el coeficiente

En este caso el valor de t es muy alto, lo cual indica que el β_0 es diferente 0. Finalmente, en amarillo tenemos el valor p para t .

FIGURA 22.8. RESULTADOS DE LA REGRESIÓN. INTERCEPTO

```
Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-15.989  -3.424  -0.313   3.531  13.142

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal  0.020370   0.001513   13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16
```

Ahora la pendiente (β_1) es el valor que nos refleja que tanto está influyendo x en y . Entre más grande sea el valor de β_1 más grande será su influencia y la pendiente será más pronunciada. Aunque siempre tiene que tomar en cuenta las unidades de sus variables. El coeficiente β_1 de 0.020370 significa que por cada incremento de 1 mg/24 h en el consumo de sal, la tensión arterial aumenta en promedio 0.020370 mmHg (valor resaltado en naranja en la [figura 22.9](#)). Este coeficiente presenta un error estándar de 0.001513 y un valor de t de 13.46 para la prueba de hipótesis (valores resaltados en verde y azul en la [figura 22.9](#), respectivamente). Además, podemos ver que el coeficiente es diferente de 0, valor de $p < 2e-16$.

FIGURA 22.9. RESULTADOS DE LA REGRESIÓN. PENDIENTE

```
Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-15.989  -3.424  -0.313   3.531  13.142

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal  0.020370   0.001513   13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16
```


¿Por qué el coeficiente β_1 debe ser diferente de 0? Un coeficiente de 0 indicaría que no hay cambios en y que se deban a x y por lo tanto estas variables no estarían relacionadas. De ahí la importancia de demostrar que β_1 es diferente de 0.

Por otro lado, hay que tener en cuenta que un valor de β_1 puede ser tanto positivo como negativo. En el ejemplo que hemos estado trabajando, β_1 fue positivo, por lo tanto, interpretamos que al aumentar x (el consumo de sal) aumenta y (la tensión arterial). Un coeficiente β_1 negativo se interpreta a la inversa, es decir, que y disminuye conforme aumenta x .

Intervalos de confianza para los coeficientes β

Un resultado que no se muestra directamente al hacer `summary()` de nuestro modelo y que es sumamente valioso es la estimación de los intervalos de confianza para los coeficientes. Nuestro objetivo es identificar si los intervalos incluyen el 0; si no lo incluyen, podríamos concluir que nuestro coeficiente es significativamente diferente de 0. De lo contrario, tendríamos que aceptar que el coeficiente podría ser 0.

En R podemos utilizar el siguiente código para la estimación de los intervalos de confianza de los coeficientes:

```
lm(Tension_Arterial~Consumo_Sal, data = df)|>
  confint()
           2.5 %          97.5 %
(Intercept) 67.15194384    82.52626615
Consumo_Sal  0.01735684    0.02338306
```

Para β_1 los valores van de 0.017 a 0.023, no incluyen al 0. Por lo tanto, podríamos asegurar que nuestro valor de β_1 es diferente de 0 con 95% de confianza, conclusión que es similar al de la prueba t . Sin embargo, recordemos que los intervalos de confianza ofrecen mayor información sobre la variabilidad de los resultados.

Además, imagine que sus intervalos van de 0.00001 a 0.00002. ¿Concluiría con la misma seguridad? Aunque el intervalo no incluye al 0, la magnitud de los coeficientes es extremadamente pequeña, lo que podría llevar a cuestionar la relevancia práctica del efecto. En este caso, aunque estadísticamente significativo, el efecto podría no ser de importancia práctica.

Valor de R^2

Otro valor que nos proporciona R al hacer `summary()` del modelo es el valor de R^2 , resaltado en la [figura 22.10](#). Este valor, también conocido como coeficiente de determinación, es de 0.699, lo que significa que aproximadamente el 69.9% de la variabilidad en la tensión arterial puede ser explicada por el consumo de sal.

El R^2 puede tomar valores de 0 a 1. Difícilmente, es más, es casi imposible en el contexto de las ciencias de la salud obtener un valor de 0.99 o de 1. Esto significaría que lo que pasa con una variable puede ser explicado en su totalidad por otra variable. Pero al tratarse de organismos vivos, debemos recordar que hay múltiples factores influyendo en la variable dependiente. En otras palabras, sabemos que en la tensión arterial influyen otras situaciones y no solo el consumo de sal.

FIGURA 22.10. RESULTADOS DE LA REGRESIÓN. COEFICIENTE DE DETERMINACIÓN

```
Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-15.989  -3.424  -0.313   3.531  13.142

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal  0.020370   0.001513    13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16
```

Por el momento no es necesario concentrarse en el valor de R^2 , del cual hablaremos en la [“Interpretación del modelo”](#).

ANOVA de la regresión

La parte final de los resultados corresponde a un ANOVA, estos resultados se encuentran resaltados en la [figura 22.11](#).

FIGURA 22.11. RESULTADOS DE LA REGRESIÓN. ANOVA

```

Call:
lm(formula = Tension_Arterial ~ Consumo_Sal, data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-15.989  -3.424  -0.313   3.531  13.142

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  74.839105   3.861251   19.38  <2e-16 ***
Consumo_Sal   0.020370   0.001513    13.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.213 on 78 degrees of freedom
Multiple R-squared:  0.699,    Adjusted R-squared:  0.6952
F-statistic: 181.1 on 1 and 78 DF,  p-value: < 2.2e-16

```

En el contexto de la regresión, un ANOVA ayuda a determinar si el modelo pudiera llegar a ser útil, no necesariamente que el modelo sea válido. El ANOVA determina si existe una relación lineal significativa entre las variables independientes y la variable dependiente.

La H_0 para este ANOVA es que todos los coeficientes de regresión (excepto el intercepto) son iguales a 0. Recuerde que coeficientes de 0 implican que las variables independientes no explican ninguna variabilidad en la variable dependiente.

El ANOVA en el contexto de la regresión lineal se utiliza para evaluar la significancia global del modelo. En otras palabras, el ANOVA responde a la pregunta: ¿el modelo de regresión en su conjunto es significativo? La prueba solo evalúa los coeficientes de forma independiente. Quizá en una regresión lineal simple esta diferencia no es tan evidente, pero en una regresión lineal múltiple se hará más notoria.

La [tabla 22.1](#) muestra la tabla de ANOVA construida para una regresión. En el contexto de nuestra regresión lineal sobre el consumo de sal y las cifras de tensión arterial, hay dos fuentes de variación: una es la propia regresión y la otra es el error. La primera fuente de variación (regresión) dice cuánta variabilidad en la tensión arterial es explicada por el consumo de sal. La segunda, la variabilidad del error, es la variabilidad en la tensión arterial que no puede ser explicada por el consumo de sal. En otras palabras, después de estimar la suma de cuadrados y los cuadrados medios, podemos calcular un valor de F. Cuanto

mayor sea ese valor de F, más significará que la regresión explica más que el error. Es decir, que el consumo de sal explica más de lo que pasa en la tensión arterial que el propio error (lo que no explica).

Para nuestro ejemplo obtuvimos un valor de F de 181.1, un valor extremadamente grande. Por lo tanto, podríamos concluir que nuestra regresión explica más que el error.

TABLA 22.1. TABLA DE ANOVA PARA LA REGRESIÓN

FUENTE DE VARIACIÓN	DF	SUMA DE CUADRADOS	CUADRADOS MEDIOS	F
Regresión	1	$\text{RegSS} = \sum (\hat{y}_i - \bar{y})^2$	$\text{RegMS} = \frac{\sum (\hat{y}_i - \bar{y})^2}{1}$	$F = \frac{\text{RegMS}}{\text{MSE}}$
Error	n-2	$\text{SSE} = \sum (y_i - \hat{y}_i)^2$	$\text{MSE} = \frac{\sum (y_i - \hat{y}_i)^2}{n - 2}$	
Total	n-1	$\text{SST} = \sum (y_i - \bar{y})^2$		

Gráficos diagnóstico para la regresión

Sí, aún no hemos terminado con nuestra regresión. Hasta aquí solo hemos interpretado todos los coeficientes, evaluado la normalidad de los residuales y demostrado que nuestro modelo de regresión puede servir para algo. Ahora, debemos terminar de validarlo. Para ello vamos a utilizar gráficas diagnósticas:

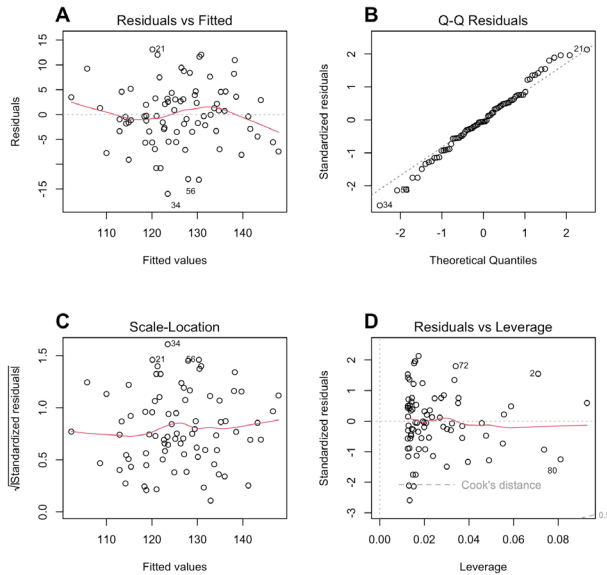
- ♦ Gráfico de residuos vs. valores ajustados: permite evaluar la homoscedasticidad de las varianzas.
- ♦ Gráfico Q-Q de residuos: permite evaluar la normalidad de los residuos.
- ♦ Gráfico de escala-localización (Spread-Location Plot): para evaluar homoscedasticidad de las varianzas.
- ♦ Residuos vs. Leverage: identificación de valores atípicos.

Para obtener estos gráficos solo debemos de utilizar la función `plot()` y aplicarla la modelo:

```
lm(Tension_Arterial~Consumo_Sal, data = df)|>
plot()
```

Si utiliza el código anterior verá 4 gráficas, que para fines didácticos se juntaron y se nombraron de la A a la D en la [figura 22.12](#).

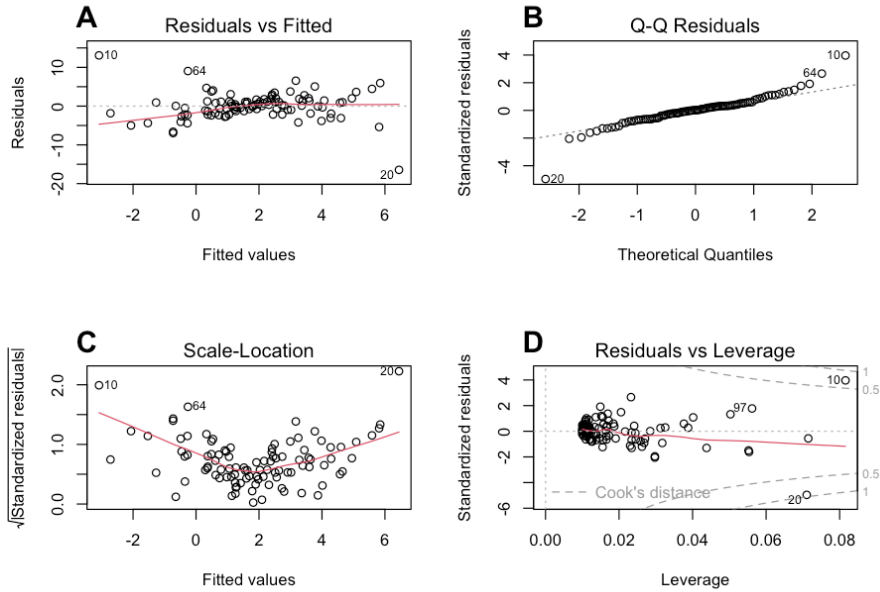
FIGURA 22.12. GRÁFICOS DIAGNÓSTICO PARA LA REGRESIÓN



- ♦ En el apartado A de la [figura 22.12](#), usted debería identificar patrones o formas en los residuos. Un patrón puede indicar una relación no lineal o heteroscedasticidad. Un modelo válido no debería tener estos patrones.
- ♦ En el apartado B de la [figura 22.12](#), debería identificar datos que se desvíen de la recta. Las desviaciones sistemáticas de la línea recta indican que los residuos no son normales.
- ♦ En el apartado C de la [figura 22.12](#), busque una dispersión uniforme de los puntos. No debe haber patrones visibles.
- ♦ En el apartado D de la [figura 22.12](#), si se observa la presencia de uno o más puntos fuera de la línea punteada de la distancia de Cook, su modelo no sería válido, ya que estaría observando valores atípicos. En la regresión, un dato atípico es aquel que tiene una influencia desproporcionada en el modelo, explicando más que el resto de los puntos, incluso más que la propia regresión.

La [figura 22.13](#) muestra un ejemplo de un modelo de regresión que no cumple con los supuestos evaluados en estas gráficas.

FIGURA 22.13. GRÁFICOS DIAGNÓSTICO PARA LA REGRESIÓN DE UN MODELO NO VÁLIDO



Resumen de los supuestos

Se hace una recopilación de los supuestos evaluados en el ejemplo de la regresión.

1. **Linealidad:** debe de evidenciarse una relación lineal entre las variables antes de realizar un modelo de regresión lineal.
 - » Utilice gráficos de dispersión.
 - » Coeficiente de correlación de Pearson.
2. **Independencia de los datos:** este concepto no se había explicado antes, sin embargo, es fácil de comprender. En una regresión las observaciones individuales no deben estar relacionadas o influenciadas entre sí.

- » Cuando se recolectan los datos debe de asegurarse la no relación de las variables, por ejemplo, no se debe de incluir mediciones repetidas del mismo sujeto.
- » En los gráficos de residuales también se hace evidente esta relación.
- 3. Homoscedasticidad: los residuos tienen varianza constante a lo largo de los valores predichos.
 - » Utilice gráficos de diagnóstico para comprobar la homoscedasticidad.
- 4. Normalidad de los residuos: los residuos del modelo deben de estar distribuidos normalmente.
 - » Histogramas de los residuos.
 - » Gráficos Q-Q de los residuos.
 - » Pruebas de normalidad.

Recuerde que el ANOVA y la prueba F nos ayudan a verificar que el modelo de regresión en su conjunto es significativo y si los coeficientes β son diferentes de 0 respectivamente.

Regresión lineal múltiple

Una de las principales ventajas de las regresiones es que podemos conocer el efecto de diversas variables predictoras sobre una variable dependiente, o en su defecto, permite controlar y ajustar por variables asociadas a nuestra variable dependiente. Para poder realizar estas dos situaciones con una regresión, debemos enfocarnos en un modelo más complejo y realizar una regresión lineal múltiple. La regresión múltiple pertenece a la estadística multivariable, donde podemos evaluar el efecto de múltiples variables sobre una sola (y).

La regresión lineal múltiple en su forma más sencilla sigue la siguiente fórmula:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon$$

Donde:

- ♦ $\beta_0, \beta_1, \beta_2 \dots + \beta_k$ son los coeficientes de la regresión para el intercepto y para cada una de las variables.
- ♦ x_1, x_2, x_k son las variables independientes.

Variables Dummy

En la regresión lineal múltiple podemos ajustar por múltiples variables, y estas variables no necesariamente tienen que ser numéricas. Es decir, podemos ajustar tanto por variables numéricas como por variables cualitativas. Usted se preguntará: ¿cómo podemos evaluar el efecto de una variable dicotómica o una variable de tres niveles sobre una variable numérica?

Cuando la variable es dicotómica, no hay mayor problema: se asigna 0 a un nivel y 1 al otro nivel. La situación se complica con una variable de tres niveles. Por ejemplo, si una de sus variables fuera el antecedente de tabaquismo y los niveles fueran: No fumador, Ex fumador y Fumador, ¿a quién le asignaría el 0 y el 1? Para poder lidiar con esta situación en las regresiones múltiples, se crean variables Dummy.

Las variables Dummy están formadas por $k - 1$ de las k categorías de la variable. Algo que se debe tener en cuenta cuando se crean las variables Dummy es que estas solo tienen sentido cuando se analizan en conjunto, nunca deben analizarse por separado. Fuera de un modelo de regresión, las variables Dummy no tienen sentido.

Supongamos que queremos realizar un modelo en el que se incluya una variable relacionada al hábito de fumar. Esta variable contiene los siguientes niveles:

- ♦ No fumador
- ♦ Exfumador
- ♦ Fumador

La creación y por tanto la introducción de esta variable al modelo requerirá de la creación de dos variables Dummy ($\cdot$), donde 0 es la ausencia del evento. La siguiente tabla muestra la creación de estas variables:

TABLA 22.2. EJEMPLO DE VARIABLE DUMMY

	X_1	X_2
No fumador	0	0
Exfumador	1	0
Fumador	0	1

En la [tabla 22.2](#) podemos identificar que uno de los niveles sirve para comparar a los otros dos y funciona como un estado basal. Podemos comparar la ausencia (no fumador) con la variable x_1 (ex fumador) y la ausencia del evento (no fumador) con x_2 (fumador). Esta explicación la podemos trasladar a los coeficientes de la regresión de ahí que se hayan explicado.

Regresión lineal múltiple R

Cuando importó la base de datos [sal_tension_arterial.csv](#) muy probablemente se percató que este *Dataframe* contiene más variables. Además, en el consumo de sal y la tensión arterial, se encuentran las siguientes variables: Comorbilidades que indica la presencia de comorbilidades en los pacientes, Fuma que indica si el paciente no fumaba, fumó o fuma y la variable Edad.

Lo primero es adaptar la base:

```
df <- read.csv("Bases/sal_tension_arterial.csv", stringsAsFactors =
T)
df$Comorbilidades <- factor(df$Comorbilidades,
                           labels=c("No", "Si"))
df$Fuma <- factor(df$Fuma,
                 labels = c("No fuma", "Exfumador", "Fumador"))

str(df)
'data.frame': 80 obs. of 6 variables:
 $ Consumo_Sal      : num 1345 1517 1657 1726 1867 ...
 $ Tension_Arterial: num 106 115 110 102 112 ...
 $ Sexo             : Factor w/ 2 levels "Hombre","Mujer": 2 2 2 2 2 1 2 2 2 1 ...
 $ Comorbilidades  : Factor w/ 2 levels "No","Si": 1 2 2 1 2 1 2 2 2 2 ...
```

```
$ Fuma : Factor w/ 3 levels "No fuma","Exfumador", ..: 1 1 1 1 1 1 1 1 1 1 ...
$ Edad : int 72 23 59 71 79 18 34 32 20 19 ...
head(df)
```

	Consumo_Sal	Tension_Arterial	Sexo	Comorbilidades	Fuma	Edad
1	1345.416	105.7535	Mujer	No	No fuma	72
2	1516.691	114.9969	Mujer	Si	No fuma	23
3	1656.653	109.9079	Mujer	Si	No fuma	59
4	1725.624	102.2093	Mujer	No	No fuma	71
5	1867.302	111.8932	Mujer	Si	No fuma	79
6	1867.469	109.5253	Hombre	No	No fuma	18

En las regresiones es sumamente importante que se respeten los niveles de los factores. Ahora vamos a proceder a realizar nuestra regresión múltiple. Cada variable simplemente se añade con el símbolo +:

```
lm(Tension_Arterial~Consumo_Sal+Comorbilidades+Fuma+Edad,
   data = df)|>
summary()
```

Call:

```
lm(formula = Tension_Arterial ~ Consumo_Sal + Comorbilidades +
    Fuma + Edad, data = df)
```

Residuals:

Min	1Q	Median	3Q	Max
-12.387	-3.113	-0.741	3.223	12.627

Coefficients:

	Estimate	Std. Error	t	value	Pr(> t)
(Intercept)	79.73327	5.39774	14.772	< 2e-16	***
Consumo_Sal	0.01534	0.00239	6.418	1.17e-08	***
ComorbilidadesSi	5.91012	1.19555	4.943	4.65e-06	***
FumaExfumador	2.71535	1.80301	1.506	0.13632	

```
FumaFumador      7.10519      2.50541  2.836  0.00589  **
Edad              0.01529      0.03051  0.501  0.61770
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.266 on 74 degrees of freedom
 Multiple R-squared: 0.7949, Adjusted R-squared: 0.781
 F-statistic: 57.35 on 5 and 74 DF, p-value: < 2.2e-16
 # Intervalos confianz

```
lm(Tension_Arterial~Consumo_Sal+Comorbilidades+Fuma+Edad,
   data = df)|>
  confint())|>
  round(4)
```

	2.5 %	97.5 %
(Intercept)	68.9780	90.4885
Consumo_Sal	0.0106	0.0201
ComorbilidadesSi	3.5279	8.2923
FumaExfumador	-0.8772	6.3079
FumaFumador	2.1131	12.0973
Edad	-0.0455	0.0761

Este es un resumen del modelo:

- ♦ **Intercepto:** 79.73327 (error estándar: 5.39774, $p < 2e-16$)
- ♦ **Consumo de sal:** 0.01534 (error estándar: 0.00239, $p = 1.17e-08$)
- ♦ **Comorbilidades (Sí):** 5.91012 (error estándar: 1.19555, $p = 4.65e-06$)
- ♦ **Exfumador:** 2.71535 (error estándar: 1.80301, $p = 0.13632$)
- ♦ **Fumador:** 7.10519 (error estándar: 2.50541, $p = 0.00589$)
- ♦ **Edad:** 0.01529 (error estándar: 0.03051, $p = 0.61770$)

Interpretación del modelo

Nos vamos a centrar en la interpretación de los coeficientes, el ANOVA, los residuales y la prueba que ya fue explicado a detalle.

1. Intercepción:

- » El valor del intercepto (79.73327) representa la tensión arterial esperada cuando todas las variables independientes son 0. Este valor es diferente de 0 ($p < 2e-16$; con intervalos al 95% de 68.9780 a 90.4885).

2. Consumo de sal:

- » El coeficiente de consumo de sal (0.01534; con intervalos al 95% de 0.0106 a 0.0201) es significativo ($p = 1.17e-08$). Esto sugiere que por cada incremento de 1 mg/24h en el consumo de sal, la tensión arterial aumenta en promedio 0.01534 mmHg, manteniendo las demás variables constantes.

3. Comorbilidades:

- » El coeficiente para comorbilidades (5.91012; con intervalos al 95% de 3.5279 a 8.2923) es significativo ($p = 4.65e-06$). Esto indica que tener comorbilidades aumenta la tensión arterial en promedio 5.91012 mmHg, manteniendo las demás variables constantes.

4. Exfumador:

- » El coeficiente para exfumadores (2.71535; con intervalos al 95% de -0.8772 a 6.3079) no es significativo ($p = 0.13632$). Note además cómo los intervalos de confianza incluyen el 0. Esto sugiere que, en este modelo, ser exfumador no tiene un efecto estadísticamente significativo en la tensión arterial, comparado con no fumador.

5. Fumador:

- » El coeficiente para fumadores (7.10519) es significativo ($p = 0.00589$; con intervalos al 95% de 2.1131 a 12.0973). Esto sugiere que ser fumador aumenta la tensión arterial en promedio 7.10519 mmHg, comparado con no fumador, manteniendo las demás variables constantes.

6. Edad:

- » El coeficiente para la edad (0.01529) no es significativo ($p = 0.61770$; con intervalos al 95% de -0.0455 a 0.0761). Esto sugiere que, en este modelo, la edad no tiene un efecto estadísticamente significativo en la tensión arterial.

Conclusión

El modelo de regresión lineal múltiple muestra que el consumo de sal, la presencia de comorbilidades y ser fumador tienen efectos significativos en la tensión arterial. Sin embargo, ser exfumador y la edad no parecen tener un impacto estadísticamente significativo en este modelo. El modelo explica aproximadamente el 79.49% de la variabilidad en la tensión arterial (R^2 múltiple = 0.7949).

Llegó el tiempo de definir a R^2 ajustado. El valor de R^2 ajustado es una medida de la bondad del ajuste de un modelo de regresión lineal que ha sido ajustada por el número de predictores en el modelo. A diferencia del R^2 simple, que puede aumentar simplemente al agregar más variables al modelo, el R^2 ajustado penaliza la inclusión de variables adicionales que no mejoran significativamente el modelo. Por lo tanto, es una medida más precisa de la calidad del ajuste del modelo, especialmente cuando se comparan modelos con diferentes números de predictores.

El valor de R^2 ajustado se calcula como:

$$R^2_{\text{ajustado}} = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right)$$

Donde: n es el número total de observaciones y p es el número de predictores del modelo.

Supuestos para la aplicación del modelo de regresión lineal múltiple

En el caso de la regresión lineal múltiple, debemos cumplir con los mismos supuestos que se revisaron en el tema de regresión lineal simple.

Sin embargo, un punto importante que se debe tomar en cuenta es que las variables no deben presentar colinealidad, es decir, no deben estar autocorrelacionadas entre sí. Dos variables independientes altamente correlacionadas inflarán los coeficientes y, por tanto, distorsionarán la realidad de nuestro modelo.

La colinealidad (o multicolinealidad, en el contexto de la regresión múltiple) se refiere a la situación en la cual dos o más variables predictoras en un modelo de regresión están altamente correlacionadas, es decir, tienen una relación alta entre sí. En los modelos de regresión, una alta colinealidad puede producir un error en la estimación de los coeficientes β .

Existen dos formas de evaluar la colinealidad. La primera es realizando una matriz de correlación, y la segunda es evaluando el factor de inflación de la varianza (VIF). En el caso de realizar una matriz de correlación, puede proceder conforme a lo revisado en el [capítulo 21](#).

El VIF se define como la proporción de variabilidad de la variable x_1 que es explicada por el resto de las variables predictoras del modelo. Cuando el VIF crece, también lo hace la varianza del coeficiente de regresión, y el modelo se vuelve inestable. Si los valores del VIF son grandes, será un indicador de la existencia de multicolinealidad y por tanto de un modelo no válido, en el que los coeficientes β se estimaron de forma incorrecta.

Para evaluar el VIF puede utilizar la función `vif()` de la librería `car`:

```
lm(Tension_Arterial~Consumo_Sal+Comorbilidades+Fuma+Edad,
  data = df)|>car::vif()
```

	GVIF	Df	GVIF^(1/(2*Df))
Consumo_Sal	3.472193	1	1.863382
Comorbilidades	1.025121	1	1.012483
Fuma	3.509892	2	1.368748
Edad	1.018282	1	1.009099

Aunque los puntos de corte pueden variar, se considera un valor de 5 indica una alta correlación entre la variable predictora y las demás variables predictoras, lo que sugiere problemas de multicolinealidad y por tanto un modelo no válido.

Aunque no es tan sencillo, si se tiene un valor alto de VIF deberán de eliminarse variables, transformar o elegir otra combinación de variables. Hacer regresiones a veces es un arte.

Selección de variables para un modelo de regresión lineal múltiple

La selección de variables que serán evaluadas es un tema complejo y no debe tomarse a la ligera. Si selecciona variables en exceso, su valor de R^2 estará sobreestimado; en caso contrario, una selección reducida podría provocar una pérdida de información.

Nuestra recomendación es la siguiente:

- ♦ Evalúe variables con plausibilidad biológica.
- ♦ Una regresión es un análisis que generalmente va precedido de otras pruebas de hipótesis, por lo que este análisis previo puede ayudarle a seleccionar sus variables.
- ♦ Defina claramente sus objetivos.

El mejor modelo es el que tiene las variables y la cantidad correcta de ellas. Hacer regresiones a veces es un arte.

Ejercicios

- ♦ **Ejercicio 22.1:** investigue en qué consiste los algoritmos para la selección de variables en los modelos de regresión: Forward method, Backward method y Stepwise method.
- ♦ **Ejercicio 22.2:** las capacidades pulmonares se refieren a los distintos volúmenes de aire característicos en la respiración humana. Un pulmón humano puede almacenar alrededor de 6 litros de aire en su interior, pero una cantidad significativamente menor es la que se inhala y exhala durante la respiración.

La base de datos [LungCapData](#) contiene datos de una serie de pacientes sobre la capacidad de fumadores y no fumadores por edad, sexo y altura. Esta base de datos es pública del autor Mike Marin, quien provee una [serie de tutoriales](#) sobre estadística y R de forma gratuita; tutoriales que recomendamos ampliamente.

La base de datos consta de las siguientes variables:

- ♦ LungCap: capacidad pulmonar
- ♦ Age: edad en años
- ♦ Height: altura de la persona
- ♦ Smoke: variable que indica si el paciente fuma o no
- ♦ Gender: hombre o mujer
- ♦ Caesarean: Si nacieron por cesárea o no

Un grupo de investigadores desea conocer cómo influye la edad (Age) y la estatura (Hieght) sobre la capacidad pulmonar, para ello se plantea realizar una regresión lineal simple y luego múltiple.

Para cada uno de sus modelos compruebe todos los supuestos y realice una interpretación.

- ♦ **Ejercicio 22.3:** utilice la base de datos [bodyfat.txt](#) y construya un modelo que permita predecir la circunferencia de cintura de un paciente (abdomen) dadas el resto de medidas antropométricas.

Puede construir un modelo añadiendo todas las variables e ir descartando aquellas que no aporten al modelo (sin significancia) o puede ir añadiendo variables hasta construir un modelo adecuado. Un modelo válido es aquel en el que todas las variables aportan a la variable de respuesta (son significativas). De hecho, la prueba de hipótesis que deseamos probar es qué adición de una variable al modelo aumenta la predicción del mismo.

En “[Ejercicios para pruebas de correlación](#)” describimos más detalles de la base de datos [bodyfat.txt](#)

Referencias

- Bhattacharya, S., Bera, O. P., Saleem, S. M., Hossain, M. M., Varshney, D. S., Kaur, R., Rana, R. K., Tripathi, S., Gokdemir, O., Bacorro, M., Mehta, K. y Singh, A. (2022). Dietary salt consumption pattern as an antecedent risk factor for hypertension: Status, vision, and future recommendations. *Clinical nutrition ESPEN*, 47, 422-430. <https://doi.org/10.1016/j.clnesp.2021.12.006>
- Dombrowsky, T. (2023). Linear regression: A beginner's guide for nursing research. *Nursing*, 53(9), 56-60. <https://doi.org/10.1097/01.NURSE.0000946844.96157.68>
- Eberly, L. E. (2007). Correlation and simple linear regression. *Methods in molecular biology*, 404, 143-164. https://doi.org/10.1007/978-1-59745-530-5_8
- Hanke, M., Dijkstra, L., Foraita, R. y Didelez, V. (2024). Variable selection in linear regression models: Choosing the best subset is not always the best choice. *Biometrical journal. Biometrische Zeitschrift*, 66(1), e2200209. <https://doi.org/10.1002/bimj.202200209>

- Hanley, J. A. (2016). Simple and multiple linear regression: sample size considerations. *Journal of clinical epidemiology*, 79, 112-119. <https://doi.org/10.1016/j.jclinepi.2016.05.014>
- Hess, A. S. y Hess, J. R. (2017). Linear regression and correlation. *Transfusion*, 57(1), 9-11. <https://doi.org/10.1111/trf.13928>
- Indrayan, A. y Malhotra, R. K. (2017). Relationships: Quantitive Outcome. En *Medical Biostatistics* (4ª ed., pp. 423-465). Chapman & Hall/CRC Biostatistics Series.
- Jahan, S. y Khan, A. R. (2012). Power of t -test for simple linear regression model with non-normal error distribution: a quantile function distribution approach. *Journal of Scientific Research*, 4(3), 609-622. <https://doi.org/10.3329/jsr.v4i3.9067>
- Krzywinski, M. y Altman, N. (2015). Multiple linear regression. *Nature methods*, 12(12), 1103-1104. <https://doi.org/10.1038/nmeth.3665>
- Lunt, M. (2015). Introduction to statistical modelling: linear regression. *Rheumatology* (Oxford, England), 54(7), 1137-1140. <https://doi.org/10.1093/rheumatology/ket146>
- Marill, K. A. (2004). Advanced statistics: linear regression, part II: multiple linear regression. *Academic emergency medicine: official journal of the Society for Academic Emergency Medicine*, 11(1), 94-102. <https://doi.org/10.1197/j.aem.2003.09.006>
- McLeod, I. (1999). Improved Spread-Location Visualization. *Journal of Computational and Graphical Statistics*, 8. [10.1080/10618600.1999.10474806](https://doi.org/10.1080/10618600.1999.10474806)
- O'Brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, 41(5), 673-690. <https://doi.org/10.1007/s11135-006-9018-6>
- Olsen, A. A., McLaughlin, J. E. y Harpe, S. E. (2020). Using multiple linear regression in pharmacy education scholarship. *Currents in pharmacy teaching & learning*, 12(10), 1258-1268. <https://doi.org/10.1016/j.cptl.2020.05.017>
- Penn State Eberly College of Science (s. f.). Lesson 11: Simple linear regression. STAT 501: Regression Methods. <https://online.stat.psu.edu/stat501/lesson/11/11.2>
- Schmidt, A. F. y Finan, C. (2018). Linear regression and the normality assumption. *Journal of Clinical Epidemiology*, 98, 146-151. <https://doi.org/10.1016/j.jclinepi.2017.12.006>
- Schmidt, A. F. y Finan, C. (2018). Linear regression and the normality assumption. *Journal of clinical epidemiology*, 98, 146-151. <https://doi.org/10.1016/j.jclinepi.2017.12.006>
- Schneider, A., Hommel, G. y Blettner, M. (2010). Linear regression analysis: part 14 of a series on evaluation of scientific publications. *Deutsches Arzteblatt international*, 107(44), 776-782. <https://doi.org/10.3238/arztebl.2010.0776>

- Schober, P. y Vetter, T. R. (2021). Linear Regression in Medical Research. *Anesthesia and analgesia*, 132(1), 108-109. <https://doi.org/10.1213/ANE.0000000000005206>
- Shrestha, N. (2020). Detecting multicollinearity in regression analysis. *American Journal of Applied Mathematics and Statistics*, 8(2), 39-42. <https://doi.org/10.12691/ajams-8-2-1>
- Takase, H., Sugiura, T., Kimura, G., Ohte, N. y Dohi, Y. (2015). Dietary Sodium Consumption Predicts Future Blood Pressure and Incident Hypertension in the Japanese Normotensive General Population. *Journal of the American Heart Association*, 4(8), e001959. <https://doi.org/10.1161/JAHA.115.001959>
- Vines, M., Tootle, G. A., Terry, L. G., Elliott, E., Corbin, J., Harley, G. L. y Therrell, M. D. (2021). A paleo perspective of alabama and florida (usa) interstate stream-flow. *Water*, 13(5), 657. <https://doi.org/10.3390/w13050657>
- Wan, F., Liu, W. y Bretz, F. (2024). Confidence sets for a level set in linear regression. *Statistics in medicine*, 43(6), 1103-1118. <https://doi.org/10.1002/sim.9996>

Regresión logística

Alberto Beltrán Ramírez

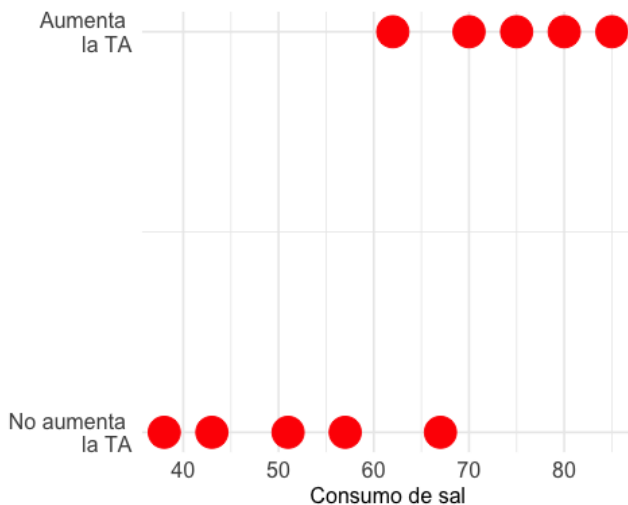
Edsaúl Emilio Pérez Guerrero

Esta es una técnica que se puede usar tanto para estadísticas tradicionales como para *Machine Learning*, así que vamos a ello.

Antes de entrar de lleno en la regresión logística, recordemos que en el [capítulo 22](#) se habló acerca del efecto del consumo de sal sobre la tensión arterial. El ejemplo mostrado en ese capítulo utilizaba datos como el consumo de sal y las cifras de tensión arterial sistólica, donde los diversos valores fueron ajustados a una línea, y gracias a ello, con ayuda del método de los cuadrados mínimos, pudimos observar que por cada mg/24H que se aumente el consumo de sal, la tensión arterial aumenta en promedio 0.020370 mmHg. Debemos recordar que gracias a las regresiones podemos conocer el efecto de diversas variables predictoras sobre una variable dependiente o, en su defecto, permite controlar y ajustar por variables asociadas a nuestra variable dependiente. En el ejemplo previo, se agregaron comorbilidades, así como una variable relacionada al consumo de tabaco, definida en tres categorías obteniendo los resultados descritos en [“Regresión lineal múltiple R”](#).

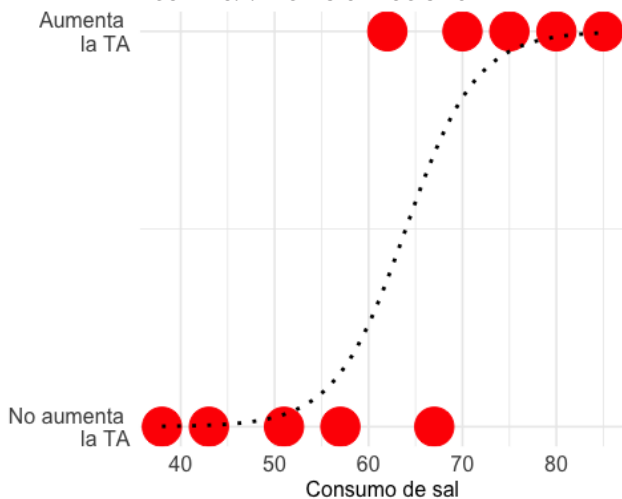
La regresión logística tiene un fundamento similar a la regresión lineal, con la diferencia de que no predice algo continuo como el incremento de la tensión arterial en (mmHG), sino que lo predice en forma dicotómica, es decir en “sí o no”, “verdadero o falso”, “positivo o negativo”, “aumenta la tensión arterial o no” (ver [figura 23.1](#)).

FIGURA 23.1. REGRESIÓN LOGÍSTICA 1



Esta regresión, no ajusta una línea a los datos como la lineal, sino que ajusta en una forma de S (ver [figura 23.2](#)).

FIGURA 23.2. REGRESIÓN LOGÍSTICA 2



Esta curva que va de 0 a 1 (ver [figura 23.3](#)), indica la probabilidad de que una persona aumente se TA basada en su consumo de sal (ver [figura 23.4](#) y [figura 23.5](#)).

FIGURA 23.3. REGRESIÓN LOGÍSTICA 3

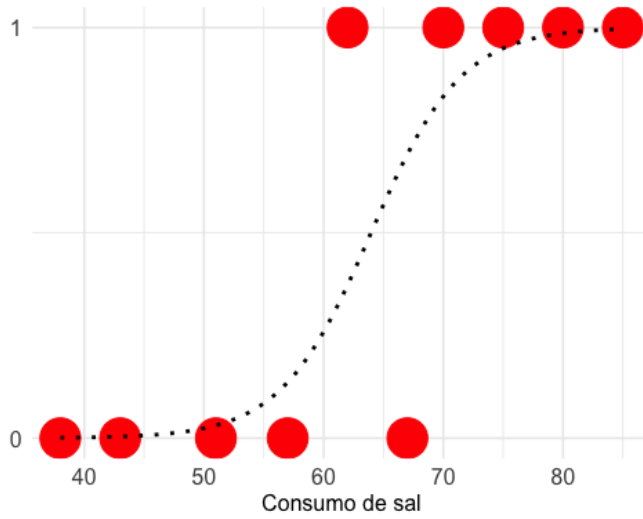


FIGURA 23.4. REGRESIÓN LOGÍSTICA 4

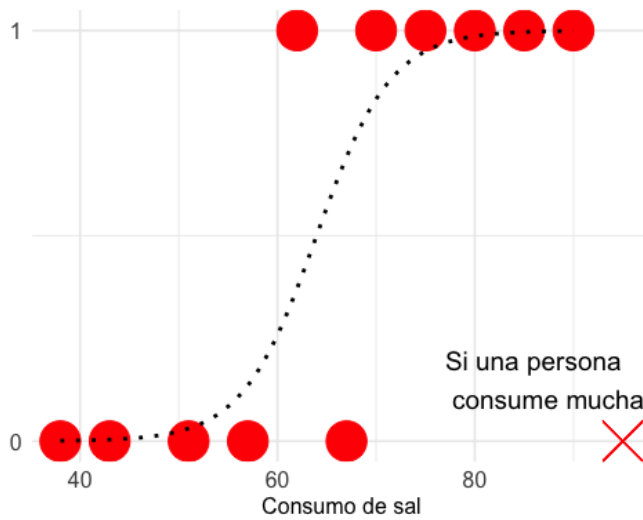
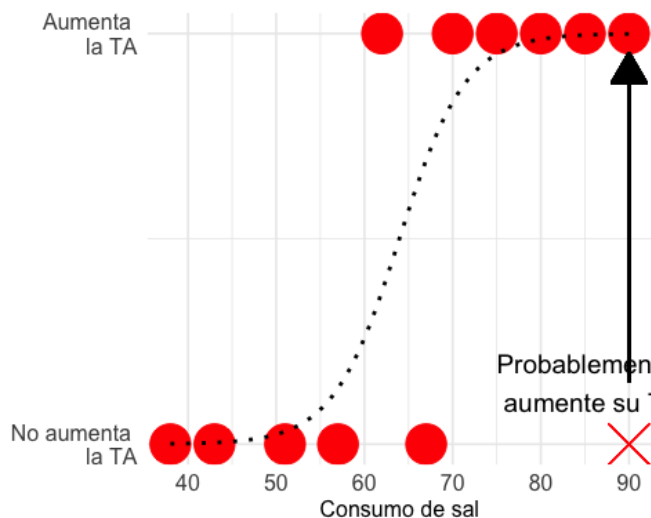


FIGURA 23.5. REGRESIÓN LOGÍSTICA 5



La regresión logística es utilizada con frecuencia para clasificación. Por ejemplo, si una persona consume $> 50\%$ de sal, lo clasificaremos como una persona con aumento de la TA, de lo contrario, será una persona sin aumento de TA (ver [figura 23.6](#) y [figura 23.7](#)); una persona que consume poca sal tendrá baja probabilidad de aumentar su TA (ver [figura 23.8](#)).

FIGURA 23.6. REGRESIÓN LOGÍSTICA 6

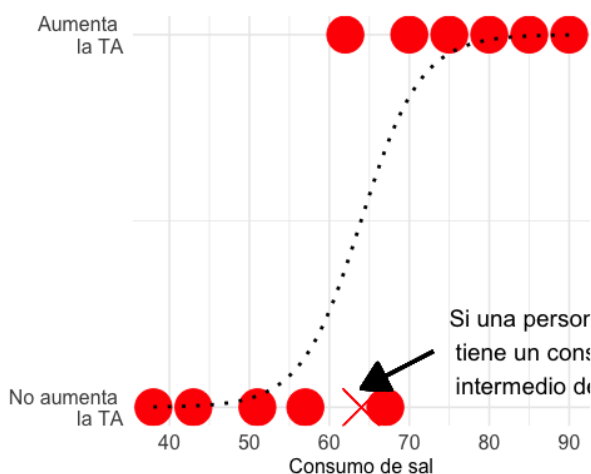


FIGURA 23.7. REGRESIÓN LOGÍSTICA 7

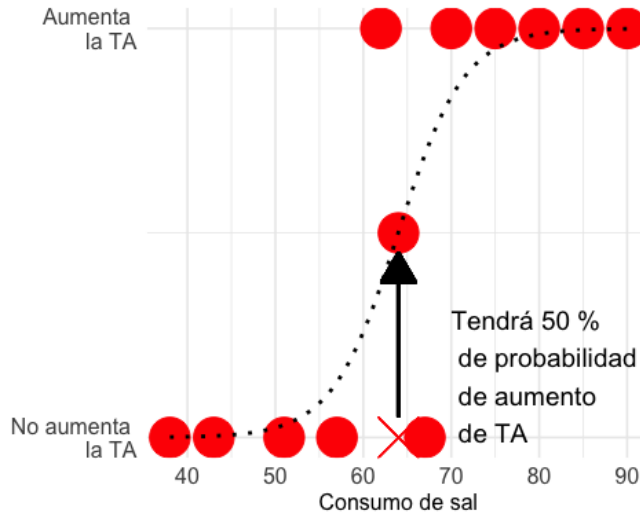
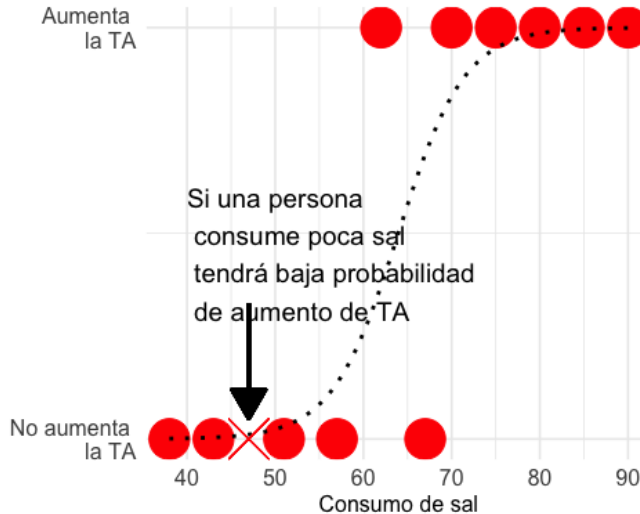


FIGURA 23.8. REGRESIÓN LOGÍSTICA 8



Al igual que en la regresión lineal, podemos hacer modelos simples, como el de este ejemplo, prediciendo si la TA aumenta o no, dependiendo del consumo de sal de cada persona.

Podemos también realizar modelos más complejos que incluyan más variables, como lo pueden ser las comorbilidades propias de las personas o si fuman o no, entre otras, como podría ser la edad o el peso.

Al igual que la regresión lineal, la regresión logística puede trabajar con variables continuas (edad, peso) o variables discretas (comorbilidades o si fuman o no), pero a diferencia de la regresión lineal, en la logística no es sencillo comparar los modelos múltiples con el simple. En la logística se busca ver si el efecto de una variable predictora es significativamente de 0, y si no lo es, esta variable no es útil en nuestra predicción.

Usted puede escoger una probabilidad, basada en el consumo de sal, de observar una persona a la que le aumente la TA, como se aprecia en la [figura 23.9](#) y la utiliza para calcular la probabilidad de observar una persona que no aumenta su TA con un consumo de sal bajo (ver [figura 23.10](#)); después, calculará la misma posibilidad, pero observando el consumo de sal de la siguiente persona (ver [figura 23.11](#)) y así sucesivamente con todas las personas (ver [figura 23.12](#)). Finalmente, multiplica todas las probabilidades juntas siendo este resultado, la probabilidad de los datos de la curva de la [figura 23.9](#).

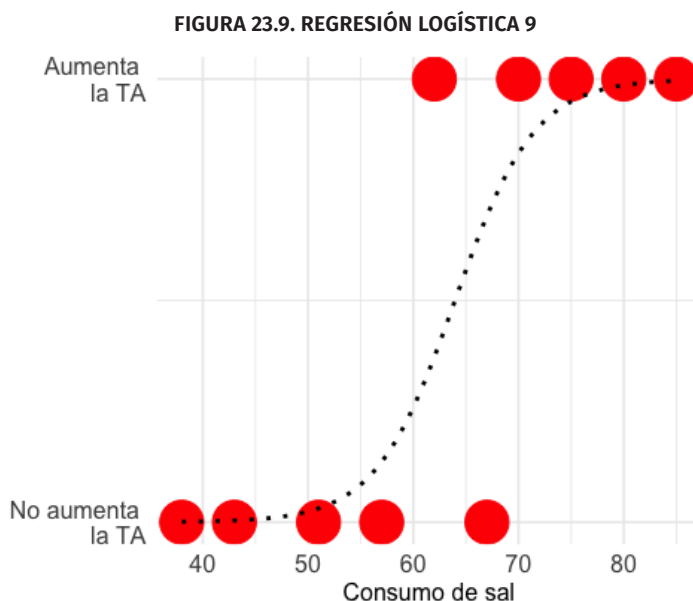


FIGURA 23.10. REGRESIÓN LOGÍSTICA 10

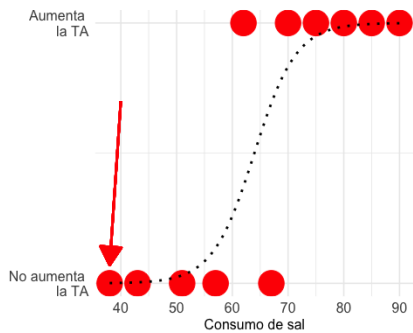


FIGURA 23.11. REGRESIÓN LOGÍSTICA 11

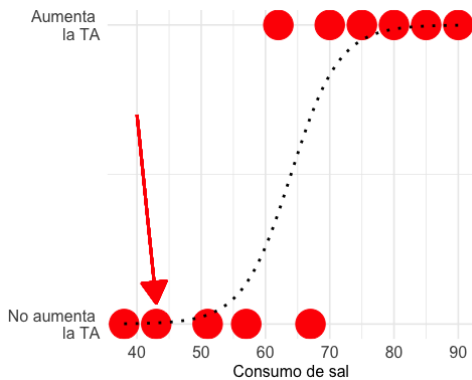
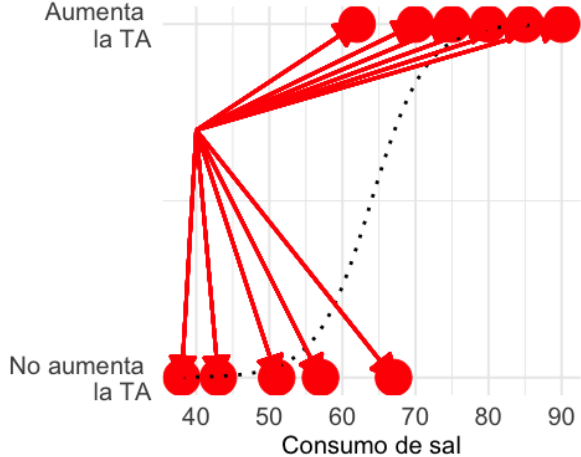


FIGURA 23.12. REGRESIÓN LOGÍSTICA 12



Después, se cambia la línea y se calcula una nueva probabilidad de datos (ver [figura 23.13](#)), repitiendo esto las veces necesarias (ver [figura 23.14](#)) hasta que la curva con la máxima probabilidad sea seleccionada (ver [figura 23.15](#)).

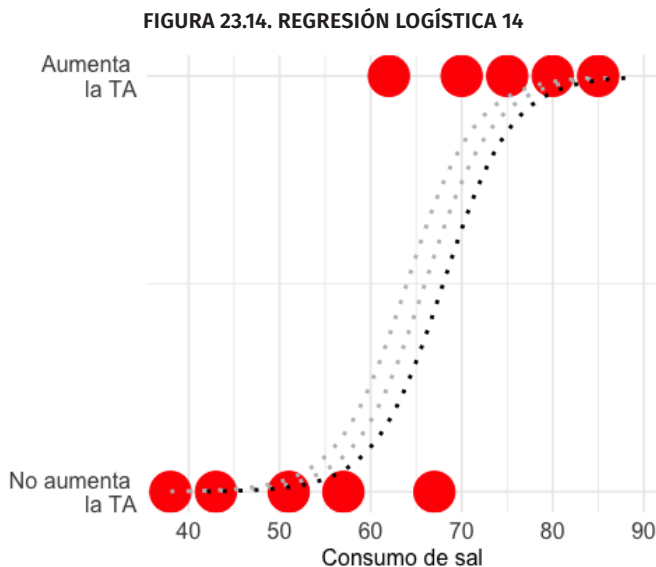
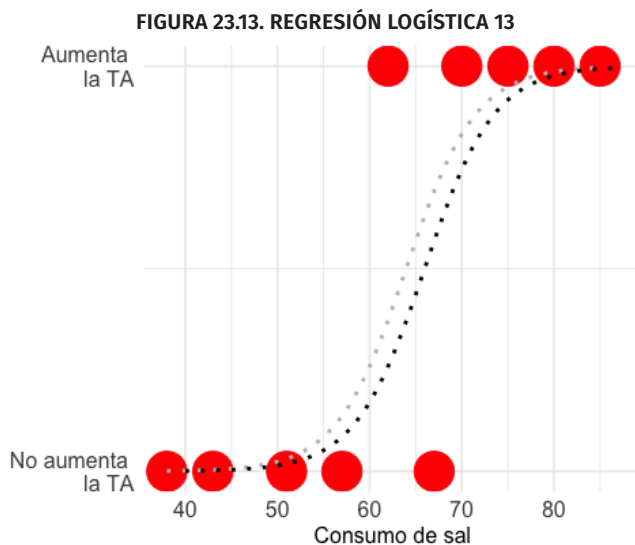
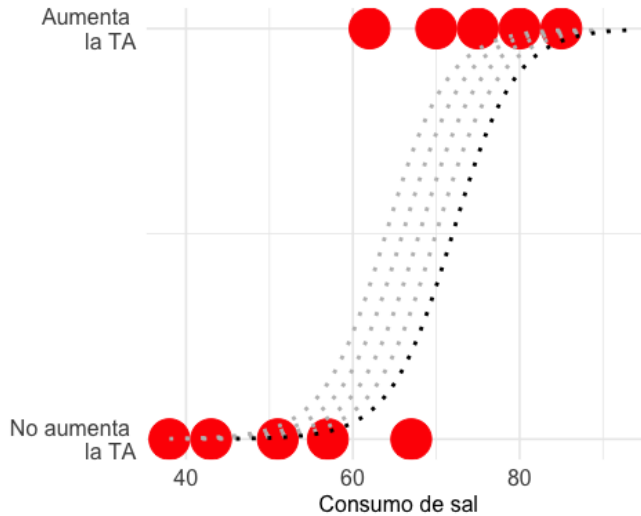


FIGURA 23.15. REGRESIÓN LOGÍSTICA 15



Características de la regresión logística

- ♦ Es un modelo que permite conocer el efecto de una o más variables sobre una variable dependiente binaria.
- ♦ Pertenece a los modelos lineales generalizados, es decir, utiliza muchas de las bases del modelo lineal, pero lo aplica a variables que no siguen una distribución normal.
- ♦ Para poder asignar todos los valores de 0 y 1 utiliza la función logística, la cual permite transformar todos los valores a 0 y 1. Se estima mediante:

$$f(z) = \frac{1}{1 + e^{-z}}$$

- ♦ A partir de la función logística, obtenemos el modelo logístico mediante la sustitución z por $\beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots \beta_k X_k$, de modo que:

$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots + \beta_k X_k)}}$$

- ♦ Donde los coeficientes se estiman mediante el método de máxima verosimilitud (la idea es encontrar los valores de los coeficientes que maximicen la función de verosimilitud, es decir, que hagan que la probabilidad de observar los datos dados de estos coeficientes sea la más alta posible). Cuando estos coeficientes son conocidos, podemos, mediante el modelo logístico, calcular la probabilidad de que un individuo presente un evento de interés. De hecho, la regresión logística es un método que asigna probabilidades
- ♦ Interpretación de los coeficientes: en la regresión logística, los coeficientes β se interpretan en términos odds ratio (ver “Odds ratio (OR)”). Un coeficiente positivo indica que un aumento en la variable independiente se asocia con un aumento en la probabilidad del evento, mientras que un coeficiente negativo indica una disminución en la probabilidad del evento.
- ♦ El uso de R^2 en la regresión lineal para la comparación de los modelos, no es posible llevarse a cabo en la regresión logística debido a sus características, por lo que utiliza la **máxima probabilidad** (\$maximum likelihood).

Supuestos

1. Linealidad en el logit: la relación entre las variables independientes y el logit (logaritmo de las probabilidades) de la variable dependiente debe ser lineal.
2. Independencia de las observaciones: las observaciones deben ser independientes entre sí.
3. No multicolinealidad: las variables independientes no deben estar altamente correlacionadas entre sí.
4. Suficiente tamaño de muestra: para obtener estimaciones precisas y estables, es necesario contar con un tamaño de muestra adecuado, especialmente si hay muchas variables independientes.

Estimación de una regresión logística en R

La estructura de código para evaluar una regresión logística en R es la siguiente:

```
# Ajustar el modelo de regresión logística
modelo ← glm(outcome ~ predictor1 +
```

```

    predictor2 + predictor3,
    data = data, family = binomial)

# Resumen del modelo
summary(modelo)

```

- ♦ **Ejemplo 23.1:** vamos a retomar el [ejemplo 23.1](#), en el cual se buscaba evaluar el efecto del consumo de sal sobre la tensión arterial en un grupo de pacientes.

Se sabe por antecedentes previos del grupo de investigación que se puede establecer un punto de corte para las cifras de tensión arterial sistólica para diagnosticar a un paciente con hipertensión o sin hipertensión. Este punto de corte es 140 mmHg.

Vamos a crear esta nueva variable en nuestro *Dataframe*, para después evaluarlo mediante una regresión logística. Si ya no tiene su base de datos en el ambiente de trabajo, impórtela de nuevo:

```

df <- read.csv("Bases/sal_tension_arterial.csv")
df$Hipertension <- cut(df$Tension_Arterial,
    breaks = c(0,140,Inf),
    labels = c("Sin hipetensión",
               "Hipertensión"))

```

Note cómo deben de crearse los factores en el orden correcto. Es decir, el nivel más bajo debe ser la no presencia del evento o característica. Y el valor más alto debe ser la presencia del evento o de la característica.

Nuestro objetivo es conocer si la hipertensión se asocia con el consumo de sal.

Ahora procedemos a crear nuestro modelo de regresión logística:

```

modelo <- glm(formula = Hipertension~Consumo_Sal, data = df,
    family=binomial())
summary(modelo)

```

Call:

```
glm(formula = Hipertension ~ Consumo_Sal, family = binomial(),
```

```
data = df)

Coefficients:
              Estimate      Std.      Error  z value Pr(>|z|)
(Intercept) -17.335882    4.522487    -3.833   0.000126 ***
Consumo_Sal  0.005539     0.001528     3.626   0.000288 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 64.064  on 79  degrees of freedom
Residual deviance: 35.589  on 78  degrees of freedom
AIC: 39.589

Number of Fisher Scoring iterations: 7
```

El coeficiente para el consumo de sal es positivo y diferente de 0, eso indica que a medida que aumente el consumo de sal también lo hace la probabilidad de tener hipertensión.

En el caso de la regresión logística, el valor del coeficiente según la función logística es un logaritmo de un odds (*log Odds*), por lo tanto es necesario elevarlo a la exponencial para obtener el valor de OR:

```
exp(modelo$coefficients) # Para el OR
(Intercept) Consumo_Sal
2.958843e-08 1.005554e+00
```

El valor del OR es de aproximadamente 1.0055, esto significa que por cada unidad adicional de consumo de sal, las odds de tener hipertensión aumentan en aproximadamente 0.555%

Para estimar los intervalos de confianza para los OR utilice el siguiente código:

```
exp_confint ← exp(confint(modelo))
```

Waiting for profiling to be done...

Mostrar resultados

exp_confint

	2.5 %	97.5 %
(Intercept)	5.634785e-13	4.658568e-05
Consumo_Sal	1.003026e+00	1.009223e+00

Los intervalos de confianza precisan algunas aclaraciones. El OR con 95% de confianza es diferente de 1 (lo cual coincide con el valor). Este intervalo es muy estrecho [1.003026, 1.009223], lo que indica una estimación precisa. Sin embargo, el efecto del consumo de sal en la hipertensión, según el OR de 1.005539, es pequeño en términos de aumento de odds por unidad de consumo. No obstante, la relevancia clínica de este hallazgo depende del contexto.

Ahora debemos de comprobar los supuestos del modelo.

♦ Linealidad logit

```
# Crear una variable log-transformada del consumo de sal
# para verificar la linealidad
df$log_Consumo_Sal <- log(df$Consumo_Sal)

# Ajustar el modelo incluyendo el término log-transformado
modelo_logit_linealidad <- glm(Hipertension ~
                               Consumo_Sal + log_Consumo_Sal,
                               family = binomial(), data = df)
summary(modelo_logit_linealidad)
```

Call:

```
glm(formula = Hipertension ~ Consumo_Sal + log_Consumo_Sal, family
= binomial(),
    data = df)
```

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept)	-488.80143	590.82819	-0.827	0.408
Consumo_Sal	-0.01733	0.02809	-0.617	0.537
log_Consumo_Sal	67.48824	84.34009	0.800	0.424

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 64.064 on 79 degrees of freedom
 Residual deviance: 34.808 on 77 degrees of freedom
 AIC: 40.808

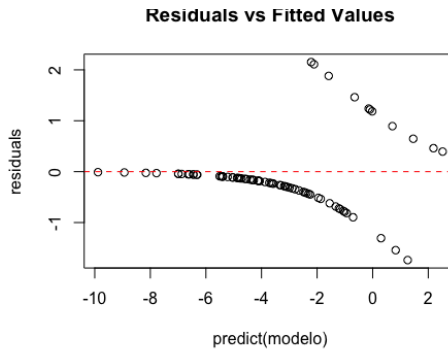
Number of Fisher Scoring iterations: 9

Si los términos log-transformados o polinómicos son significativos, puede indicar una desviación de la linealidad en el logit.

♦ Diagnóstico de los residuales

```
# Calcular los residuos
residuals <- residuals(modelo, type = "deviance")

# Graficar los residuos
plot(predict(modelo), residuals,
     main = "Residuals vs Fitted Values")
abline(h = 0, lty = 2, col = "red")
```



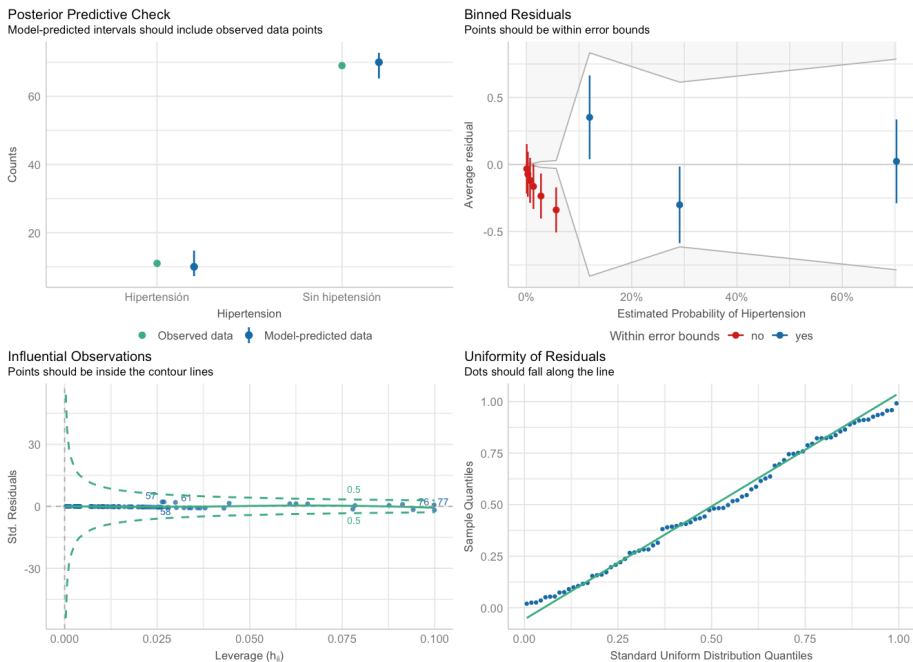
Esta gráfica ayuda a identificar cualquier patrón en los residuos que pueda indicar un mal ajuste del modelo

La librería `performance` ofrece una serie de funciones que permiten la evaluación complementaria del modelo:

```
install.packages("performance")
performance::check_model(modelo)
```

Esta función nos muestra cuatro gráficas (ver [figura 23.16](#)) que trataremos de explicar sencillamente.

FIGURA 23.16. GRÁFICOS DIAGNÓSTICO PARA LA REGRESIÓN LOGÍSTICA



El gráfico de la esquina superior izquierda es el “Posterior Predictive Check”. En este se comparan los datos observados con los datos predichos por el modelo. En una gráfica adecuada (como la nuestra), los intervalos de

confianza de los datos predichos deberían incluir los datos observados. El segundo gráfico, el que se encuentra en la esquina superior derecha, se conoce como Binned plot. En este gráfico los residuos deberían estar dentro de los límites de error (representados por las líneas negras). En nuestra gráfica, la mayoría de los puntos azules están dentro de los límites de error, lo cual es positivo. Sin embargo, hay algunos puntos rojos que indican residuos fuera de los límites de error, lo que podría sugerir algún ajuste inadecuado del modelo.

El gráfico que se encuentra en la esquina inferior izquierda, nos ayuda a identificar outliers, es decir puntos que se encuentran fuera de la línea punteada. El último gráfico es un gráfico Q-Q que ya hemos explicado con anterioridad.

Una de las bondades de la regresión logística es que podemos ajustar por múltiples variables. Vamos a construir modelo para conocer el efecto de otras variables sobre la hipertensión.

♦ Ejemplo 23.2

```
modelo_ajustado <- glm(formula = Hipertension~Consumo_Sal+
                        Comorbilidades+Edad+Sexo, data = df,
                        family = binomial())
```

```
summary(modelo_ajustado)
```

Call:

```
glm(formula = Hipertension ~ Consumo_Sal + Comorbilidades + Edad +
     Sexo, family = binomial(), data = df)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-24.467462	7.377361	-3.317	0.000911	***
Consumo_Sal	0.007423	0.002360	3.145	0.001658	**
Comorbilidades	2.102044	1.218514	1.725	0.084512	.
Edad	0.010306	0.025350	0.407	0.684337	
SexoMujer	-0.236731	1.005797	-0.235	0.813924	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 64.064 on 79 degrees of freedom
 Residual deviance: 30.858 on 75 degrees of freedom
 AIC: 40.858

Number of Fisher Scoring iterations: 7

```
exp(modelo_ajustado$coefficients)
```

(Intercept)	Consumo_Sal	Comorbilidades	Edad	SexoMujer
2.365463e-11	1.007451e+00	8.182875e+00	1.010359e+00	7.892033e-01

```
confint(modelo_ajustado)|>
```

```
exp()
```

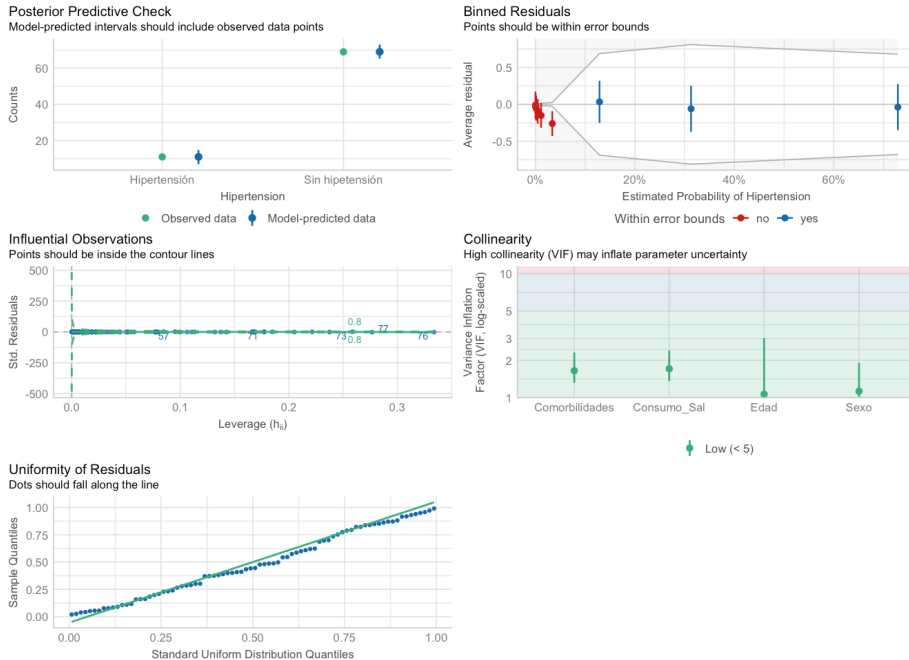
	2.5 %	97.5 %
(Intercept)	1.238574e-19	2.073955e-06
Consumo_Sal	1.003852e+00	1.013715e+00
Comorbilidades	1.008988e+00	1.466818e+02
Edad	9.615658e-01	1.065827e+00
SexoMujer	9.733261e-02	5.809038e+00

Para este modelo, las variables que contribuyen a la elevación de la tensión arterial solo es el consumo de sal. El resto de variables no tienen efecto sobre las cifras de tensión arterial.

El consumo de sal es un predictor significativo de hipertensión. Un aumento en el consumo de sal se asocia con un incremento en las odds de hipertensión. El odds ratio (OR) es aproximadamente 1.007, lo que significa que, por cada unidad adicional de consumo de sal, las odds de hipertensión aumentan en 0.7423%.

Para revisar la validez del modelo utilizamos la función `check_model()`:

```
performance::check_model(modelo_ajustado)
```



En estas gráficas prácticamente tenemos la misma interpretación del ejemplo anterior. Ahora, al incluir más variables también es necesario que se evalúe la colinealidad con el VIF. En este caso se cumple con el supuesto de colinealidad, sin embargo, el gráfico de Binned Residuals presenta algunas señales de alarma que pueden poner en riesgo la validez de mi modelo.

En el [ejemplo 23.2](#) observamos que solo una de las variables se asoció con la hipertensión. En este caso el resto de variables deberían de excluirse del modelo, sin embargo, para esta situación y contexto en específico suponiendo que nuestro modelo es válido, podríamos concluir que el consumo de sal es un predictor independiente para la hipertensión.

Ejercicios

- ♦ **Ejercicio 23.1:** se desea identificar las variables asociadas al bajo peso al nacer mediante una regresión logística. Para ello, se parte del *Dataframe* `birthwt` de la librería *MASS*. Realice lo siguiente:
 - a) Adecúe la base cambiando variables y creando factores
 - b) Realice un modelo de regresión logística con todas las variables de la base
 - c) Seleccione solo aquellas que se asocian con el bajo peso al nacer
 - d) Valide su modelo
 - e) Interprete

- ♦ **Ejercicio 23.2:** la Organización Mundial de la Salud ha estimado que cada año ocurren 12 millones de muertes a nivel mundial debido a enfermedades cardíacas. La mitad de las muertes en Estados Unidos y otros países desarrollados se deben a enfermedades cardiovasculares. El pronóstico temprano de las enfermedades cardiovasculares puede ayudar en la toma de decisiones sobre cambios en el estilo de vida en pacientes de alto riesgo y, a su vez, reducir las complicaciones. Esta investigación tiene como objetivo identificar los factores de riesgo más relevantes de la enfermedad cardíaca, así como predecir el riesgo general utilizando la regresión logística. Puede consultar más información [aquí](#).

El conjunto de datos está disponible públicamente en el sitio web de Kaggle, y proviene de un estudio cardiovascular en curso sobre los residentes de la ciudad de Framingham, Massachusetts. El objetivo de la clasificación es predecir si el paciente tiene un riesgo a 10 años de enfermedad coronaria futura (CHD, por sus siglas en inglés). El conjunto de datos proporciona información de los pacientes. Incluye más de 4 000 registros y 15 atributos.

Cada atributo es un potencial factor de riesgo. Hay factores de riesgo demográficos, conductuales y médicos.

A continuación, se describen las variables de la base de datos y su significado:

- ♦ Demographic variables
- ♦ Sex: male or female (Nominal)

- ◇ Age: age of the patient; (continuous: although the recorded ages have been truncated to whole numbers, the concept of age is continuous); behavioral
- ◇ Current Smoker: whether or not the patient is a current smoker (nominal)
- ◇ Cigs Per Day: the number of cigarettes that the person smoked on average in one day (can be considered continuous as one can have any number of cigarettes, even half a cigarette)
- ◇ Medical (history) variables
- ◇ BP Meds: whether or not the patient was on blood pressure medication (nominal)
- ◇ Prevalent Stroke: whether or not the patient had previously had a stroke (nominal)
- ◇ Prevalent Hyp: whether or not the patient was hypertensive (nominal)
- ◇ Diabetes: whether or not the patient had diabetes (nominal), medical (current) variables
- ◇ Tot Chol: total cholesterol level (continuous)
- ◇ Sys BP: systolic blood pressure (continuous)
- ◇ Dia BP: diastolic blood pressure (continuous)
- ◇ BMI: Body Mass Index (continuous)
- ◇ Heart Rate: heart rate (Continuous: in medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values)
- ◇ Glucose: glucose level (Continuous), predict variable (desired target)
- ◇ Dependent variable (TenYearCHD)
- ◇ 10 year risk of coronary heart disease CHD (binary: “1”, means “Yes”, “0” means “No”)

Su objetivo es realizar modelos de regresión logística para identificar variables asociadas al riesgo cardiovascular a los diez años.

Puede descargar la base de datos de [aquí](#). Puede encontrar más información sobre el resultado [aquí](#).

Deberá realizar lo siguiente:

1. Análisis bivariado agrupando por la variable TenYearCHD para identificar variables asociadas al riesgo cardiovascular que pudieran incluirse en el modelo

2. Seleccionar y ajustar un modelo de regresión logística con las variables seleccionadas
3. Evaluar los supuestos del modelo
4. Realizar la prueba de Hosmer-Lemeshow
5. Calcular la pseudo
6. Estimar los OR y sus intervalos de confianza
7. Interpretar sus OR
8. Concluir

Referencias

- Cornillon, P. A. (2012). Classification. En P. A. Cornillon, *R for Statistics* (pp. 181-208). Taylor & Francis Group.
- Dalgaard, P. (2008). Logistic Regression (2ª ed.). En P. Dalgaard, *Introductory Statistics with R* (pp. 228-249). Springer.
- De Jespus Celis, A. (2020). Análisis Estratificado y Regresión Logística. En A. De Jespus Celis, *Bioestadística* (pp. 209-230). Editorial Médica.
- El Sanharawi, M. y Naudet, F. (2013). Comprendre la régression logistique [Understanding logistic regression]. *Journal Français d’Ophtalmologie*, 36(8), 710-715. <https://doi.org/10.1016/j.jfo.2013.05.008>
- Ghatak, A. (2017). Classification. En *Machine Learning with R* (pp. 115-176). Springer. <https://doi.org/10.1007/978-981-10-6808-9>
- Gosho, M., Ohigashi, T., Nagashima, K., Ito, Y. y Maruo, K. (2023). Bias in odds ratios from logistic regression methods with sparse data sets. *Journal of Epidemiology*, 33(6), 265-275. <https://doi.org/10.2188/jea.JE20210089>
- Harris, J. K. (2021). Primer on binary logistic regression. *Family medicine and community health*, 9(Suppl 1), e001290. <https://doi.org/10.1136/fmch-2021-001290>
- Hoffmann, J. P. (2021). A brief introduction to logistic regression. En *Linear Regression Models: Applications in R* (pp. 337-354). Chapman & Hall/CRC. <https://doi.org/10.21105/joss.03139>
- Indrayan, A. y Malhotra, R. K. (2018). Relationships: Qualitative Dependent. En A. Indrayan y R. K. Malhotra, *Medical Biostatistics* (4ª ed., pp. 469-502). CRC Press.

- James, G., Witten, D., Hastie, T. y Tibshirani, R. (2013). Classification. En G. James, D. Witten, T. Hastie y R. Tibshirani, *An Introduction to Statistical Learning with Applications in R* (pp. 127-174). Springer.
- Kasza, J. (2015). Stata tip 125: binned residual plots for assessing the fit of regression models for binary outcomes. *The Stata Journal: Promoting Communications on Statistics and Stata*, 15(2), 599-604. <https://doi.org/10.1177/1536867x1501500219>
- Koletsis, D. y Pandis, N. (2017). Conditional logistic regression. *American Journal of Orthodontics and Dentofacial Orthopedics*, 151(6), 1191-1192. <https://doi.org/10.1016/j.ajodo.2017.04.009>
- LaValley, M. P. (2008). Logistic regression. *Circulation*, 117(18), 2395-2399. <https://doi.org/10.1161/CIRCULATIONAHA.106.682658>
- Long, R. G. (2008). The crux of the method: assumptions in ordinary least squares and logistic regression. *Psychological reports*, 103(2), 431-434. <https://doi.org/10.2466/pro.103.2.431-434>
- Lüdecke et al. (2021). Performance: An R Package for Assessment, Comparison and Testing of Statistical Models. *Journal of Open Source Software*, 6(60), 3139.
- Lydersen, S. (2022). Logistic regression with more than two categories. *Tidsskrift for den Norske Lægeforening*, 142(10). <https://doi.org/10.4045/tidsskr.21.0786>
- MacFarland, T. W. y Yates, J. M. (2021). Statistical Test(s). En T. W. MacFarland y J. M. Yates, *Using R for Biostatistics* (pp. 488-528). Springer.
- Martínez Pérez, J. A. y Pérez Martín, P. S. (2024). Regresión logística [Logistic regression]. *Semergen*, 50(1), 102086. <https://doi.org/10.1016/j.semerg.2023.102086>
- Moscarelli, M. (2023). Logistic Regression. En M. Moscarelli, *Biostatistics With 'R': A Guide for Medical Doctors* (pp. 181-206). Springer.
- Nick, T. G. y Campbell, K. M. (2007). Logistic regression. *Methods in molecular biology*, 404, 273-301. https://doi.org/10.1007/978-1-59745-530-5_14
- Pandis, N. (2017a). Logistic regression: Part 1. *American Journal of Orthodontics and Dentofacial Orthopedics*, 151(4), 824-825. <https://doi.org/10.1016/j.ajodo.2017.01.017>
- Pandis, N. (2017b). Logistic regression: Part 2. *American Journal of Orthodontics and Dentofacial Orthopedics*, 151(5), 1008. <https://doi.org/10.1016/j.ajodo.2017.02.002>
- Ranganathan, P., Pramesh, C. S. y Aggarwal, R. (2017). Common pitfalls in statistical analysis: Logistic regression. *Perspectives in clinical research*, 8(3), 148-151. https://doi.org/10.4103/picr.PICR_87_17

- Riffenburgh, R. H. (2005). *Regression and Correlation Models* En R. H. Riffenburgh, *Statistics in Medicine* (2ª ed., pp. 446-486). Elsevier.
- Sainani, K. L. (2014). Logistic regression. *PM&R*, 6(12), 1157-1162. <https://doi.org/10.1016/j.pmrj.2014.10.006>
- Sainani, K. L. (2021). Multinomial and ordinal logistic regression. *PM&R*, 13(9), 1050-1055. <https://doi.org/10.1002/pmrj.12622>
- Schober, P. y Vetter, T. R. (2021). Logistic Regression in Medical Research. *Anesthesia and analgesia*, 132(2), 365-366. <https://doi.org/10.1213/ANE.0000000000005247>
- Schober, P. y Vetter, T. R. (2021). Logistic regression in medical research. *Anesthesia & Analgesia*, 132(2), 365-366. <https://doi.org/10.1213/ANE.0000000000005247>
- Shalev-Shwartz, S. y Ben-David, S. (2014). A formal learning model. En *Understanding Machine Learning: From Theory to Algorithms* (pp. 43-50). Cambridge University Press.
- Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia medica*, 24(1), 12-18. <https://doi.org/10.11613/BM.2014.003>
- Stoltzfus, J. C. (2011). Logistic regression: a brief primer. *Academic emergency medicine: official journal of the Society for Academic Emergency Medicine*, 18(10), 1099-1104. <https://doi.org/10.1111/j.1553-2712.2011.01185.x>
- Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic Emergency Medicine*, 18(10), 1099-1104. <https://doi.org/10.1111/j.1553-2712.2011.01185.x>
- Wan, F. (2022). Conditional or unconditional logistic regression for frequency matched case-control design? *Statistics in Medicine*, 41(6), 1023-1041. <https://doi.org/10.1002/sim.9313>
- Wang, Q. Q., Yu, S. C., Qi, X., Hu, Y. H., Zheng, W. J., Shi, J. X. y Yao, H. Y. (2019). [Overview of logistic regression model analysis and application]. *Zhonghua Yu Fang Yi Xue Za Zhi*, 53(9), 955-960. <https://doi.org/10.3760/cma.j.issn.0253-9624.2019.09.018>
- Wiley, M. y Wiley, J. F. (2019). GLM2. En M. Wiley y J. F. Wiley, *Advanced R Statistical Programming and Data Models: Analysis, Machine Learning, and Visualization* (pp. 123-164). Apress.
- Zabor, E. C., Reddy, C. A., Tendulkar, R. D. y Patil, S. (2022). Logistic regression in clinical studies. *International Journal of Radiation Oncology, Biology, Physics*, 112(2), 271-277. <https://doi.org/10.1016/j.ijrobp.2021.08.007>

Curvas ROC

Alberto Beltrán Ramírez

Dentro de las Ciencias de la Salud, el uso de diversas pruebas para el diagnóstico de enfermedades o para la aplicación o no de cierto tratamiento es esencial e imperativo. Algunas de estas pruebas arrojan valores continuos como resultado, mismos que deben ser transformados de manera dicotómica y clasificados para poder determinar si el resultado es positivo o negativo. Las curvas ROC se utilizan para valorar el rendimiento de dichas pruebas.

Para este capítulo necesitaremos cargar las siguientes librerías:

```
library(ggplot2)
library(gridExtra)
library(dplyr)
```

Curvas ROC

Para entender la aplicación de las curvas ROC (característica operativa del receptor por su traducción del inglés), primero debemos definir los términos “sensibilidad y “especificidad”. La sensibilidad es la probabilidad de que una prueba diagnóstica sea positiva en un paciente que tiene la enfermedad; la especificidad es la probabilidad que tiene una prueba diagnóstica de ser negativa

en un paciente que no tiene la enfermedad. Los falsos positivos son aquellos individuos que no están enfermos, pero obtienen un resultado positivo, mientras que los falsos negativos son aquellos individuos que están enfermos, pero obtienen una prueba negativa (ver [tabla 24.1](#)).

TABLA 24.1. TABLA 2X2 PARA ESTIMAR SENSIBILIDAD Y ESPECIFICIDAD

CONDICIÓN PREDICHA			
Condición real		Prueba (+)	Prueba (-)
	Enfermedad (+)	a	b
	Enfermedad (-)	c	d

La curva característica operativa del receptor (ROC) se dibuja con el eje x como 1 - especificidad (falsos positivos) y el eje y como sensibilidad. Sensibilidad = $a/(a + b)$, especificidad = $d/(c + d)$, falsos negativos = $b/(a + b)$, falsos positivos = $c/(c + d)$ y precisión = $(a + d)/(a + b + c + d)$.

Las fórmulas matemáticas para el cálculo de estas medidas son:

$$\text{Sensibilidad} = \frac{VP}{VP+FN}$$

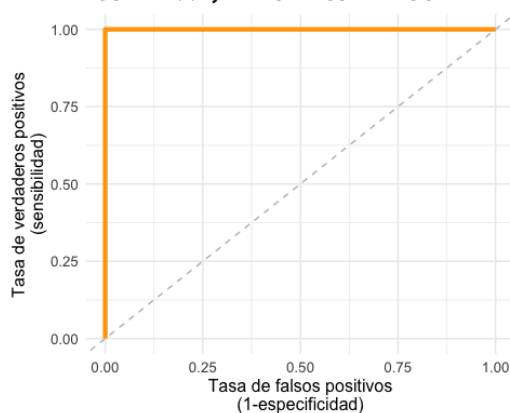
$$\text{Especificidad} = \frac{VP}{VNP+FP}$$

Donde: VP = verdaderos positivos; FN = falsos negativos; VN = verdaderos negativos y FP = falsos positivos.

Las curvas ROC son un método analítico de discriminación representado mediante una gráfica que muestra la habilidad diagnóstica de una prueba basada en el valor del umbral y la discriminación de todos los valores umbrales posibles.

De manera sencilla, una curva ROC es un gráfico que en el eje de las X representa 1 – *especificidad* (falsos positivos) y en el eje de las Y la sensibilidad (verdaderos positivos), así como el área bajo la curva (ABC). La [figura 24.1](#) muestra un ejemplo.

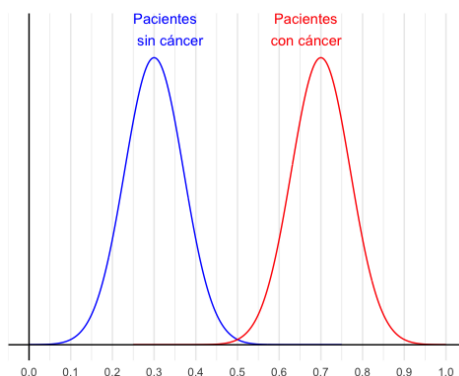
FIGURA 24.1. EJEMPLO DE CURVA ROC 1



Cuando una prueba diagnóstica es realizada, los individuos enfermos y los no enfermos no podrán agruparse de manera totalmente aislada, lo que significa que en algún punto se sobrepondrán.

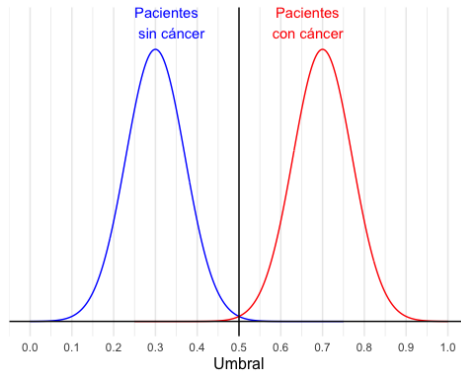
Por ejemplo, tenemos un modelo diagnóstico para saber si un individuo tiene cáncer o no ilustrado en la [figura 24.2](#), donde la línea azul representa a los individuos que no lo tienen, mientras que la línea roja representa a los individuos que sí. Ahora se debe elegir un valor establecido como punto de corte o umbral y que todos aquellos que presenten valores por encima de este serán definidos como positivos para cáncer, mientras que los que se encuentren por debajo serán definidos como negativos para cáncer.

FIGURA 24.2. EJEMPLO DE MODELO DIAGNÓSTICO 1



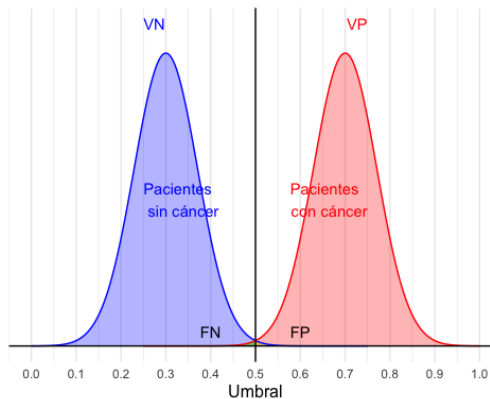
En este ejemplo, el umbral ha sido establecido en 0.5, por lo cual todos los valores positivos por encima de este serán los verdaderos positivos y los valores negativos por encima del umbral serán los falsos positivos. Todos los valores negativos por debajo del umbral serán los verdaderos negativos y los valores positivos por debajo de este serán los falsos negativos (ver [figura 24.3](#)).

FIGURA 24.3. EJEMPLO DE MODELO DIAGNÓSTICO 2



Como se mencionó previamente, los falsos positivos (FP) serán aquellos individuos que no están enfermos, pero que obtienen un resultado positivo, mientras que los falsos negativos (FN) son aquellos individuos que están enfermos, pero que obtienen una prueba negativa (ver [figura 24.4](#)).

FIGURA 24.4. EJEMPLO DE MODELO DIAGNÓSTICO 3



Si disminuimos el valor del umbral obtendremos más valores de los falsos negativos, aumentando la sensibilidad, pero disminuyendo la especificidad (ver [figura 24.5](#)). Por el contrario, si aumentamos el umbral obtenemos más valores de los falsos positivos, lo que aumenta la especificidad, pero disminuye la sensibilidad (ver [figura 24.6](#)).

FIGURA 24.5. EJEMPLO DE MODELO DIAGNÓSTICO 4

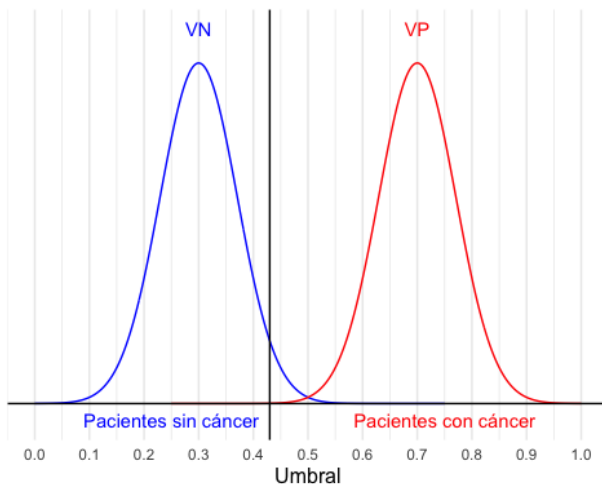
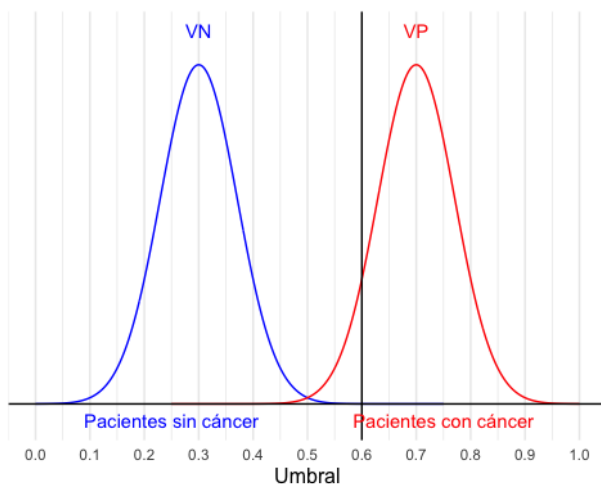
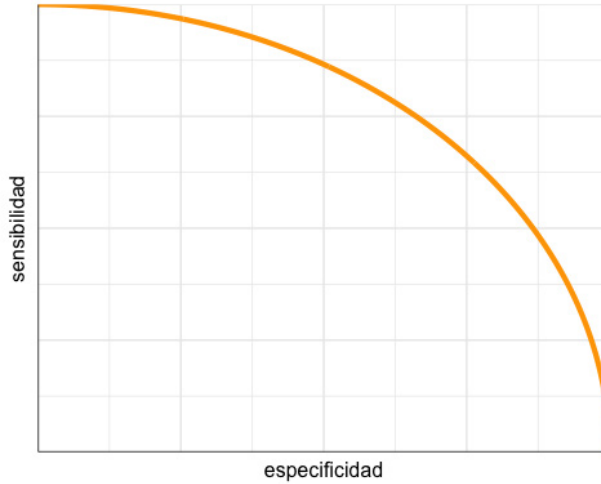
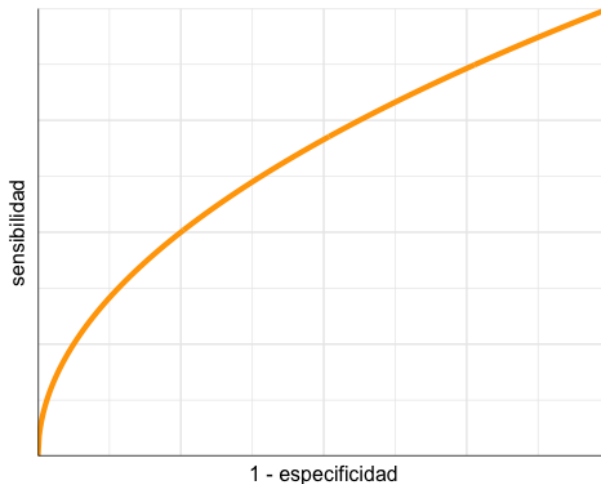


FIGURA 24.6. EJEMPLO DE MODELO DIAGNÓSTICO 5



Si graficamos esta relación que le acabamos de presentar, obtendríamos una curva que iría de arriba hacia abajo y de izquierda a derecha (ver [figura 24.7](#)). Sin embargo, para trazar una curva ROC se utiliza el parámetro 1 menos especificidad, lo que cambia la dirección de la curva de abajo hacia arriba manteniendo el desplazamiento de izquierda a derecha (ver [figura 24.8](#)).

FIGURA 24.7. EJEMPLO DE CURVA ROC 2**FIGURA 24.8. EJEMPLO DE CURVA ROC 3**

Debemos recordar que la especificidad nos muestra la tasa negativa verdadera, mientras que al utilizar “ $1 - \text{especificidad}$ ” en la curva ROC se nos mostrará la tasa de falsos positivos en el eje de las X y la sensibilidad (tasa de positivos verdaderos) en el eje de las Y:

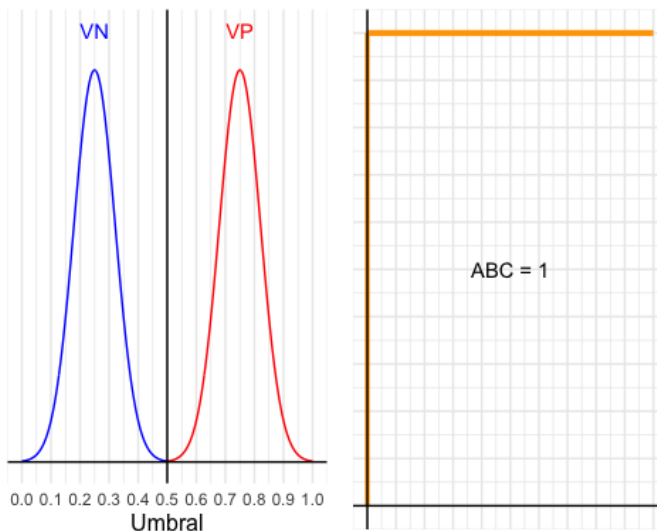
$$1 - \text{especificidad} = \frac{FP}{VN+FP}$$

Área bajo la curva (ABC)

El valor obtenido en este rubro nos permite conocer qué tan bien funcionan los resultados del modelo empleado. Mientras más cercano sea el valor a 1 mejor será el desempeño y por tanto una mayor utilidad.

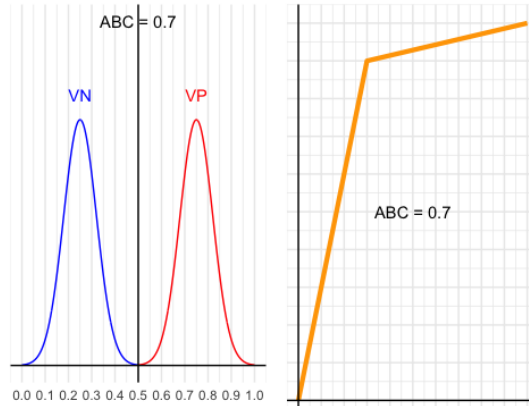
La situación ideal se ilustra en la [figura 24.9](#), donde las curvas de positivos y negativos no se superponen en lo absoluto. Esto significa que el modelo tendría una medida ideal de separación, siendo capaz de distinguir entre los individuos positivos y negativos.

FIGURA 24.9. SITUACIÓN IDEAL



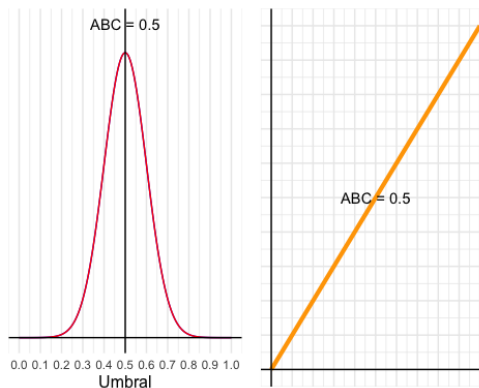
Cuando las distribuciones de positivos y negativos se superponen, se introducen errores en los modelos, los cuales pueden aumentar o disminuir dependiendo del umbral. Cuando un ABC tiene un valor de 0.7 significa que existe 70% de probabilidad de que el modelo distinga correctamente entre los positivos y negativos (ver [figura 24.10](#)).

FIGURA 24.10. ABC 0.7



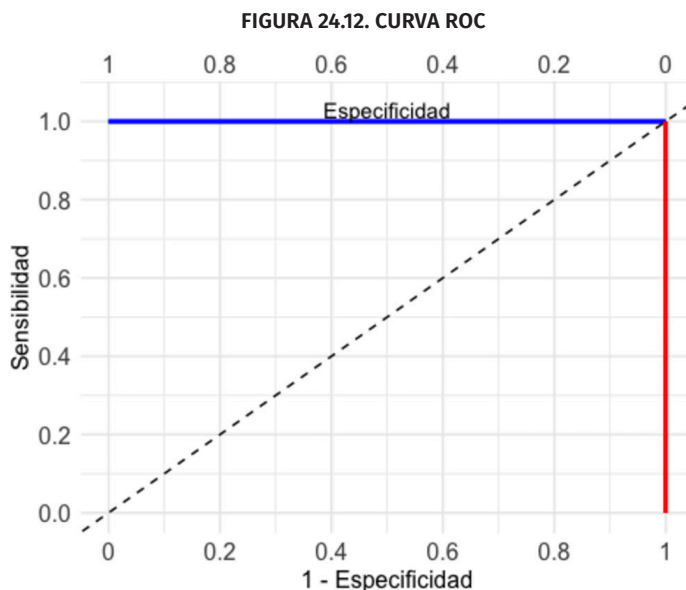
La peor situación a la que puede enfrentarse es cuando el ABC es de 0.5, ya que el modelo no posee la capacidad de distinguir entre los positivos y negativos (ver [figura 24.11](#)). Este 50% de probabilidad de discriminar los positivos de los negativos es igual a tirar una moneda al aire para definir si tienen la enfermedad o no.

FIGURA 24.11. ABC 0.5



La curva ROC cuantifica la capacidad de un indicador diagnóstico para discriminar entre enfermos y sanos. Recuerde que en la [figura 24.2](#) vemos en azul la distribución del indicador en los sanos y en rojo la de los enfermos. Al compartir ciertos valores del indicador la clasificación no puede ser perfecta, lo que indica que cualquier umbral o punto de corte clasificará erróneamente algunos de los casos de la zona superpuesta.

En la [figura 24.12](#) verá cómo aparece la curva ROC. Esta curva indica para cada posible umbral la proporción de positivos en ambas poblaciones: en el eje de las ordenadas para los enfermos y en el eje de las abscisas para los sanos. Así, la línea vertical roja indica la sensibilidad o proporción de positivos en los enfermos, la cual interesa que sea alta; por su parte, la línea horizontal azul indica la proporción de positivos en los sanos que conviene que sea baja; su complementario es la especificidad.

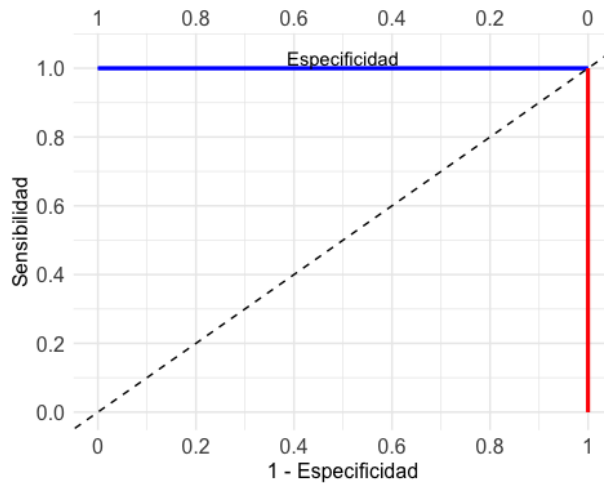


En la [figura 24.12](#) observe cómo la numeración es inversa en los ejes horizontales. En el eje inferior se muestra “” que, como se mencionó previamente, nos informará la proporción de positivos en los sanos (falsos positivos) y en el eje superior se muestra la especificidad su complementario.

Esto se traduce a que en el eje inferior en la proporción de positivos en los sanos (“ $1 - \text{especificidad}$ ”) si encontramos un valor de 10%, este equivaldrá en el eje superior a 90% de especificidad. Dicho de otra manera, por cada punto porcentual que aumente la tasa de falsos positivos, la tasa de verdaderos negativos descenderá un punto porcentual; es decir, son inversamente proporcionales.

En la [figura 24.13](#) situaremos el umbral en el extremo izquierdo de los valores. Si declaramos como positivos a todos los casos por encima de ese punto, tendríamos un 100% de positivos tanto para los sanos como para los enfermos. En la curva ROC (ver [figura 24.13](#)) estos valores del 100% se sitúan en el extremo superior derecho. La línea vertical roja situada a la derecha corresponde al número 1 en la escala de la izquierda que indica que el 100% de los enfermos son clasificados correctamente como positivos. Por su parte, la línea azul indica que el 100% de los sanos también están a la derecha del umbral y son clasificados como positivos, pero erróneamente. En términos sencillos, este umbral o punto de corte clasifica bien a todos los enfermos, pero mal a todos los sanos.

FIGURA 24.13. EJEMPLO DE CURVA ROC 4



Si desplazamos el umbral hacia la derecha, se mostrará en la curva ROC cómo varían las proporciones, disminuyendo el error en el momento de clasificar a los sanos, pero empeorando al clasificar a los enfermos. Al continuar con

el aumento o desplazamiento del umbral hacia la derecha, podemos observar cómo descende la proporción de positivos en los sanos y cómo el acierto en los enfermos se mantiene igual. Sin embargo, al mantener el aumento del umbral hacia la derecha, obtenemos menos positivos en los sanos, pero permitiendo catalogar algunos negativos en los enfermos. En concreto, primero aumentará la especificidad y después bajará la sensibilidad (ver [figura 24.14](#) a [figura 24.17](#))

FIGURA 24.14. EJEMPLO DE CURVA ROC 5

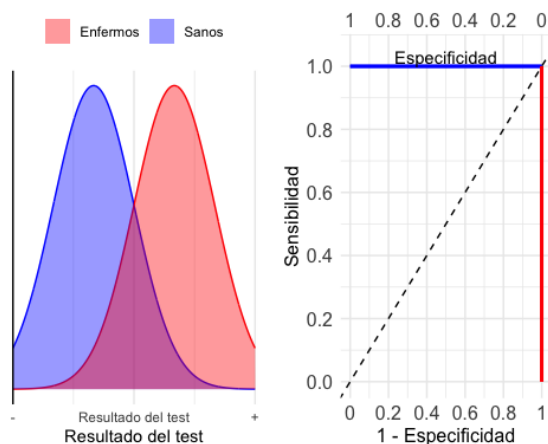


FIGURA 24.15. EJEMPLO DE CURVA ROC 6

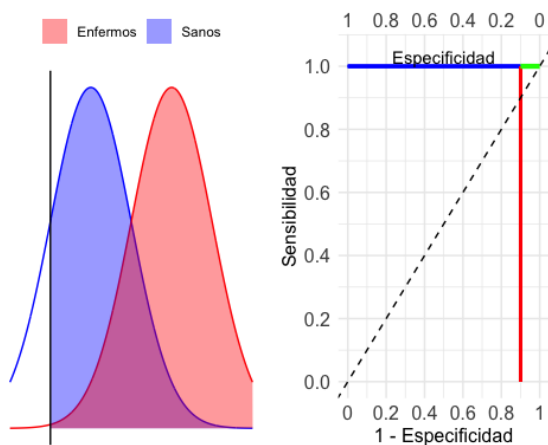


FIGURA 24.16. EJEMPLO DE CURVA ROC 7

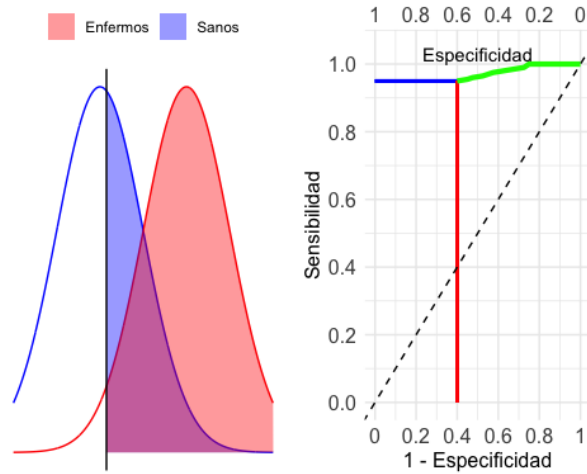
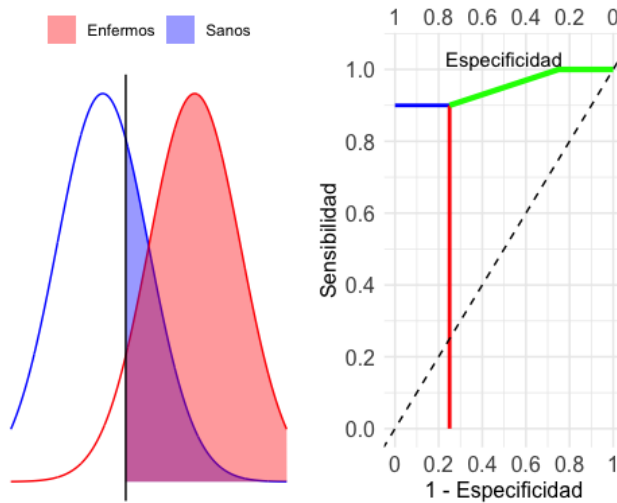
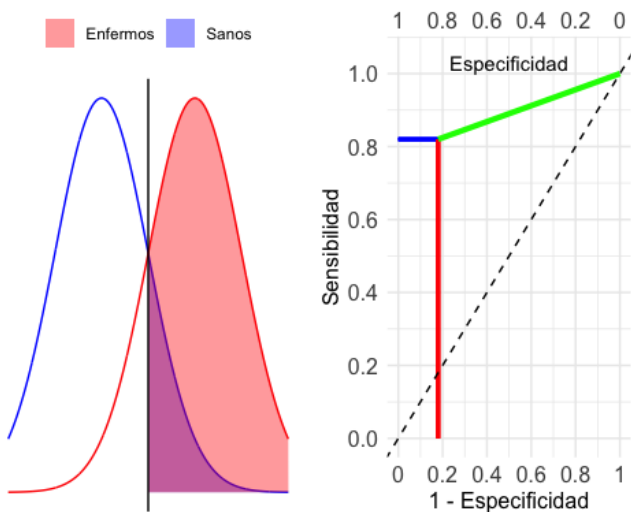


FIGURA 24.17. EJEMPLO DE CURVA ROC 8



En la [figura 24.18](#) se muestra un punto de corte donde la sensibilidad y especificidad son iguales, concretamente 0.82, clasificando de manera correcta la misma proporción de enfermos y sanos. Cabe recalcar que en este punto se ha ganado 82% de especificidad y solo se ha perdido 18% de sensibilidad.

FIGURA 24.18. EJEMPLO DE CURVA ROC 9



En este punto, debido al rápido descenso de la sensibilidad, si continúa con el aumento del umbral, se aproximará al punto en el que todos los casos serían declarados como negativos, por lo que ambas proporciones de positivos serían de 0, lo que sería excelente para los sanos, pero pésimo para los enfermos (ver [figura 24.19](#) a [figura 24.25](#)).

FIGURA 24.19. EJEMPLO DE CURVA ROC 10

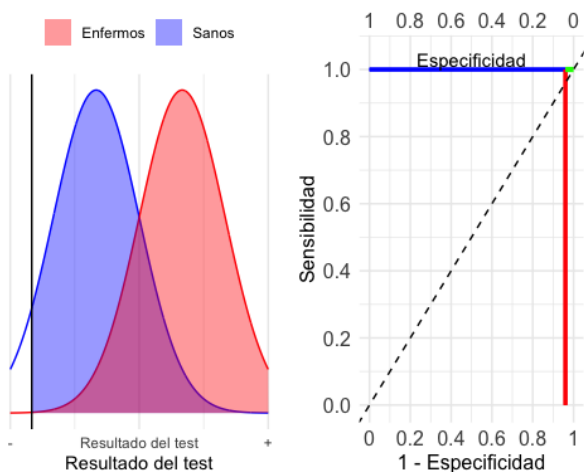


FIGURA 24.20. EJEMPLO DE CURVA ROC 11

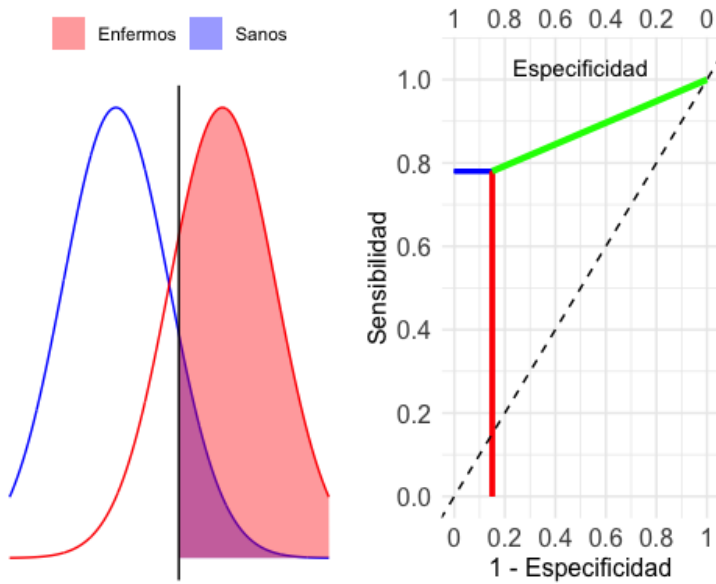


FIGURA 24.21. EJEMPLO DE CURVA ROC 12

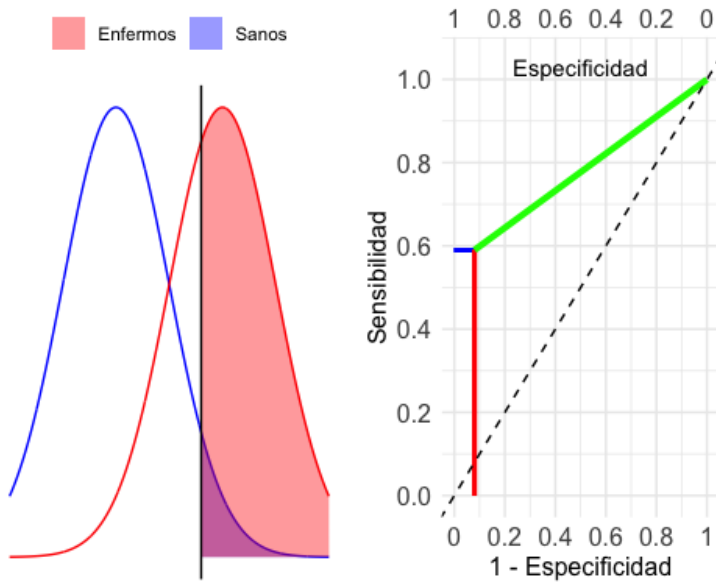


FIGURA 24.22. EJEMPLO DE CURVA ROC 13

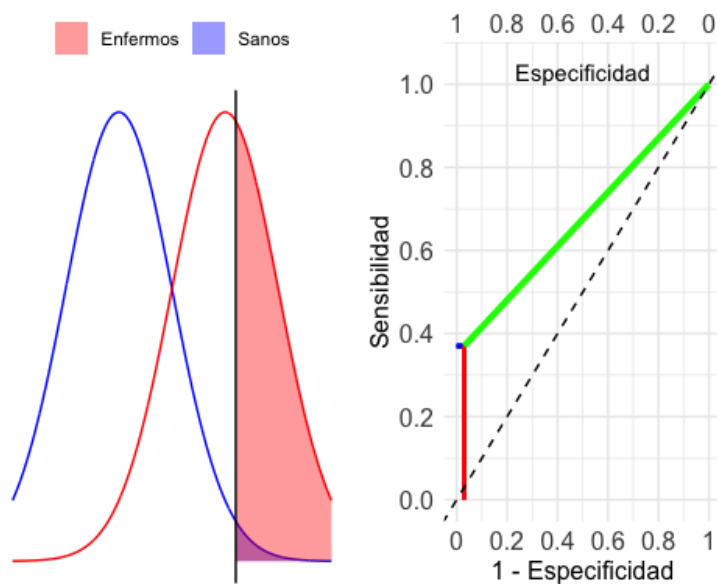


FIGURA 24.23. EJEMPLO DE CURVA ROC 14

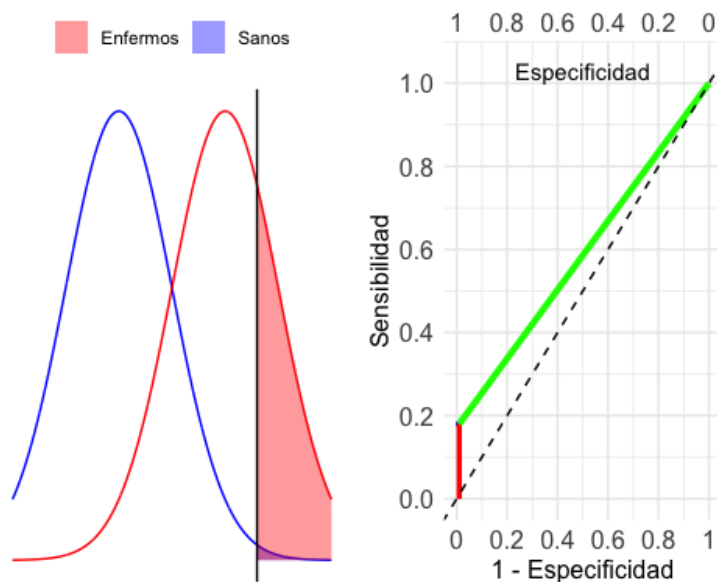


FIGURA 24.24. EJEMPLO DE CURVA ROC 15

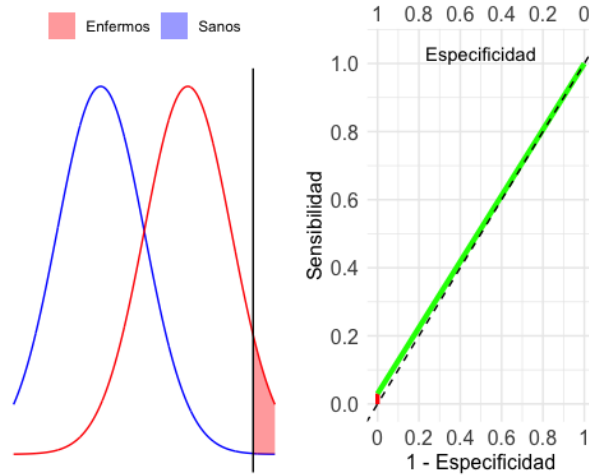
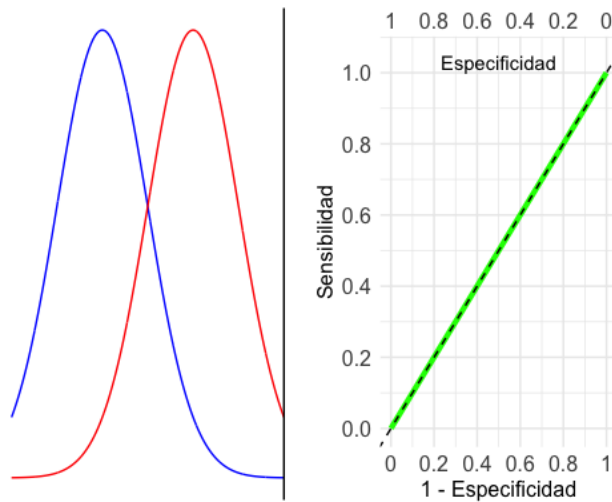


FIGURA 24.25. EJEMPLO DE CURVA ROC 16

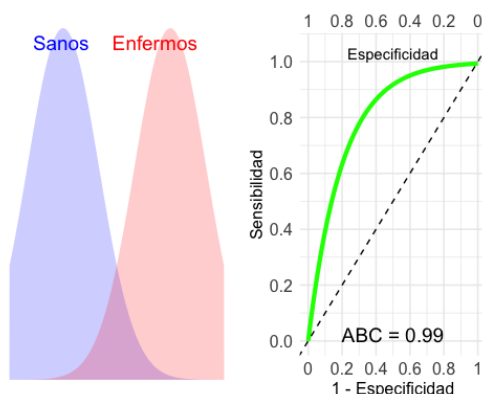


De esta forma, la curva ROC muestra simultáneamente la sensibilidad y especificidad del umbral.

Si las distribuciones de negativos y positivos casi no se superpusieran, el umbral sería capaz de discriminar muy bien a los sanos de los enfermos y la curva ROC cubriría o tendría un área bajo la curva muy grande. En la [figura](#)

24.26 el área bajo la curva es de 0.99 o 99%, lo que indica que la probabilidad de que un enfermo obtenga un valor más extremo que un sano es del 99%; o también que de todas las posibles parejas formadas por un enfermo y un sano en 99% de ellas el enfermo tiene un valor más patológico que el sano.

FIGURA 24.26. EJEMPLO DE CURVA ROC 17



Mientras más se superpongan las curvas, más se parecerán los resultados de sanos y enfermos. La capacidad del umbral de discriminar será más baja y, por tanto, más baja es la probabilidad de que un enfermo obtenga un valor mayor que un sano (ver figura 24.27 a figura 24.29).

FIGURA 24.27. EJEMPLO DE CURVA ROC 18

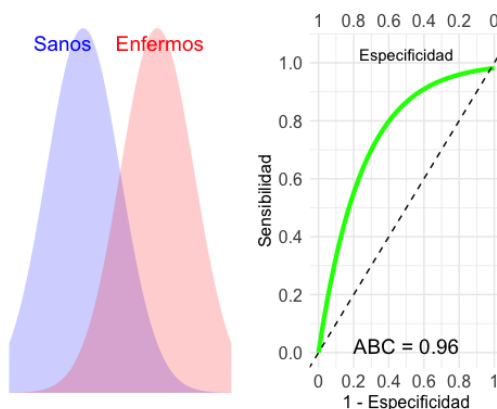


FIGURA 24.28. EJEMPLO DE CURVA ROC 19

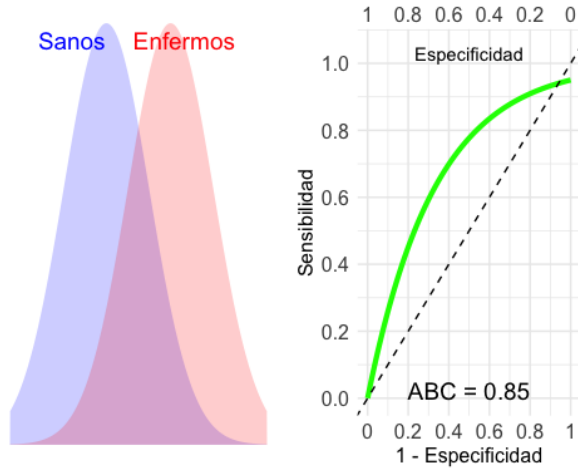
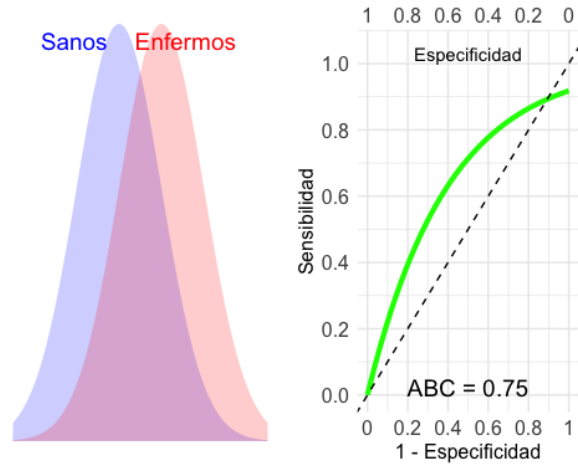


FIGURA 24.29. EJEMPLO DE CURVA ROC 20



En este punto debe aclararse que la curva ROC no determina un punto de corte, sino las proporciones de positivos que tendría cada valor del umbral. Asimismo, tampoco nos dice qué valor del umbral debemos escoger. Debe saberse también que los gráficos utilizados en los ejemplos han asumido una distribución normal de los datos, pero es útil ante cualquier tipo de distribución.

Índice de Youden

Es uno de los varios métodos que existen para establecer el mejor punto de corte para la discriminación dicotómica de la curva ROC, obtenido de la resta:

$$\text{Índice de Youden} = \text{Sensibilidad} - (1 - \text{Especificidad})$$

Y cuyos valores entran en el rango de +1 a -1, donde el valor ideal es +1, lo que indicaría que una prueba contaría con 100% de sensibilidad y 0% de falsos positivos (Roy-García *et al.*, 2023; Martínez-Cambor y Pardo-Fernández, 2019).

Roy-García *et al.* (2023) en su artículo “Curvas ROC: características generales y su uso en la práctica clínica” utilizan los siguientes datos para ejemplificar el uso del índice de Youden.

La [tabla 24.2](#) muestra el desempeño de cada umbral en forma individual. En este ejemplo se utilizarán niveles de antígeno prostático específico (APE).

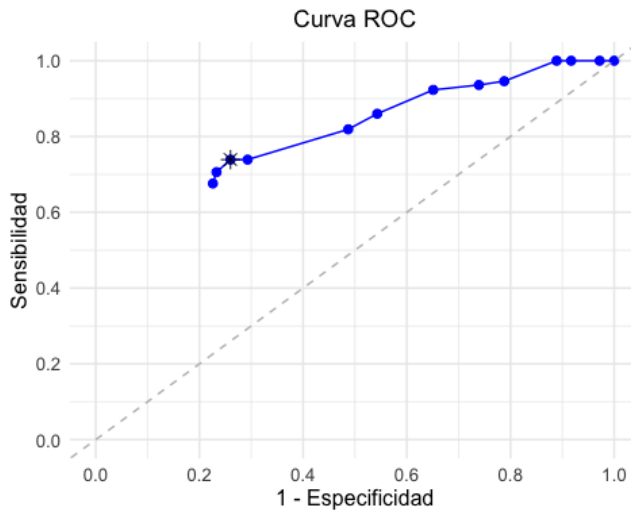
TABLA 24.2 RENDIMIENTO DIAGNÓSTICO DEL APE EN DIFERENTES PUNTOS DE CORTE

ANTÍGENO PROSTÁTICO (NG/ML)	SENSIBILIDAD	1 – ESPECIFICIDAD (FALSOS POSITIVOS)	ÍNDICE DE YOUTEN
3.4	1.000	1.000	0.000
5.1	1.000	0.972	0.028
6.7	1.000	0.917	0.083
7.4	1.000	0.889	0.111
8.2	0.946	0.788	0.158
9.1	0.936	0.739	0.197
10.1	0.923	0.651	0.272
12.8	0.860	0.543	0.317
14.3	0.819	0.487	0.332
19.8	0.739	0.293	0.446
20.7	0.739	0.260	0.479
22.0	0.706	0.233	0.470
24.2	0.676	0.226	0.450

Con un valor de APE de 3.4 ng/mL la sensibilidad que se tiene es del 100%, mismo valor que en los falsos positivos, lo que hace que descartemos este valor como punto de corte, ya que no es capaz de distinguir entre sanos y enfermos obteniendo un valor de 0.0 en el índice de Youden. Si utilizamos el punto de corte de 9.1 tendremos una sensibilidad de 93.6%, pero la cantidad de falsos positivos permanece con un porcentaje alto (73.9%) con un valor de 0.197 en el índice de Youden; mientras que con un punto de corte de 20.7 tenemos una sensibilidad del 73.9% y un 26% de falsos positivos, así como un índice de Youden de 0.479, siendo el más alto de todos los presentados y por tanto determina el mejor punto de corte.

Este punto de corte presentará un área bajo de la curva de 0.769 o 76.9% (ver [figura 24.30](#)).

FIGURA 24.30. EJEMPLO DE CURVA ROC 21



Curvas ROC en R

- ♦ **Ejemplo 24.1:** la base de datos “Base_ROC.csv”, contiene datos de 165 pacientes que padecen cierta enfermedad. A todos se les tomaron muestras de sangre para obtener valores de 3 biomarcadores relacionados con

esta enfermedad, denominados biomarcador 1, 2 y 3, obteniendo valores diversos tanto para cada paciente como para cada biomarcador. Contiene las siguientes variables:

- » Paciente: el número consecutivo de los pacientes incluidos en la base.
- » Biomarcador_1: el valor obtenido en sangre de un biomarcador determinado.
- » Biomarcador_2: el valor obtenido en sangre de un segundo biomarcador determinado.
- » Biomarcador_3: el valor obtenido en sangre de un tercer biomarcador determinado.
- » Diagnóstico: si el paciente es sano o enfermo.

Utilizando la base de datos [Base_ROC.csv](#), genere una curva ROC para poder definir el mejor punto de corte para el biomarcador 1. Haga un gráfico con la información de esta tabla y genere un archivo .csv que muestre la sensibilidad, especificidad e índice de Youden para cada punto de corte.

El siguiente código en R puede ser de ayuda para resolver el problema:

```
# Cargar las librerías necesarias
library(pROC) # Para generar la curva ROC
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'
The following objects are masked from 'package:stats':

    cov, smooth, var

# Leer los datos
data <- read.csv("Bases/Base_ROC.csv")

# Convertir la columna de diagnóstico a un factor con niveles 'EN-
FERMO' y 'SANO'
data$Diagnostico <- factor(data$Diagnostico, levels = c("SANO", "ENFERMO"))

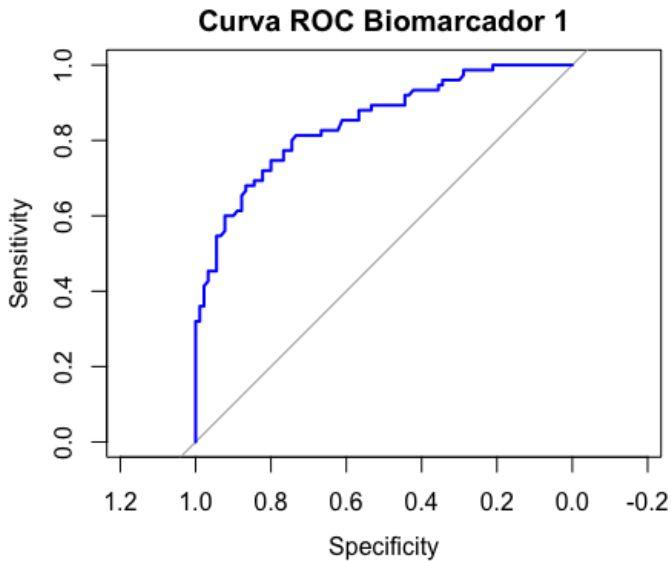
# Generar la curva ROC
```

```

roc <- roc(response = data$Diagnostico, predictor = data$Biomarcador_1)
Setting levels: control = SANO, case = ENFERMO
Setting direction: controls < cases
# Generar un gráfico de la curva ROC

plot(roc, main = "Curva ROC Biomarcador 1", col = "blue", lwd = 2)

```



```
# Obtener los datos de la curva ROC
```

```
print(roc)
```

Call:

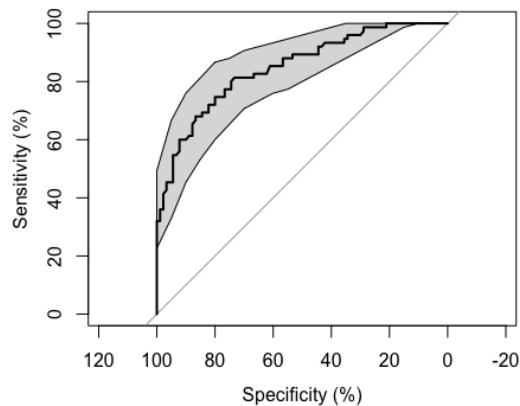
```
roc.default(response = data$Diagnostico, predictor = data$Biomarcador_1)
```

Data: data\$Biomarcador_1 in 90 controls (data\$Diagnostico SANO) < 75 cases (data\$Diagnostico ENFERMO).

Area under the curve: 0.8478

Podemos generar algunos gráficos adicionales para evaluar la curva ROC. El siguiente código muestra cómo realizar un gráfico de la curva ROC con intervalos de confianza:

```
plot(
  x = roc(
    response = data$Diagnostico,          # Evento.
    predictor = data$Biomarcador_1,       # biomarcador que se está
    evaluando p
    percent = TRUE,                      # Porcentaje de sensibi-
    lidad y especificidad.
    ci = TRUE,                           # Estimación de los IC 95%
    of = "se",                           # Cálculo del IC sensi-
    bilidad.
    sp = seq(0, 100, 5)                  # Del 0 al 100 de 5 en 5
  ),
  ci.type = "shape"                      # Forma en la que se
  mostrarn los IC
)
Setting levels: control = SANO, case = ENFERMO
Setting direction: controls < cases
```



También podemos obtener los valores de sensibilidad, especificidad e índice de Youden para cada punto de corte:

```
roc_df <- data.frame(
  sens = roc$sensitivities,          # Sensibilidad
  spec = 1 - roc$specificities,      # 1 - Especificidad (falsos positivos)
  Youden = roc$sensitivities + (1 - roc$specificities) - 1 # Índice de Youden
)

# Primeras 6 filas.
head(roc_df)
```

	sens	spec	Youden
1	1	1.0000000	1.0000000
2	1	0.9888889	0.9888889
3	1	0.9777778	0.9777778
4	1	0.9666667	0.9666667
5	1	0.9555556	0.9555556
6	1	0.9444444	0.9444444

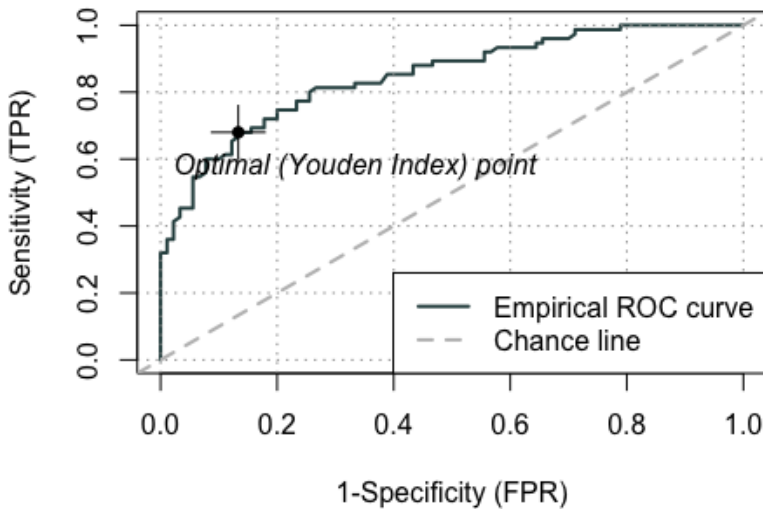
```
# Guardar el dataframe `roc_df` en un archivo CSV
write.csv(roc_df, "ROC_Biomarcador_1.csv", row.names = FALSE)
```

La explicación del código anterior es la siguiente:

- ♦ Se creó un *Dataframe* llamado `roc_df` que contiene tres columnas derivadas de un objeto ROC previamente calculado. Este objeto ROC contiene información sobre la sensibilidad y especificidad de un modelo. Las columnas del *Dataframe* son:
 1. `sens`: contiene los valores de sensibilidad (verdaderos positivos) extraídos del objeto `roc`.
 2. `spec`: contiene los valores de $(1 - \text{especificidad})$, lo que representa los falsos positivos.
 3. `Youden`: se calcula el índice de Youden, que es una métrica que se utiliza para encontrar el punto óptimo de corte en un ROC.
- ♦ Este índice se define como $\text{sensibilidad} + \text{especificidad} - 1$, y en este caso, la especificidad se está restando de 1 para facilitar su uso.

Finalmente, utilizando la librería ROCit podemos generar otro gráfico para conocer el punto óptimo de corte:

```
install.packages("ROCit") # Instalar la librería ROCit
library(ROCit) # Llamamos a la librería
# Objeto con la curva ROC
Punto_Optimo <- rocit(class = as.numeric(data$Diagnostico), score =
data$Biomarcador_1)
# Graficar
plot(Punto_Optimo)
```



A continuación, se explica el código anterior:

- ♦ Se creó un objeto llamado `Punto_Optimo` que contiene la información de la curva ROC.
- ♦ La función `rocit` toma dos argumentos principales:
 - » `class`: es la variable que contiene la clasificación binaria o categórica del diagnóstico (por ejemplo, sano/enfermo).
 - » Es necesario.

- » score: es el valor de los biomarcadores (por ejemplo, un marcador bioquímico) que será usado para clasificar los datos en la curva ROC.
 - » En este caso, los valores de `data$Biomarcador_1` se utilizan como puntuaciones o predicciones continuas.
- ♦ Se realizó una gráfica de la curva ROC, utilizando la función `plot` y pasando como argumento el objeto `Punto_Optimo`.

Ejercicios

- ♦ **Ejercicio 24.1:** utilizando la base de datos [Base_ROC.csv](#), genere una curva ROC, para poder definir el mejor punto de corte para el biomarcador 2. Haga un gráfico con la información de esta tabla y genere un archivo .csv que muestre la sensibilidad, especificidad e índice de Youden para cada punto de corte.

Resolución ejercicio 24.1

```
roc <- roc(response = data$Diagnostico, predictor = data$Biomarcador_2)
plot(roc, main = "Curva ROC Biomarcador 2", col = "blue", lwd = 2)
print(roc)
plot(x = roc(response = data$Diagnostico, predictor = data$Biomarcador_1, percent = TRUE, ci = TRUE, of = "se", sp = seq(0, 100, 5)), ci.type="shape")
roc_df <- data.frame(
  sens = roc$sensitivities,
  spec = 1 - roc$specificities,
  Youden = roc$sensitivities + (1 - roc$specificities) - 1 )
head(roc_df)
write.csv(roc_df, "ROC_Biomarcador_2.csv", row.names = FALSE)
Punto_Optimo <- rocit(class = as.numeric(data$Diagnostico), score = data$Biomarcador_1)
plot(Punto_Optimo)
```

- ♦ **Ejercicio 24.2:** utilizando la base de datos `Base_ROC.csv`, genere una curva ROC, para poder definir el mejor punto de corte para el biomarcador 3. Haga un gráfico con la información de esta tabla y genere un archivo .csv que muestre la sensibilidad, especificidad e índice de Youden para cada punto de corte.

Resolución ejercicio 24.2

```
roc <- roc(response = data$Diagnostico, predictor = data$Biomarcador_3)
plot(roc, main = "Curva ROC Biomarcador 3", col = "blue", lwd = 2)
print(roc)
plot(x = roc(response = data$Diagnostico, predictor = data$Biomarcador_1, percent = TRUE, ci = TRUE, of = "se", sp = seq(0, 100, 5)), ci.type="shape")
roc_df <- data.frame(
  sens = roc$sensitivities,
  spec = 1 - roc$specificities,
  Youden = roc$sensitivities + (1 - roc$specificities) - 1 )
head(roc_df)
write.csv(roc_df, "ROC_Biomarcador_2.csv", row.names = FALSE)
Punto_Optimo <- rocit(class = as.numeric(data$Diagnostico), score = data$Biomarcador_1)
plot(Punto_Optimo)
```

Referencias

- Fong, Y., Yin, S., y Huang, Y. (2016). Combining biomarkers linearly and nonlinearly for classification using the area under the ROC curve. *Statistics in Medicine*, 35(21), 3792--3809. <https://doi.org/10.1002/sim.6977>
- Hillis, S. L. (2016). Equivalence of binormal likelihood--ratio and bi--chi--squared ROC curve models. *Statistics in Medicine*, 35(12), 2031--2057. <https://doi.org/10.1002/sim.6857>
- Janssens, A. C. J. W., y Martens, F. K. (2020). Reflection on modern methods: Revisiting the area under the ROC Curve. *International Journal of Epidemiology*, 49(4), 1397--1403. <https://doi.org/10.1093/ije/dyaa101>

- Kim, J., y Hwang, I. C. (2020). Drawing Guidelines for Receiver Operating Characteristic Curve in Preparation of Manuscripts. *Journal of Korean Medical Science*, 35(24), e171. <https://doi.org/10.3346/jkms.2020.35.e171>
- Kumar, R., y Indrayan, A. (2011). Receiver operating characteristic (ROC) curve for medical researchers. *Indian Pediatrics*, 48(4), 277--287. <https://doi.org/10.1007/s13312-011-0075-1>
- Mandrekar, J. N. (2010). Receiver Operating Characteristic Curve in Diagnostic Test Assessment. *Journal of Thoracic Oncology*, 5(9), 1315--1316. <https://doi.org/10.1097/JTO.0b013e3181ec173d>
- Martínez-Cambor, P., y Pardo-Fernández, J. C. (2019). The Youden Index in the Generalized Receiver Operating Characteristic Curve Context. *The International Journal of Biostatistics*, 15(1), 20180060. <https://doi.org/10.1515/ijb-2018-0060>
- Moscarelli, M. (2023). *Biostatistics With "R": A Guide for Medical Doctors* [Internet]. Cham: Springer International Publishing. Retrieved July 11, 2024, from <https://link.springer.com/10.1007/978-3-031-33073-5>
- Mossman, D. (1999). Three-way ROCs. *Medical Decision Making*, 19(1), 78--89. <https://doi.org/10.1177/0272989X9901900108>
- Nahm, F. S. (2022). Receiver operating characteristic curve: Overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 75(1), 25--36. <https://doi.org/10.4097/kja.21227>
- Obuchowski, N. A. (2005). *Fundamentals of Clinical Research for Radiologists*.
- Park, S. H., Goo, J. M., y Jo, C. H. (2004). Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists. *Korean Journal of Radiology*, 5(1), 11--18. <https://doi.org/10.3348/kjr.2004.5.1.11>
- Pepe, M. S. (2000). An Interpretation for the ROC Curve and Inference Using GLM Procedures. *Biometrics*, 56(2), 352--359. <https://doi.org/10.1111/j.0006-341X.2000.00352.x>
- Riffenburgh, R. H. (2006). *Statistics in Medicine* (2nd ed.). Amsterdam: Elsevier Academic Press. Capítulo 15, 241-279.
- Rousson, V., y Zumbunn, T. (2011). Decision curve analysis revisited: Overall net benefit, relationships to ROC curve analysis, and application to case-control studies. *BMC Medical Informatics and Decision Making*, 11(1), 45. <https://doi.org/10.1186/1472-6947-11-45>

- Roy-García, I. A., Paredes-Manjarrez, C., Moreno-Palacios, J., Rivas-Ruiz, R., y Flores-Pulido, A. A. (2023). ROC curves: General characteristics and their usefulness in clinical practice. Retrieved June 5, 2024, from <https://zenodo.org/doi/10.5281/zenodo.8319791>
- Saito, T., y Rehmsmeier, M. (2017). Precrec: Fast and accurate precision–recall and ROC curve calculations in R. *Bioinformatics*, 33(1), 145–147. <https://doi.org/10.1093/bioinformatics/btw570>
- Staartjes, V. E., Regli, L., y Serra, C. (Eds.). (2022). *Machine Learning in Clinical Neuroscience: Foundations and Applications* [Internet]. Cham: Springer International Publishing. (Acta Neurochirurgica Supplement; vol. 134). Retrieved June 5, 2024, from <https://link.springer.com/10.1007/978-3-030-85292-4>
- Stevenson, M. (2019). epiR: Tools for the Analysis of Epidemiological Data [Internet]. Retrieved from <https://CRAN.R-project.org/package=epiR>
- Trebach, J., y Su, M. K. (2022). Biostatistics and Epidemiology for the Toxicologist: Rock the ROC Curve. *Journal of Medical Toxicology*, 18(2), 163–167. <https://doi.org/10.1007/s13181-021-00845-7>
- Vexler, A., Yu, J., Zhao, Y., Hutson, A. D., y Gurevich, G. (2018). Expected p-values in light of an ROC curve analysis applied to optimal multiple testing procedures. *Statistical Methods in Medical Research*, 27(12), 3560–3576. <https://doi.org/10.1177/0962280218755402>
- Yang, Y., y Smith, A. M. (2022). fullROC: An R package for generating and analyzing eyewitness-lineup ROC curves. *Behavior Research Methods*, 55(3), 1259–1274. <https://doi.org/10.3758/s13428-022-01860-2>

Análisis de los componentes principales

Edsaúl Emilio Pérez Guerrero

El análisis multivariado es un conjunto de técnicas estadísticas que se utilizan para analizar conjuntos de datos que contienen múltiples variables (todas al mismo tiempo). A diferencia de las técnicas estadísticas clásicas (como una prueba t de Student, ANOVA, regresión lineal simple, etc.), que se utilizan para analizar una variable a la vez, las técnicas multivariadas se utilizan para analizar múltiples variables simultáneamente. Estas técnicas buscan entender cómo se relacionan varias variables dependientes entre sí y con un conjunto de variables independientes o explicativas. Es importante que el lector no confunda los términos “multivariado” y “multivariable”. El segundo se refiere a técnicas en las que se estudia una única variable dependiente en función de múltiples variables independientes o predictoras (por ejemplo, la regresión lineal múltiple). En este contexto, se busca entender el efecto de cada variable independiente sobre la variable dependiente mientras se controla por las otras variables en el modelo.

En este capítulo, nos centramos en una de las técnicas de análisis multivariado más utilizada: el análisis de componentes principales (ACP).

¿Cuándo se requiere realizar un análisis multivariado?

La [tabla 25.1](#) muestra los primeros 12 datos del *Dataframe* Bodyfat. Esta base de datos tiene en total 15 variables relacionadas con mediciones de antropometría. Se puede observar, además, que no existe ninguna variable para clasificar a los

pacientes, solo tenemos mediciones (variables numéricas). Este es un claro ejemplo que los datos requieren un enfoque multivariado. Por su parte, la [tabla 25.2](#) muestra los datos de expresión de 12 genes para 6 células. En este caso en específico hay más variables que casos, por lo que es necesario tratar a los datos antes de realizar pruebas de hipótesis o de llegar a otras conclusiones. Este es otro claro ejemplo de que los datos deben de ser tratados utilizando estadística multivariada.

TABLA 25.1. VISIÓN DE LOS PRIMEROS 12 DATOS DE LA BASE DE DATOS BODYFAT

	Density	Fat	Age	Weight	Height	Neck
1	1.0708	12.3	23	154.25	67.75	36.2
2	1.0853	6.1	22	173.25	72.25	38.5
3	1.0414	25.3	22	154.00	66.25	34.0
4	1.0751	10.4	26	184.75	72.25	37.4
5	1.0340	28.7	24	184.25	71.25	34.4
6	1.0502	20.9	24	210.25	74.75	39.0
7	1.0549	19.2	26	181.00	69.75	36.4
8	1.0704	12.4	25	176.00	72.50	37.8
9	1.0900	4.1	25	191.00	74.00	38.1
10	1.0722	11.7	23	198.25	73.50	42.1
11	1.0830	7.1	26	186.25	74.50	38.5
12	1.0812	7.8	27	216.00	76.00	39.4

TABLA 25.2. VISIÓN DE LA BASE DE DATOS DE EXPRESIÓN GÉNICA

	Gene_1	Gene_2	Gene_3	Gene_4	Gene_5	Gene_6
Cell_1	0.9148060	0.7365883	0.9346722	0.4749971	0.08243756	0.737595618
Cell_2	0.9370754	0.1346666	0.2554288	0.5603327	0.51421178	0.811055141
Cell_3	0.2861395	0.6569923	0.4622928	0.9040314	0.39020347	0.388108283
Cell_4	0.8304476	0.7050648	0.9400145	0.1387102	0.90573813	0.685169729
Cell_5	0.6417455	0.4577418	0.9782264	0.9888917	0.44696963	0.003948339
Cell_6	0.5190959	0.7191123	0.1174874	0.9466682	0.83600426	0.832916080

	Gene_7	Gene_8	Gene_9	Gene_10	Gene_11	Gene_12
Cell_1	0.007334147	0.03743103	0.9709666	0.03893649	0.6756073	0.2712866
Cell_2	0.207658973	0.97353991	0.6188382	0.74879539	0.9828172	0.8281585
Cell_3	0.906601408	0.43175125	0.3334272	0.67727683	0.7595443	0.6932048
Cell_4	0.611778643	0.95757660	0.3467482	0.17126433	0.5664884	0.2405447
Cell_5	0.379559241	0.88775491	0.3984854	0.26108796	0.8496897	0.0429888
Cell_6	0.435771585	0.63997877	0.7846928	0.51441293	0.1894739	0.1404791

En ambos casos, la estadística multivariada es útil para explorar la estructura de los datos y encontrar patrones ocultos en ellos. En el caso de la base de datos de antropometría, el ACP se puede utilizar para reducir la dimensionalidad de los datos y encontrar patrones ocultos en ellos. En el caso de la base de datos de expresión génica, el ACP se puede utilizar para identificar patrones de expresión génica y encontrar genes que se expresan de manera similar en diferentes células.

Otras situaciones en las que se utiliza estadística multivariada podrían ser:

- ♦ Explorar relaciones complejas: cuando se tiene múltiples variables que podrían estar correlacionadas o interrelacionadas entre sí, y se desea entender cómo interactúan de manera conjunta. Por ejemplo, el *DataFrame* Bodyfat
- ♦ Reducción de dimensionalidad: si se tiene un conjunto de datos con muchas variables (dimensiones), el análisis multivariado no puede ayudar a reducir la cantidad de variables sin perder demasiada información. Por ejemplo, la base de datos con expresión génica podría abordarse con este enfoque.
- ♦ Segmentación o clasificación: en situaciones donde se necesita clasificar o segmentar individuos u objetos en grupos basados en múltiples características (por ejemplo, segmentación de pacientes por perfiles clínicos) las técnicas como el análisis de clúster o análisis discriminante son apropiadas. Este enfoque lo revisaremos en el [capítulo 27](#).
- ♦ Análisis de patrones y tendencias: cuando se desea identificar patrones ocultos, tendencias o estructuras subyacentes en un conjunto de datos, como en estudios de comportamiento, análisis epidemiológicos o encuestas. Por ejemplo, el análisis factorial que se verá en [capítulo 26](#).

Existen otras situaciones en las que se utiliza el análisis multivariado, como por ejemplo los modelos predictivos, que se escapan de los objetivos de este libro.

Introducción al análisis de componentes principales

El ACP es una técnica de reducción de la dimensionalidad que se utiliza para reducir la complejidad de los datos, manteniendo la mayor cantidad de información posible. En otras palabras, el ACP se utiliza para transformar un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas (llamadas componentes principales), que explican la mayor cantidad de variabilidad en los datos. El ACP es una técnica no supervisada, lo que significa que no requiere una variable dependiente para realizar el análisis. En cambio, el ACP se utiliza para explorar la estructura de los datos y encontrar patrones ocultos en ellos.

Vamos a tomar un ejemplo de la vida común para comprender qué es la reducción de dimensionalidad aplicada en el ACP. Supongamos que usted recibe una invitación para ir al cine a ver una película en 3D. De esta manera, usted puede comprender la película perfectamente, ya que es capaz de percibir la altura, el ancho y la profundidad (las tres dimensiones). Sin embargo, días después regresa al cine, pero esta vez decide ver una película en 2D (ancho y profundidad); su cerebro, a pesar de percibir una dimensión menos, es capaz de procesar la imagen y comprender perfectamente la película. En otras palabras, su cerebro ha reducido la dimensión de la imagen en 3D a una imagen en 2D y ha conservado la información. El ACP hace algo similar con los datos: toma un conjunto de datos en un espacio de alta dimensión y lo proyecta en un espacio de menor dimensión, conservando la mayor cantidad de información posible. Toma un conjunto de datos en un espacio de alta dimensión y lo proyecta en un espacio de baja dimensión, manteniendo la mayor cantidad de información posible.

¿Cuándo se utiliza el análisis de componentes principales?

Antes de entrar en los conceptos y en la realización de una ACP, describiremos en qué situaciones se utiliza esta maravillosa técnica:

1. Reducción de la dimensionalidad: el ACP se utiliza para reducir la cantidad de variables en un conjunto de datos, manteniendo la mayor cantidad

de información posible. Esto es útil cuando se trabaja con conjuntos de datos que contienen muchas variables y se desea reducir la complejidad de los datos.

2. Identificación de patrones: el ACP se utiliza para identificar patrones en los datos y encontrar relaciones entre las variables. Esto es útil cuando se desea explorar la estructura de los datos y encontrar patrones ocultos en ellos.
3. Compresión de datos: el ACP se utiliza para comprimir datos manteniendo la mayor cantidad de información posible. Esto es útil cuando se desea almacenar o transmitir datos de manera más eficiente.
4. Preparación de datos: el ACP se utiliza para preparar datos para su posterior análisis, como lo pueden ser modelos de *machine learning*, de regresión, pruebas de hipótesis o análisis de clúster (ver [capítulo 27](#)).
5. Análisis exploratorio: en el ACP no hay valores de p . De hecho, es una técnica exploratoria. El ACP se utiliza para explorar la estructura de los datos y encontrar patrones que no se pueden ver en una primera aproximación. Esto es útil cuando se desea comprender la relación entre las variables y cómo se relacionan entre sí. Además, permita la detección de datos atípicos y visualizar los datos en un espacio de menor dimensión.

Conceptos básicos del análisis de componentes principales

En esta sección, describiremos los conceptos teóricos básicos del ACP y, posteriormente, una vez que todos hablemos el mismo lenguaje, realizaremos un ejemplo práctico.

- ♦ Covarianza: este concepto se revisó superficialmente en el [capítulo 21](#). Aquí profundizaremos un poco más en él. La covarianza es una medida de la relación entre dos variables. Indica cómo cambian juntas dos variables (si lo hacen o no al mismo tiempo). Si la covarianza es positiva, significa que las dos variables cambian en la misma dirección. Si la covarianza es negativa, significa que las dos variables cambian en direcciones opuestas. Si la covarianza es 0, significa que las dos variables no están relacionadas.

- ♦ Matriz de covarianza: la matriz de covarianza es una matriz cuadrada que contiene las covarianzas entre todas las variables en un conjunto de datos. En el ACP se utiliza para identificar las direcciones de mayor variabilidad. La matriz de covarianza es una herramienta fundamental en el ACP, ya que se utiliza para calcular los componentes principales.
- ♦ Componentes principales: los componentes principales son nuevas variables no correlacionadas que se obtienen a partir de una combinación lineal de las variables originales.
- ♦ Multicolinealidad: este fenómeno, que es perjudicial para la regresión, no lo es para el ACP. La multicolinealidad indica qué tan correlacionadas están las variables.
- ♦ *Eigenvalues* y *eigenvectors*: los *eigenvalues* o autovalores son los valores propios de la matriz de covarianza, y los *eigenvectors* o autovectores son los vectores propios de la matriz de covarianza. Los *eigenvalues* indican cuánta varianza explica cada componente principal, y los *eigenvectors* indican la dirección de cada componente principal. Estos dos conceptos pertenecen al álgebra lineal y se utilizan para representar matrices cuadradas. El fundamento matemático de estos valores se escapa de los fines de este libro. Sin embargo, vamos a tratar de ejemplificarlo utilizando un ejemplo. Imaginemos que tenemos una caja cuya estructura es similar a una matriz. Esta caja, al ser sometida a una fuerza, se deforma. Por ejemplo, la podemos aplastar. Los *eigenvalues* serían la magnitud de la fuerza que se le aplica a la caja, y los *eigenvectors* serían la dirección en la que se deforma la caja. En otras palabras, los *eigenvalues* indican en qué medida la caja se aplasta o se mantiene estable en esas direcciones. En términos de ACP, estos *eigenvalues* representan la cantidad de varianza que cada componente principal captura. Cuando usted realiza un ACP, está buscando las direcciones (*eigenvectors*) a lo largo de las cuales los datos tienen la mayor variación (los ejes principales de la caja). Los *eigenvalues* nos dirían cuánto de esa variación existe en cada una de esas direcciones. Si la caja se aplasta mucho en una dirección, hay poca varianza (bajo eigenvalue), y si no se aplasta, hay mucha varianza (alto eigenvalue).

Implementación de un análisis de componentes principales en R

Para realizar un ejemplo para la implementación de un ACP en R vamos a utilizar el *Dataframe Breast Cancer Wisconsin (Diagnostic)*, la cual contiene datos a partir de una imagen digitalizada de una aspiración con aguja fina de masas mamarias. Todas estas mediciones reflejan características del tumor. La explicación puntal de cada uno de ellas se puede encontrar en [este enlace](#). Este arreglo de datos es una base de datos pública contenida en el repositorio [UC Irvine Machine Learning Repository](#).

En este libro utilizaremos una versión simplificada de esta base que contiene solo promedios de las mediciones digitales.

Las variables contenidas en este conjunto de datos son:

- ◇ ID
- ◇ Diagnosis
- ◇ radius_mean
- ◇ perimeter_mean
- ◇ area_mean
- ◇ smoothness_mean
- ◇ compactness_mean
- ◇ concavity_mean
- ◇ concave_points_mean
- ◇ symmetry_mean
- ◇ fractal_dimension_mean

Puede descargar a la versión simplificada de la base en [este enlace](#).

En los siguientes párrafos vamos explicar de forma sencilla la realización de un ACP en R. Tome en cuenta que explicación no incluye todas gráficas y análisis que se pueden realizar con un ACP, pero sí los más importantes. Además, un ACP siempre debe de ser acompañado de un análisis más profundo y de experiencia. Un investigador sin experiencia no podrá obtener la misma información que un investigador con experiencia.

Vamos con la explicación:

1. Importación de la base de datos

Importe su base de datos en R. Si después de descargar la base la guardó en su carpeta de Bases, el siguiente código debería de funcionar:

```
df ← read.csv("Bases/BasePCA.csv")
```

Si utiliza la función `head(df)` podrá visualizar las primeras filas de la base de datos. Con ello podrá percatarse de que hay 3 columnas que no son necesarias para el análisis. La primera columna es X, la cual es una variable de indexación con números consecutivos. La segunda es ID que contiene un número de identificación del paciente. La tercera columna es el diagnóstico. Uno de los requisitos para realizar un ACP es que las variables sean numéricas, por lo que la variable X, ID y Diagnosis deberán excluirse del análisis. Podemos sobrescribir la base y eliminar las variables con el siguiente código:

```
df ← df[1:100, -c(1, 2, 3)]
```

Por cuestiones didácticas y de visualización solo vamos a utilizar las primeras 100 filas de la base de datos. Se dejó intencionalmente disponible para descargar la base completa para que pueda repetir el proceso con la base completa.

A partir de ahora nuestra base `df` contiene solo variables numéricas y 100 casos.

2. Normalización

La normalización de datos es un paso importante en el ACP. Este proceso implica escalar los datos para que tengan una media de 0 y una desviación estándar de 1. Esto es importante porque las variables en un conjunto de datos pueden tener diferentes escalas y unidades (por ejemplo, `radius_mean`, `area_mean` y `smoothness_mean`), lo que puede afectar el resultado del ACP. La normalización de datos asegura que todas las variables tengan la misma escala y unidades, lo que facilita la comparación entre ellas. Además de que facilita la detección de valores atípicos.

Para realizar la normalización vamos a utilizar la función `scale()` de R:

```
df_norm ← scale(df) # Nos dará como resultado una base con variables normalizadas
```

3. Librerías necesarias

Aunque el ACP se puede realizar con las funciones base de R, en este libro vamos a utilizar la librería `factoextra` que contiene funciones más amigables para realizar un ACP. Para instalar la librería puede utilizar el siguiente código:

```
install.packages("factoextra")
```

Una vez instalada la librería, vamos a cargarla en nuestro entorno de trabajo:

```
library(factoextra)
```

Vamos a utilizar la función `prcomp()` para realizar el ACP. Esta función calcula los componentes principales (CP) de un conjunto de datos. La función `prcomp()` se alimenta con la base de datos normalizada y devuelve un objeto que contiene los componentes principales:

```
pca ← prcomp(df_norm)
```

Tome en cuenta que dentro la función `prcomp()` se puede normalizar la base de datos con el código: `prcomp(df, center = TRUE, scale. = TRUE)`, sin embargo, en este caso ya hemos normalizado la base de datos.

4. Resumen de nuestro ACP

En el ACP se busca que los CP expliquen la mayor cantidad de varianza en los datos. Entre más alta sea la varianza explicada por los CP, mejor será el ACP. Para obtener la varianza explicada por cada CP, podemos utilizar la función `summary()`:

```
summary(pca)
```

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	2.4980	1.6867	0.91998	0.64419	0.60087	0.39008	0.26791
Proportion of Variance	0.5673	0.2586	0.07694	0.03773	0.03282	0.01383	0.00653
Cumulative Proportion	0.5673	0.8259	0.90282	0.94055	0.97337	0.98720	0.99373

	PC8	PC9	PC10	PC11
Standard deviation	0.24831	0.08212	0.02426	6.982e-17
Proportion of Variance	0.00561	0.00061	0.00005	0.000e+00
Cumulative Proportion	0.99933	0.99995	1.00000	1.000e+00

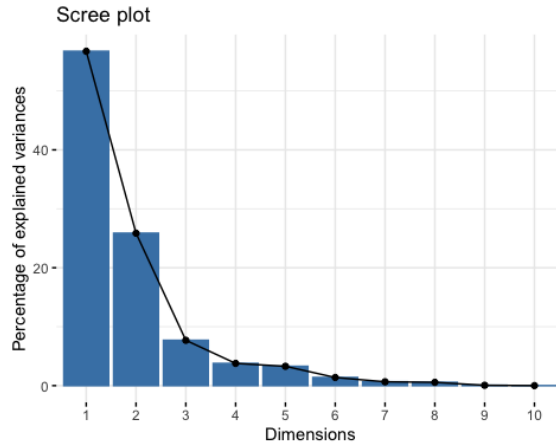
En el resultado anterior debemos de destacar algunos puntos importantes:

- ◇ En un ACP tendremos tantos CP como variables en la base de datos. En este caso, tenemos 10 variables y por tanto 10 CP. No tiene sentido quedarnos con los 10, ya que nuestro objetivo es reducir la dimensionalidad de los datos. Por lo tanto, debemos seleccionar solo los componentes principales que expliquen la mayor cantidad de varianza en los datos.
- ◇ El PC1 (componente principal 1) siempre será el que más explique la variación en los datos. El PC2 (componente principal 2) será el segundo componente que más explique la variación en los datos, y así sucesivamente.
- ◇ El valor de lo cada uno de los CP se encuentra en la columna Standard deviation y corresponde a la raíz cuadrada de los *eigenvalues*. La varianza explicada por cada CP se encuentra en la columna Proportion of Variance, es decir, cuánto contribuye cada CP a la variabilidad total de los datos.

5. Gráfico de la varianza explicada por cada CP

Para visualizar la varianza explicada por cada CP, podemos utilizar la función `fviz_eig()` de la librería `factoextra`:

```
fviz_eig(pca)
```


FIGURA 25.1. GRÁFICO DE LA VARIANZA EXPLICADA POR CADA COMPONENTE PRINCIPAL

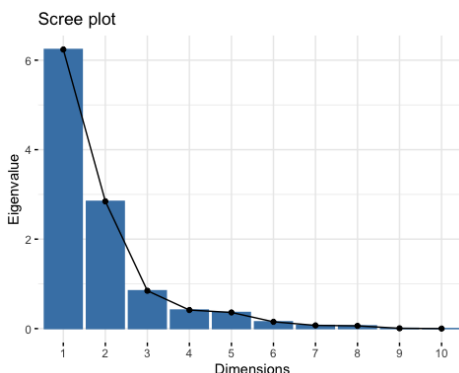
La [figura 25.1](#) muestra un *Scree plot*, el cual muestra la varianza explicada por cada componente principal. Esta gráfica muestra los mismos resultados realizados en el punto anterior. Sin embargo, graficar la varianza explicada por cada CP se suele utilizar para seleccionar los CP que son más representativos. Existe una técnica visual empírica, conocida como *la regla del codo*, que se utiliza para seleccionar los CP que son más representativos. Podemos imaginar el *Scree plot* como un brazo y visualizar como doble “codo” en este caso el componente 3. Por lo tanto, para estos datos los CP más importantes son los 3 primeros. Este recurso visual no es exento a errores y siempre se requiere de un análisis más profundo y de experiencia.

6. Gráfico de los eigenvalues

Otra forma que tenemos de seleccionar los CP más importantes es mediante un gráfico de los *eigenvalues*. Para realizar este gráfico podemos utilizar la función `fviz_eig()` de la librería `factoextra`:

```
fviz_eig(pca, choice = "eigenvalue")
```

FIGURA 25.2. GRÁFICO DE LOS EIGENVALUES



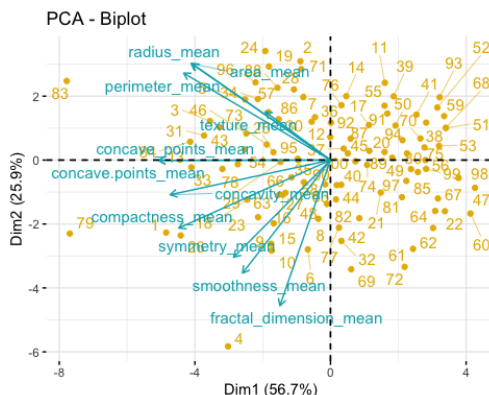
Se suele seleccionar aquellos CP con un *eigenvalue* mayor a 1. En este caso, según lo que se muestra en la [figura 25.2](#), se podrían seleccionar los primeros dos componentes.

7. Biplots

Los biplots son gráficos que muestran las observaciones y las variables en el mismo espacio. Los biplots son útiles para visualizar la relación entre las observaciones y las variables en un ACP. Para realizar un biplot en R podemos utilizar la función `fviz_pca_biplot()` de la librería `factoextra`:

```
fviz_pca_biplot(pca, repel = T, col.var = "#00AFBB", col.ind = "#E7B800")
```

FIGURA 25.3. BIPLLOT DE LOS DOS PRIMEROS COMPONENTES PRINCIPALES



Vamos a profundizar en la explicación de la [figura 25.3](#):

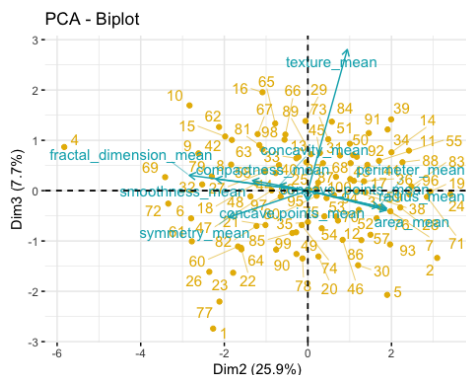
- ♦ Los ejes representan el PC1 y el PC2 respectivamente, en este caso Dim1 explica 56.7% y Dim2 25.9%. Juntos explican 82.6% de la varianza en los datos. Esto nos los conocíamos al hacer `summary(pca)`.
- ♦ Los puntos amarillos representan las observaciones (en este caso las filas de la base de datos) y los vectores azules representan las variables (en este caso las columnas de la base de datos).
- ♦ Las observaciones cercanas entre sí comparten características similares, mientras que las que están más alejadas presentan diferencias más pronunciadas en sus características.
- ♦ Las flechas azules representan a las variables originales, en este caso representados como vectores.
- ♦ La dirección de las flechas indica cómo cada variable contribuye a los componentes principales. Por ejemplo, las variables `radius_mean`, `area_mean` y `perimeter_mean` tienen una fuerte influencia en la dirección positiva de Dim1.
- ♦ Las flechas más largas, como las de `radius_mean` y `area_mean`, indican que estas variables tienen una gran influencia en la varianza explicada por el componente correspondiente.
- ♦ La dirección de los vectores indica la relación entre las variables y los componentes principales. Por ejemplo, las variables que están cerca entre sí en el gráfico tienen una relación positiva, mientras que las variables que están en direcciones opuestas tienen una relación negativa.
- ♦ Las variables que están cerca del origen del gráfico (por ejemplo `concave.points_mean`) tienen una baja correlación con el CP1.

Recordemos que de acuerdo con nuestro Scree plot los CP más importantes son los 3 primeros, aunque solamente 2 se encontraban por arriba de la unidad. Seleccionar 2 o 3 componentes dependerá del uso que se le quiera dar a los datos.

Si quisiéramos graficar los componentes 2 y 3, podemos hacerlo con el siguiente código:

```
fviz_pca_biplot(pca, axes = c(2, 3), repel = TRUE, col.var = "#00AFBB",
col.ind = "#E7B800")
```

FIGURA 25.4. BIPLLOT DE LOS COMPONENTES 2 Y 3



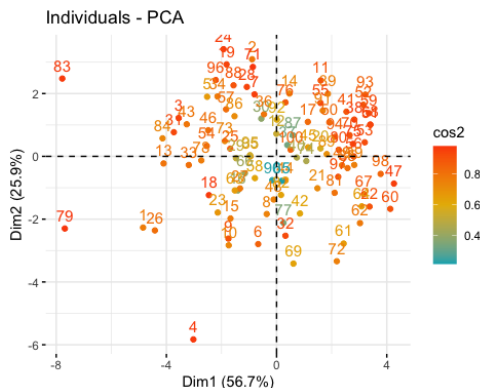
La interpretación de la [figura 25.4](#) es similar al anterior, solo que ahora estamos viendo la relación entre el PC2 y el PC3.

8. Patrones de datos

Uno de los objetivos de realizar un ACP es identificar patrones en los datos. Para visualizar estos patrones, podemos utilizar la función `fviz_pca_ind()` de la librería `factoextra`. Con este código podemos visualizar las observaciones en el espacio de los componentes principales:

```
fviz_pca_ind(pca,col.ind = "cos2", gradient.cols = c("#00AFBB",
"#E7B800", "#FC4E07"), repel = F)
```

FIGURA 25.5. PATRONES DE LOS DATOS EN EL ESPACIO DE LOS COMPONENTES PRINCIPALES



Note cómo en la [figura 25.5](#) utilizamos el argumento `col.ind = "cos2"`, este argumento nos permite colorear los puntos de acuerdo a su contribución a los componentes principales. Un valor alto de `cos2` (colores rojos) indica que el individuo está bien representado en este plano. En cambio, valores bajos de `cos2` (colores verdes) indican que el individuo no está tan bien representado en los dos primeros componentes y probablemente necesite de otros componentes para ser mejor descrito. Por lo tanto, con este gráfico podríamos identificar los individuos especialmente donde se encuentran los datos y de si se forman patrones entre ellos. También utilizamos el argumento `repel = F` para que los puntos no se repelan entre sí y poder identificar adecuadamente los patrones.

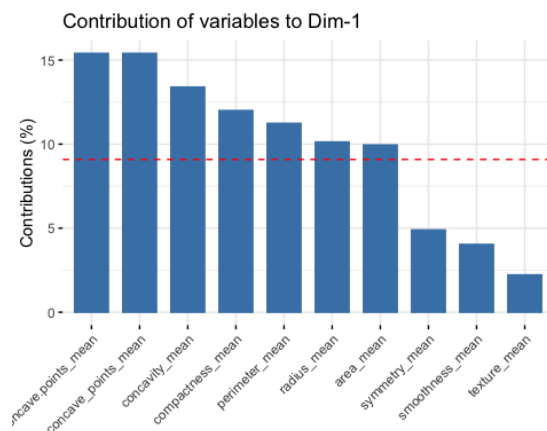
Los puntos que se encuentran junto a otros puntos indican que comparten características similares. Por otro lado, los puntos más alejados, como los números 83, 79 y 4, podrían ser *outliers* o individuos que presentan características únicas o muy diferentes en relación con el resto de los datos

9. Contribución de las variables

Para visualizar la contribución de las variables a los componentes principales, podemos utilizar la función `fviz_contrib()` de la librería `factoextra`:

```
fviz_contrib(pca, choice = "var", axes = 1, top = 10) #axes = 1 indica
que se graficará el primer componente principal
```

FIGURA 25.6. CONTRIBUCIÓN DE LAS VARIABLES AL PRIMER COMPONENTE PRINCIPAL

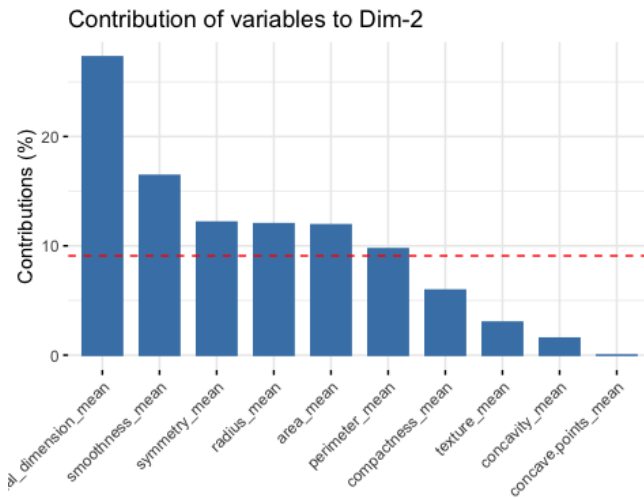


La [figura 25.6](#) nos dice que, para el primer componente principal (Dim-1), las variables relacionadas con concavidad y compactación (como `concave.points_mean` y `concavity_mean`) son las que más explican la varianza en los datos. Por otro lado, variables como `texture_mean` tienen una influencia menor en esta dimensión

La [figura 25.7](#) muestra la contribución de las variables al segundo componente principal (Dim-2):

```
fviz_contrib(pca, choice = "var", axes = 2, top = 10) # axes = 2 indica que se graficará el primer componente principal
```

FIGURA 25.7. CONTRIBUCIÓN DE LAS VARIABLES AL SEGUNDO COMPONENTE PRINCIPAL



10. Sigüientes pasos

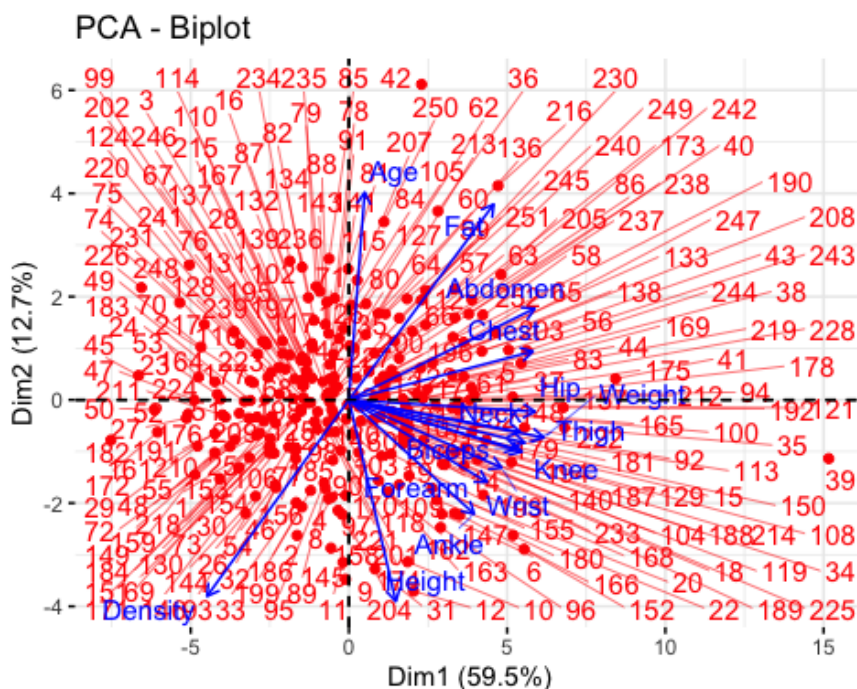
Una vez que hemos realizado un ACP, podemos utilizar los componentes principales como variables en análisis posteriores, como regresión, clasificación, clustering, etc. Recordemos que los CP son variables no correlacionadas que capturan la mayor cantidad de varianza en los datos. Por lo tanto, son útiles para reducir la dimensionalidad de los datos y encontrar patrones ocultos en ellos.

Ejercicio

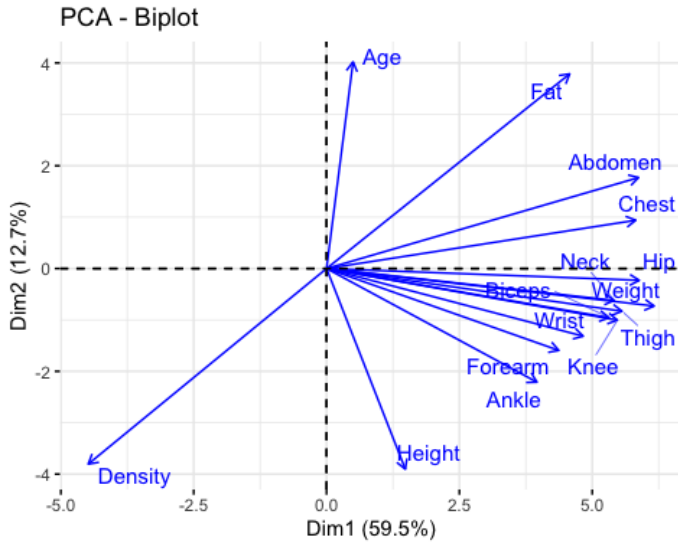
- ♦ **Ejercicio 25.1:** en el [capítulo 21](#) se utilizó la base de datos “bodyfat” para realizar un análisis de correlación. Realice un ACP con la base de datos “bodyfat” y realice un biplot con los dos primeros componentes principales.

Resolución ejercicio 25.1

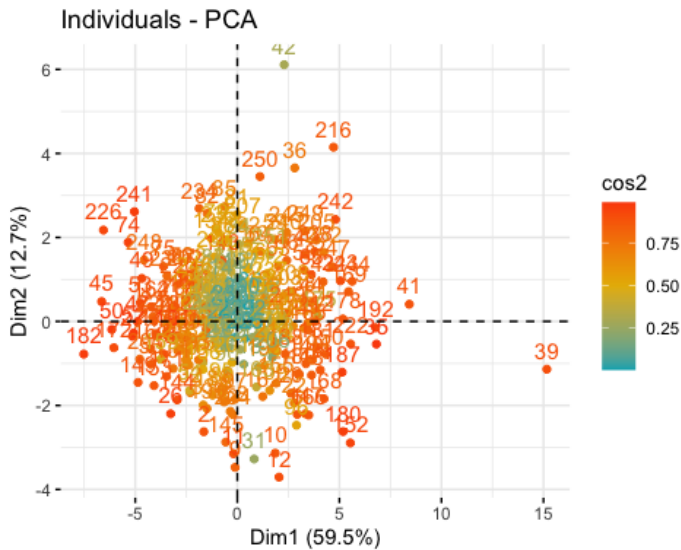
```
bodyfat <- read.csv("Bases/bodyfat.txt", sep = "")
prcomp_bodyfat <- prcomp(bodyfat, center = TRUE, scale. = TRUE)
fviz_pca_biplot(prcomp_bodyfat, axes = c(1, 2),
                 repel = TRUE, col.var = "blue", col.ind = "red")
```



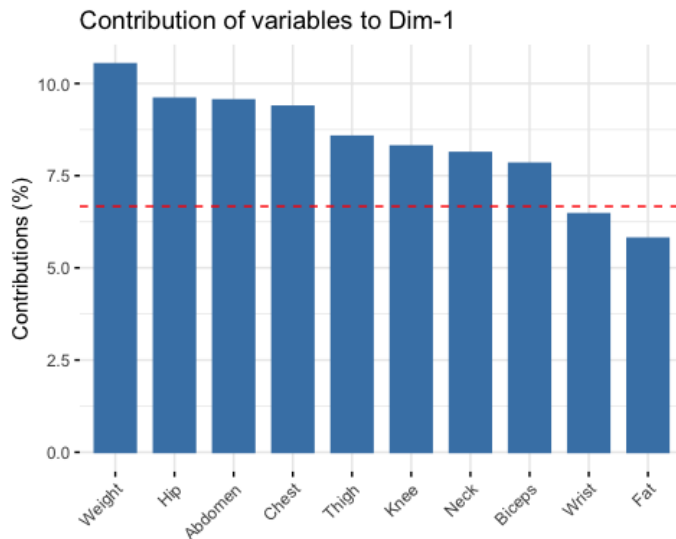
```
fviz_pca_biplot(prcomp_bodyfat, axes = c(1, 2), repel = TRUE,
               col.var = "blue", col.ind = "red", geom = "var")
```



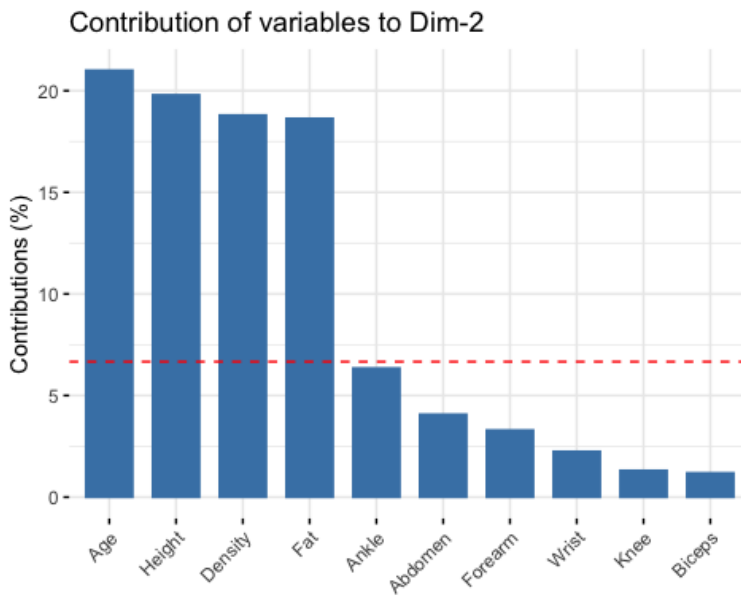
```
fviz_pca_ind(prcomp_bodyfat,col.ind = "cos2",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = F)
```




```
fviz_contrib(prcomp_bodyfat, choice = "var", axes = 1, top = 10)
```



```
fviz_contrib(prcomp_bodyfat, choice = "var", axes = 2, top = 10)
```



Referencias

- Chan, Y. H. (2004). Biostatistics 302. Principal component and factor analysis. Singapore medical journal, 45(12), 558-566.
- Giuliani, A. (2017). The application of principal component analysis to drug discovery and biomedical data. Drug discovery today, 22(7), 1069-1076. <https://doi.org/10.1016/j.drudis.2017.01.005>
- Groth, D., Hartmann, S., Klie, S. y Selbig, J. (2013). Principal components analysis. Methods in molecular biology, 930, 527-547. https://doi.org/10.1007/978-1-62703-059-5_22
- Hatfield, M. P. y Lovas, S. (2014). Conformational sampling techniques. Current pharmaceutical design, 20(20), 3303-3313. <https://doi.org/10.2174/13816128113199990603>
- Jolliffe, I. T. y Morgan, B. J. (1992). Principal component analysis and exploratory factor analysis. Statistical methods in medical research, 1(1), 69-95. <https://doi.org/10.1177/096228029200100105>
- Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer Science+Business Media. <https://doi.org/10.1007/b98835>
- Jolliffe, I. T. y Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 374(2065), 20150202. <https://doi.org/10.1098/rsta.2015.0202>
- Kassambara, A. y Mundt, F. (2020). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7.
- Sainani, K. L. (2014). Introduction to principal components analysis. PM & R: the journal of injury, function, and rehabilitation, 6(3), 275-278. <https://doi.org/10.1016/j.pmrj.2014.02.001>
- Vidal, R., Ma, Y. y Sastry, S. S. (2016). Principal Component Analysis. Generalized Principal Component Analysis. Interdisciplinary Applied Mathematics, 40. https://doi.org/10.1007/978-0-387-87811-9_2
- Wolberg, W., Mangasarian, O., Street, N. y Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. <https://doi.org/10.24432/C5DW2B>.
- Zhang, Z. (2016a). Introduction to machine learning: k-nearest neighbors. Annals of translational medicine, 4(11), 218. <https://doi.org/10.21037/atm.2016.05.01>
- Zhang, Z. (2016b). Introduction to machine learning: decision tree. Annals of translational medicine, 4(11), 237. <https://doi.org/10.21037/atm.2016.05.17>

Análisis factorial

Edsaúl Emilio Pérez Guerrero

Imagine que pertenece a un grupo de investigación interesado en evaluar la calidad de vida en un grupo de 200 pacientes. Para ello su equipo ha diseñado una encuesta que incluye 10 preguntas que miden la calidad de vida, como bienestar físico, bienestar emocional, bienestar social, bienestar económico, bienestar laboral, bienestar familiar, bienestar de pareja, bienestar sexual, bienestar espiritual y bienestar intelectual. Cada pregunta se mide en una escala de 1 a 5, donde 1 es la peor calidad de vida y 5 la mejor calidad de vida. La [tabla 26.1](#) y la [tabla 26.2](#) muestra los primero 6 datos de esta encuesta para las primeros 5 variables y las últimas 5 variables respectivamente.

TABLA 26.1. PRIMEROS 6 DATOS DE LAS PRIMERAS 5 VARIABLES DE LA ENCUESTA DE CALIDAD DE VIDA

	bienestar físico	bienestar emocional	bienestar social	bienestar económico
1	0.09801342	1.4617995	0.6015058	-0.03823524
2	-0.10027509	0.5242437	0.1362290	-0.31932404
3	0.61348422	0.1829938	-0.1415626	0.87894073
4	-0.42661693	0.7870909	0.2055095	-0.19617563
5	0.28883070	-0.1380941	-1.4818414	0.24078704
6	0.62280574	-0.2281063	0.2821633	1.01502448

```

bienestar laboral
1      -0.7707327
2      -0.4042276
3       1.2211643
4      -0.1303594
5       0.1686186
6       1.5470173
    
```

**TABLA 26.2. ÚLTIMOS 6 DATOS DE LAS ÚLTIMAS 5 VARIABLES
DE LA ENCUESTA DE CALIDAD DE VIDA**

```

bienestar familiar actividad pareja actividad sexual satisfacción espiritual
1      0.95489354      -0.4162317      -0.2487751      1.1042055
2      0.98446319      -0.6638312      -0.6064157      -0.8161672
3     -0.35957138      -0.2710965      -0.5974846      -0.4805509
4     -0.02533525       0.3654877       0.4703861       0.4268258
5     -1.06235981       0.7445146       0.4720719       0.9759910
6     -0.34284766      -0.5643742      -1.2364353      -1.1546142

satisfacción intelectual
1      -0.2783455
2      -0.8499992
3      -0.6112165
4      -0.2202697
5       0.6899408
6      -0.6970749
    
```

Esta encuesta tiene 10 variables que miden la calidad de vida de los pacientes. Sin embargo, es posible que algunas de estas variables estén correlacionadas entre sí. Por ejemplo, es posible que las variables de bienestar físico, bienestar emocional y bienestar social estén correlacionadas o depender uno de otro. Por ejemplo, una persona con buen bienestar físico también puede tener buen bienestar emocional y social. Por otro lado, es posible que las variables de bienestar económico, bienestar laboral y bienestar familiar estén correlacionadas entre sí, ya que una persona con buen bienestar económico

también puede tener buen bienestar laboral y familiar. Por lo tanto, es posible que estas 10 variables puedan ser resumidas en un número menor de variables que capturen la mayor cantidad de información posible. Una forma de hacer esto es mediante un análisis factorial.

Análisis factorial

El análisis factorial es una técnica estadística perteneciente al análisis multivariado que permite reducir la dimensionalidad de un conjunto de datos (como el de la encuesta). En otras palabras, el análisis factorial permite resumir un conjunto de variables en un número menor de variables no observadas llamadas factores (variables que no se pueden observar a simple vista). Estos factores son variables latentes que no pueden ser observadas directamente, pero que pueden ser inferidas a partir de las variables observadas. Los factores son contruidos de tal manera que capturan la mayor cantidad de información posible de las variables observadas.

Hasta este punto el análisis factorial parece muy similar al análisis de los componentes principales. Vamos a revisar las diferencias entre ambos métodos.

Diferencias entre análisis factorial y análisis de componentes principales

Tanto el análisis factorial y el análisis de componentes principales son técnicas de reducción de dimensionalidad que permiten resumir un conjunto de variables en un número menor de variables no observadas. Sin embargo, existen diferencias importantes entre ambos métodos; la principal de ellas es que en el análisis factorial se buscan factores latentes (variables no observables directamente) que expliquen la correlación entre las variables observadas, mientras que en el análisis de componentes principales se buscan explicar el comportamiento de las variables utilizando componentes lineales de las variables observables. En otras palabras, el análisis factorial busca factores latentes que expliquen la correlación entre las variables observadas, mientras que el análisis de componentes principales busca componentes lineales que expliquen la varianza de las variables observadas.

Existen otras diferencias entre ambos métodos, como la forma en que se construyen los factores y los componentes, la forma en que se interpretan los resultados y la forma en que se utilizan los resultados. Sin embargo, estas diferencias son más técnicas y no las describiremos a detalle en este libro, pero sí se mencionan en la [tabla 26.3](#).

TABLA 26.3. DIFERENCIAS ENTRE ANÁLISIS FACTORIAL Y ANÁLISIS DE COMPONENTES PRINCIPALES

CARACTERÍSTICA	ANÁLISIS FACTORIAL	ANÁLISIS DE COMPONENTES PRINCIPALES
Objetivo	Identificar factores latentes que explican las correlaciones entre las variables observadas.	Encontrar nuevas variables (componentes) que maximicen la varianza explicada de los datos originales.
Naturaleza de los factores/componentes	Factores latentes (no observables directamente).	Componentes lineales de las variables originales (observables).
Varianza explicada	Se enfoca en explicar la varianza común entre las variables.	Busca explicar la mayor cantidad de varianza total en los datos.
Supuestos	Asume la existencia de factores comunes y únicos.	No hace suposiciones sobre la estructura de los datos.
Interpretación	Los factores se interpretan en términos de las variables que tienen las cargas factoriales más altas.	Los componentes se interpretan en términos de su contribución a la varianza total.
Rotación de factores	Se utiliza para mejorar la interpretabilidad de los factores.	Generalmente no se utiliza, aunque es posible.

Tipos de análisis factorial

Existen dos tipos de análisis factorial: el análisis factorial exploratorio y el análisis factorial confirmatorio.

- ♦ **Análisis factorial exploratorio:** este tipo de análisis factorial busca identificar factores latentes que expliquen la correlación entre las variables observadas. Este enfoque se utiliza cuando no se tienen una idea clara o hipótesis de cómo se podrían agrupar las variables. Su propósito principal es explorar cómo se agrupan las variables de un conjunto de datos.

- ♦ **Análisis factorial confirmatorio:** este tipo análisis se realiza de forma subsecuente al análisis factorial exploratorio. Este busca confirmar los factores latentes propuestos en el análisis exploratorio. En ciertas ocasiones se puede realizar un análisis factorial confirmatorio sin haber realizado un análisis factorial exploratorio. En este último caso, se parte de una hipótesis previa o de un modelo teórico previamente propuesto.

En este libro nos enfocaremos en el análisis factorial exploratorio, ya que es el tipo de análisis factorial más común y utilizado en la práctica.

En la [tabla 26.4](#) se muestra un resumen de las diferencias entre el análisis factorial exploratorio y el análisis factorial confirmatorio.

TABLA 26.4. DIFERENCIAS ENTRE ANÁLISIS FACTORIAL EXPLORATORIO Y ANÁLISIS FACTORIAL CONFIRMATORIO

CARACTERÍSTICA	ANÁLISIS FACTORIAL EXPLORATORIO (AFE)	ANÁLISIS FACTORIAL CONFIRMATORIO (AFC)
Propósito	Descubrir factores subyacentes	Confirmar un modelo teórico previo
Modelo teórico	No requiere un modelo teórico	Requiere un modelo teórico definido
Hipótesis	Exploratorio, sin hipótesis clara	Confirmatorio, con hipótesis clara
Fase de investigación	Etapas inicial para desarrollar teorías o identificar patrones	Etapas posterior, cuando ya se ha desarrollado una teoría o modelo que se quiere probar

Aplicaciones del análisis factorial

Algunas de las aplicaciones del análisis factorial se enlistan a continuación. Puede dar clic en cada una de las aplicaciones y ver un artículo relacionado:

1. [Desarrollo y validación de cuestionarios y escalas de medición.](#) El análisis factorial es ampliamente utilizado en el desarrollo de pruebas psicológicas y escalas de personalidad.
2. [Reducción de dimensionalidad en conjuntos de datos.](#) En diversos estudios donde se tienen muchos datos, el análisis factorial puede ser útil para reducir la dimensionalidad de los datos y encontrar patrones ocultos.

3. **Identificación de factores latentes.** Este aspecto es muy utilizado en psicología y en salud pública en encuestas y cuestionarios.
4. **Modelos de predicción.** En la búsqueda de biomarcadores para diagnosticar enfermedades, el análisis factorial puede ayudar a reducir la complejidad de los datos biológicos y agrupar diferentes mediciones que están relacionadas con un mismo fenómeno subyacente, como la progresión de una enfermedad. También muchos modelos predictivos de *machine learning* utilizan antes una aproximación como el análisis factorial para reducir la dimensionalidad de los datos.
5. **Medicina personalizada.** En estudios de medicina personalizada, el análisis factorial se utiliza para identificar factores subyacentes en los datos clínicos, ya que recordemos que en la medicina personalizada se utilizan una gran cantidad de datos clínicos y genómicos que necesitan ser tratados antes de ser utilizados en la práctica clínica.
6. **Análisis de datos genómicos.** En estudios de genómica, el análisis factorial se utiliza para reducir la dimensionalidad de los datos y encontrar patrones ocultos en los datos genómicos.

Fundamentos matemáticos del análisis factorial

El objetivo principal del análisis factorial es descomponer una matriz de correlación o matriz de covarianza para identificar factores subyacentes que expliquen la correlación entre las variables observadas. El concepto de matriz de correlación fue descrito en “[Conceptos básicos del análisis de componentes principales](#)”.

En el análisis factorial, la matriz de correlación se descompone en: una matriz de cargas factoriales y una matriz de factores. La matriz de cargas factoriales contiene las correlaciones entre las variables observadas y los factores latentes, mientras que la matriz de factores contiene las correlaciones entre los factores latentes y las variables observadas. La matriz de factores es una matriz ortogonal, lo que significa que los factores son independientes entre sí, es decir, los factores no están correlacionados entre sí.

El modelo factorial postula que cada variable observada X_i es una combinación lineal de factores latentes $F_1, F_2, \dots, F_m$ más un término de error. Por lo tanto, el modelo factorial se puede expresar como:

$$X_i = \lambda_{i1}F_1 + \lambda_{i2}F_2 + \dots + \lambda_{im}F_m + \epsilon_i$$

Donde:

- ♦ X_i es la **variable observada**.
- ♦ $F_1, F_2, \dots, F_m$ son los **factores latentes** (no observables).
- ♦ $\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im}$ son las **cargas factoriales** que representan el peso o la contribución de cada factor a la variable observada.
- ♦ ϵ_i es el **término de error** o varianza específica de la variable X_i que no está explicada por los factores.

Por su parte, las **cargas factoriales** son coeficientes que representan la relación lineal entre las variables observadas y los factores latentes. Mientras que la **matriz de cargas factoriales** Δ es una matriz de dimensión $p \times m$, donde p es el número de variables observadas y m es el número de factores. Cada elemento λ_{ij} de esta matriz indica qué tan fuertemente la variable X_i está asociada con el factor F_j .

El modelo factorial en términos matriciales puede escribirse como:

$$X = \Delta F + \epsilon$$

Donde:

- ♦ X es el vector de variables observadas ($p \times 1$).
- ♦ Δ es la matriz de cargas factoriales ($p \times m$).
- ♦ F es el vector de factores latentes ($m \times 1$).
- ♦ ϵ es el vector de errores ($p \times 1$).

El análisis factorial se puede realizar utilizando diferentes métodos, como el método de los mínimos cuadrados ordinarios (MCO), el método de máxima verosimilitud (MV), el método de los mínimos cuadrados no ponderados (MCNP), el método de los mínimos cuadrados ponderados (MCP), entre otros. En este libro nos enfocaremos en el método de los mínimos cuadrados ordinarios (MCO), ya que es el método más común y utilizado en la práctica.

Implementación del análisis factorial en R

♦ **Ejemplo 26.1:** para realizar un análisis factorial en R vamos a seguir los siguientes pasos:

1. Instalar y cargar las librerías necesarias.
2. Cargar los datos.
3. Evaluar la adecuación de los datos para el análisis factorial.
4. Realizar el análisis factorial exploratorio.
5. Decidir cuántos factores retener.
6. Aplicar rotación a los factores para mejorar la interpretabilidad.
7. Interpretar los factores.

Conforme usted adquiera experiencia podrá modificar esta lista de pasos. Por ejemplo, si ya tiene experiencia en análisis factorial, puede omitir la evaluación de la adecuación de los datos y pasar directamente al análisis factorial.

Instalar y cargar las librerías necesarias

En este libro utilizaremos las librerías `psych` (que ya hemos utilizado) y `GPArotation`:

```
# Instalar librerías necesarias
install.packages("psych") # Si no está instalada
install.packages("GPArotation")
Procedemos a cargar las librerías necesarias:
library(psych)
library(GPArotation)
```

Para este ejemplo vamos a crear una base de datos sobre la encuesta de calidad de vida de la que hablamos al inicio del capítulo. Le recomendamos copiar y pegar el código en su consola de R para que pueda seguir el ejemplo:

```
# Simular datos de una encuesta de calidad de vida en 200 pacientes
# Establecer la semilla para la reproducibilidad
set.seed(123)
```

```

# Definir el número de observaciones
n ← 200

# Generar 3 factores latentes (comunes) que influirán en las variables
factor1 ← rnorm(n, mean = 0, sd = 1) # Factor relacionado con bienestar físico
factor2 ← rnorm(n, mean = 0, sd = 1) # Factor relacionado con bienestar emocional
factor3 ← rnorm(n, mean = 0, sd = 1) # Factor relacionado con actividad diaria

# Generar variables observadas que están correlacionadas con los factores
bienestar_fisico ← 0.4 * factor1 + rnorm(n, mean = 0, sd = 0.3)
bienestar_emocional ← 0.6 * factor2 + rnorm(n, mean = 0, sd = 0.4)
bienestar_social ← 0.5 * factor2 + rnorm(n, mean = 0, sd = 0.5)
bienestar_economico ← 0.4 * factor1 + rnorm(n, mean = 0, sd = 0.3)
bienestar_laboral ← 0.6 * factor1 + rnorm(n, mean = 0, sd = 0.4)
bienestar_familiar ← 0.5 * factor2 + rnorm(n, mean = 0, sd = 0.5)
actividad_pareja ← 0.4 * factor3 + rnorm(n, mean = 0, sd = 0.3)
actividad_sexual ← 0.6 * factor3 + rnorm(n, mean = 0, sd = 0.4)
satisfaccion_espiritual ← 0.5 * factor3 + rnorm(n, mean = 0, sd = 0.5)
satisfaccion_intelectual ← 0.5 * factor3 + rnorm(n, mean = 0, sd = 0.5)

# Crear el dataframe con las variables
calidad_vida ← data.frame(
  bienestar_fisico,
  bienestar_emocional,
  bienestar_social,
  bienestar_economico,
  bienestar_laboral,
  bienestar_familiar,
  actividad_pareja,
  actividad_sexual,
  satisfaccion_espiritual,
  satisfaccion_intelectual
)

```

Este código fue creado intencionalmente para poder aplicar un análisis factorial. Por ello, podrá ver objetos como `factor1`, `factor2`, `factor3`, que son los factores latentes que influyen en las variables observadas.

Le recomendamos que antes de realizar un análisis factorial (o cualquier otro análisis) revise los datos para asegurarse que los datos sean adecuados. En el caso del análisis factorial, es importante que los datos cumplan con ciertos supuestos, como la normalidad y la homocedasticidad. Pero además es importante que las variables estén correlacionadas entre sí, ya que el análisis factorial busca identificar factores latentes que expliquen la correlación entre las variables. Para ello recurrimos a dos pruebas:

- ❖ La prueba de esfericidad de Bartlett. Función `cortest.bartlett()` de la librería `psych`.
- ❖ La prueba de Kaiser-Meyer-Olkin (KMO). Función `KMO()` de la librería `psych`.

En R podemos hacer estas pruebas con el siguiente código:

```
cortest.bartlett(calidad_vida)|>
  print()
R was not square, finding R from data
$chisq
[1] 712.5325

$p.value
[1] 6.145533e-121

$df
[1] 45
```

En el caso de la prueba de esfericidad de Bartlett, la hipótesis nula es que la matriz de correlación es una matriz de identidad, es decir, que las variables son independientes entre sí. Por lo tanto, si el p es menor a 0.05, rechazamos la hipótesis nula y concluimos que las variables están correlacionadas entre sí. En este caso nuestros datos son adecuados para un análisis factorial:

```
KMO(calidad_vida)|>
  print()
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = calidad_vida)
Overall MSA = 0.74
MSA for each item =
      bienestar_fisico      bienestar_emocional      bienestar_social
0.71      0.68      0.70
      bienestar_economico      bienestar_laboral      bienestar_familiar
0.72      0.70      0.67
      actividad_parejaactividad_sexualsatisfaccion_espiritual
0.78      0.76      0.85
satisfaccion_intelectual
                        0.84
```

En el caso de la prueba de KMO, el valor de debe ser mayor a 0.6 para que los datos sean adecuados para un análisis factorial. En este caso nuestros datos son adecuados para un análisis factorial.

En el caso de que los datos no cumplan con estos criterios, es posible que el análisis factorial no sea adecuado para los datos. Si es así, es recomendable realizar una matriz de correlación y revisar si alguna de las variables no presenta correlación o una correlación baja comparada con el resto.

Ahora procedemos a realizar el análisis factorial exploratorio. Para ello utilizamos la función `fa()` de la librería `psych`:

```
# Realizar el análisis factorial exploratorio
# Aquí extraemos inicialmente 3 factores
resultado_afe <- fa(calidad_vida, nfactors = 3, rotate = "none")

# Revisar el resultado del análisis factorial
print(resultado_afe)
Factor Analysis using method = minres
Call: fa(r = calidad_vida, nfactors = 3, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	MR1	MR2	MR3	h2	u2	com
bienestar_fisico	0.09	0.74	-0.24	0.61	0.39	1.3
bienestar_emocional	-0.05	0.24	0.71	0.57	0.43	1.2
bienestar_social	-0.04	0.33	0.63	0.51	0.49	1.5
bienestar_economico	0.10	0.75	-0.19	0.60	0.40	1.2
bienestar_laboral	0.12	0.75	-0.32	0.68	0.32	1.4
bienestar_familiar	-0.11	0.29	0.66	0.53	0.47	1.4
actividad_pareja	0.80	-0.06	0.13	0.66	0.34	1.1
actividad_sexual	0.84	-0.08	0.09	0.72	0.28	1.0
satisfaccion_espiritual	0.68	-0.07	-0.01	0.47	0.53	1.0
satisfaccion_intelectual	0.67	-0.0	0.06	0.46	0.54	1.0

	MR1	MR2	MR3
SS loadings	2.31	1.93	1.57
Proportion Var	0.23	0.19	0.16
Cumulative Var	0.23	0.42	0.58
Proportion Explained	0.40	0.33	0.27
Cumulative Proportion	0.40	0.73	1.00

Mean item complexity = 1.2

Test of the hypothesis that 3 factors are sufficient.

df null model = 45 with the objective function = 3.66 with Chi Square = 712.53
df of the model are 18 and the objective function was 0.06

The root mean square of the residuals (RMSR) is 0.02

The df corrected root mean square of the residuals is 0.02

The harmonic n.obs is 200 with the empirical chi square 4.14 with prob < 1

The total n.obs was 200 with Likelihood Chi Square = 10.97 with prob < 0.9

Tucker Lewis Index of factoring reliability = 1.027

RMSEA index = 0 and the 90 % confidence intervals are 0 0.029

BIC = -84.4

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	MR1	MR2	MR3
Correlation of (regression) scores with factors	0.93	0.91	0.88
Multiple R square of scores with factors	0.86	0.84	0.78
Minimum correlation of possible factor scores	0.72	0.67	0.56

Una interpretación puntal de los resultados anteriores es la siguiente:

- ♦ **MR1, MR2, MR3:** estas columnas representan las cargas factoriales para cada variable en los tres factores extraídos. Las cargas factoriales indican qué tan fuertemente está asociada cada variable con el respectivo factor. Un valor alto (positivo o negativo) en una de estas columnas significa que la variable está fuertemente asociada con ese factor.
- ♦ **h2 (comunalidad):** esta columna muestra la comunalidad de cada variable, que representa la proporción de la varianza de esa variable explicada por los factores. Valores cercanos a 1 indican que la mayor parte de la varianza de la variable es explicada por los factores, mientras que valores bajos indican lo contrario
- ♦ **u2 (varianza específica):** esta columna muestra la varianza específica de cada variable, que es la proporción de la varianza de la variable que no está explicada por los factores. Se calcula como $1 - h2$. Un valor alto de u2 indica que el factor explica menos de la varianza de la variable.
- ♦ **com (complejidad):** mide cuántos factores explican una variable. Valores más cercanos a 1 indican que una variable carga principalmente en un solo factor, mientras que valores mayores indican que la variable tiene cargas significativas en múltiples factores. Un valor alto sugiere que la variable está relacionada con varios factores.

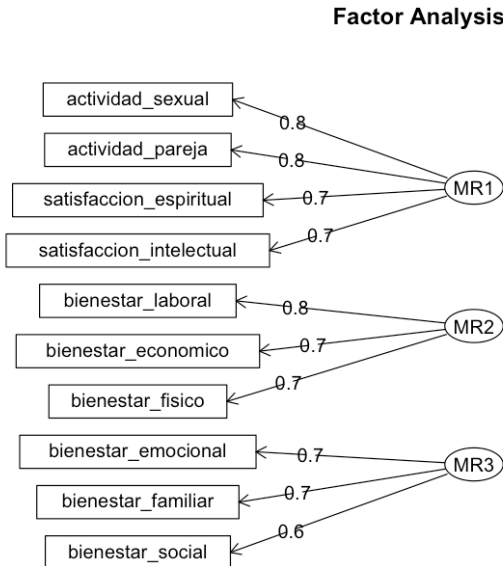
Ahora bien, de estos resultados también podemos ver que:

- ♦ La variable `bienestar_fisico` tiene una carga fuerte en el Factor 2 (MR2) con un valor de 0.74. Esto indica que este factor explica la mayor parte de la varianza en esta variable. La comunalidad de 0.61 sugiere que alrededor del 61% de la varianza en esta variable es explicada por los factores.

- ♦ La variable `bienestar_emocional` tiene una carga más fuerte en el Factor 3 (MR3) con 0.71, lo que indica que esta variable está principalmente asociada con ese factor. Su comunalidad es 0.57, lo que significa que aproximadamente el 57% de su varianza es explicada por los factores.
- ♦ Por otro lado, las variables `actividad_pareja` y `actividad_sexual` tienen cargas muy altas en el Factor 1 (MR1) con valores de 0.80 y 0.84, respectivamente. Esto sugiere que estas dos variables están altamente relacionadas con el primer factor. Sus comunalidades (0.66 y 0.72) también son bastante altas, lo que indica que los factores explican bien la varianza de estas variables.

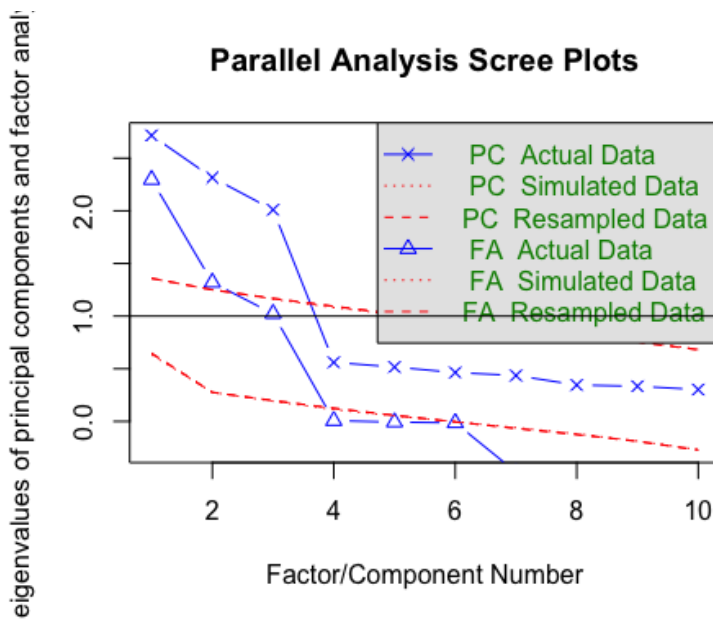
Una mejor manera de visualizar los resultados del análisis factorial es con la función `fa.diagram()` de la librería `psych`. Esta función nos permite visualizar los factores y las variables que están asociadas a cada factor:

```
# Visualizar los factores
fa.diagram(resultado_afe)
```



En este caso seleccionamos arbitrariamente 3 factores. Sin embargo, es posible que necesitemos seleccionar un número diferente de factores. Para ello podemos utilizar diferentes criterios, como el criterio de *scree* y el criterio de Kaiser. Para evaluar estos criterios podemos utilizar la función `fa.parallel()` de la librería `psych`, que nos permite visualizar y evaluar en un gráfico ambos criterios:

```
# Graficar el scree plot para decidir cuántos factores retener
fa.parallel(calidad_vida, fa = "both", n.iter = 100)
```



Parallel analysis suggests that the number of factors = 3 and the number of components = 3

```
# Verificar los valores propios
print(resultado_afe$values)
[1] 2.309073121 1.928692760 1.567682327 0.079347176 0.058039741
[6] 0.022919171 -0.004370959 -0.029197342 -0.035101419 -0.091636594
```

Este gráfico muestra un análisis paralelo o *Parallel Analysis Scree Plot* que nos ayudará a seleccionar los factores con el criterio *scree* y el criterio de Kaiser. En el *Parallel Analysis Scree Plot* la línea azul con “X” muestra los *eigenvalues* de los datos reales, mientras que la línea azul con triángulo muestra los resultados de los *eigenvalues*, pero para el análisis factorial. Las líneas punteadas muestran los datos simulados. En este gráfico buscamos factores que estén por encima de la línea del 1 para el criterio de Kaiser y el doble en el *scree*. Para estos datos en específico la mejor selección son 3 factores.

Finalmente, podemos aplicar una rotación a los factores para mejorar la interpretabilidad de los resultados. En este caso utilizaremos la rotación Varimax, que es una de las rotaciones más comunes y utilizadas en la práctica. Para ello utilizamos la función `fa()` con el argumento `rotate = “varimax”`:

```
# Realizar el análisis factorial con rotación Varimax
resultado_rot_varimax <- fa(calidad_vida, nfactors = 3, rotate = "varimax")
print(resultado_rot_varimax)
```

Factor Analysis using method = minres
Call: fa(r = calidad_vida, nfactors = 3, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix

	MR1	MR2	MR3	h2	u2	com
bienestar_fisico	0.00	0.78	0.03	0.61	0.39	1
bienestar_emocional	0.01	-0.03	0.75	0.57	0.43	1
bienestar_social	0.00	0.08	0.71	0.51	0.49	1
bienestar_economico	0.02	0.77	0.09	0.60	0.40	1
bienestar_laboral	0.02	0.82	-0.03	0.68	0.32	1
bienestar_familiar	-0.06	0.03	0.73	0.53	0.47	1
actividad_pareja	0.81	-0.01	0.04	0.66	0.34	1
actividad_sexual	0.85	-0.01	0.00	0.72	0.28	1
satisfaccion_espiritual	0.68	0.01	-0.09	0.47	0.53	1
satisfaccion_intelectual	0.68	0.05	0.00	0.46	0.54	1

	MR1	MR2	MR3
SS loadings	2.30	1.89	1.62
Proportion Var	0.23	0.19	0.16

```

Cumulative Var      0.23 0.42 0.58
Proportion Explained 0.40 0.33 0.28
Cumulative Proportion 0.40 0.72 1.00

```

```
Mean item complexity = 1
```

```
Test of the hypothesis that 3 factors are sufficient.
```

```

df null model = 45 with the objective function = 3.66 with Chi Square = 712.53
df of the model are 18 and the objective function was 0.06

```

```
The root mean square of the residuals (RMSR) is 0.02
```

```
The df corrected root mean square of the residuals is 0.02
```

```
The harmonic n.obs is 200 with the empirical chi square 4.14 with prob < 1
```

```
The total n.obs was 200 with Likelihood Chi Square = 10.97 with prob < 0.9
```

```
Tucker Lewis Index of factoring reliability = 1.027
```

```
RMSEA index = 0 and the 90 % confidence intervals are 0 0.029
```

```
BIC = -84.4
```

```
Fit based upon off diagonal values = 1
```

```
Measures of factor score adequacy
```

	MR1	MR2	MR3
Correlation of (regression) scores with factors	0.93	0.92	0.88
Multiple R square of scores with factors	0.86	0.84	0.78
Minimum correlation of possible factor scores	0.72	0.68	0.56

Después de la rotación es más fácil interpretar los factores. Por ejemplo, el Factor 1 ahora está compuesto principalmente por las variables `actividad_pareja` y `actividad_sexual`, mientras que el Factor 2 está compuesto principalmente por las variables `bienestar_fisico`, `bienestar_economico` y `bienestar_laboral`. El Factor 3 está compuesto principalmente por las variables `bienestar_emocional`, `bienestar_social`, `bienestar_familiar`, `satisfaccion_espiritual` y `satisfaccion_intelectual`.

Pasos siguientes:

- ❖ Interpretar los factores y darles nombres.
- ❖ Realizar un análisis factorial confirmatorio para validar los resultados (que no se revisarán en este libro).
- ❖ Eliminar variables que no contribuyen a los factores.
- ❖ Utilizar los factores en análisis posteriores, como análisis de clúster, análisis de regresión, análisis de varianza, entre otros.
- ❖ Probar con otros métodos de rotación, como la rotación Oblimin, Pro-max, entre otros.

Ejercicios

- ❖ **Ejercicio 26.1:** utilice el *Dataframe* [Breast Cancer Wisconsin \(Diagnostic\)](#) que se utilizó en el [capítulo 25](#) y realice un análisis factorial exploratorio. Elimine las variables que no sean numéricas antes de realizar el análisis factorial.

Resolución ejercicio 26.1

```
df <- read.csv("Bases/BasePCA.csv")
df <- df[,c(-1,-2,-3)]
cortest.bartlett(df)|>
  print()
KMO(df)|>
  print()
resultado_afe <- fa(df, nfactors = 2, rotate = "none")
print(resultado_afe)
fa.diagram(resultado_afe)
fa.parallel(df, fa = "both", n.iter = 100)
print(resultado_afe$values)
resultado_rot_varimax <- fa(df, nfactors = 3, rotate = "varimax")
print(resultado_rot_varimax)
```

- ♦ **Ejercicio 26.2:** en el [capítulo 21](#) se utilizó la base de datos “bodyfat”. Con ella realice un análisis factorial exploratorio:

Resolución ejercicio [ejercicio 26.2](#)

```
df <- read.csv("Bases/bodyfat.txt", sep = " ")
cortest.bartlett(df)|>
  print()
KMO(df)|>
  print()
df <- df[,c(-1,-3)] # Quitar Age
cortest.bartlett(df)|>
  print()
KMO(df)|>
  print()
resultado_afe <- fa(df, nfactors = 2, rotate = "none")
print(resultado_afe)
fa.diagram(resultado_afe)
fa.parallel(df, fa = "both", n.iter = 100)
print(resultado_afe$values)
resultado_rot_varimax <- fa(df, nfactors = 3, rotate = "varimax")
print(resultado_rot_varimax)
```

Referencias

- Alavi, M., Visentin, D. C., Thapa, D. K., Hunt, G. E., Watson, R. y Cleary, M. (2020). Exploratory factor analysis and principal component analysis in clinical studies: Which one should you use? *Journal of advanced nursing*, 76(8), 1886-1889. <https://doi.org/10.1111/jan.14377>
- Brown, T. A. (2015). *Confirmatory factor analysis for applied research* (2nd ed.). Guilford Press.
- De Winter, J. C., Dodou, D. y Wieringa, P. A. (2009). Exploratory Factor Analysis With Small Sample Sizes. *Multivariate behavioral research*, 44(2), 147-181. <https://doi.org/10.1080/00273170902794206>

- Domínguez-Lara, S. A. (2016). Análisis factorial exploratorio y complejidad factorial: más allá de las rotaciones [Exploratory factorial analysis and factorial complexity: Beyond rotations]. *Enfermería clínica*, 26(6), 401. <https://doi.org/10.1016/j.enfcli.2016.06.001>
- Fayers, P. M. y Hand, D. J. (1997). Factor analysis, causal indicators and quality of life. *Quality of life research: an international journal of quality of life aspects of treatment, care and rehabilitation*, 6(2), 139-150. <https://doi.org/10.1023/a:1026490117121>
- Finch, W. H. (2020). *Exploratory factor analysis*. SAGE Publications, Inc. <https://doi.org/10.4135/9781544339900>
- Garson, G. D. (2023). *Factor analysis and dimension reduction in R: A social scientist's toolkit*. Routledge.
- Gorsuch, R. L. (1997). Exploratory factor analysis: its role in item analysis. *Journal of personality assessment*, 68(3), 532-560. https://doi.org/10.1207/s15327752jpa6803_5
- Guillot-Valdés, M., Guillén-Riquelme, A., Sierra, J. C. y Buela-Casal, G. (2022). Network and Exploratory Factorial Analysis of the Depression Clinical Evaluation Test. *International journal of environmental research and public health*, 19(17), 10788. <https://doi.org/10.3390/ijerph191710788>
- Izquierdo, I., Olea, J. y Abad, F. J. (2014). Exploratory factor analysis in validation studies: uses and recommendations. *Psicothema*, 26(3), 395-400. <https://doi.org/10.7334/psicothema2013.349>
- Nájera, P., Abad, F. J. y Sorrel, M. A. (2023). Is exploratory factor analysis always to be preferred? A systematic comparison of factor analytic techniques throughout the confirmatory-exploratory continuum. *Psychological methods*. <https://doi.org/10.1037/met0000579>
- Shrestha, N. (2021). Factor Analysis as a Tool for Survey Analysis. *American Journal of Applied Mathematics and Statistics*, 9(1), 4-11.
- Tavakol, M. y Wetzell, A. (2020). Factor Analysis: a means for theory and instrument development in support of construct validity. *Int J Med Educ*, 11, 245-247. doi: 10.5116/ijme.5f96.of4a. PMID: 33170146; PMCID: PMC7883798.
- Walter, S. D. y Belo, I. J. (2023). Design and analysis of factorial clinical trials: The impact of one treatment's effectiveness on the statistical power and required sample size of the other. *Statistical methods in medical research*, 32(6), 1124-1144. <https://doi.org/10.1177/09622802231163332>

Willmer, M., Westerberg Jacobson, J. y Lindberg, M. (2019). Exploratory and Confirmatory Factor Analysis of the 9-Item Utrecht Work Engagement Scale in a Multi-Occupational Female Sample: A Cross-Sectional Study. *Frontiers in psychology*, 10, 2771. <https://doi.org/10.3389/fpsyg.2019.02771>

Introducción al análisis de clústeres

Alberto Beltrán Ramírez

Desde el inicio de los tiempos, el ser humano se ha encargado de agrupar objetos similares para generar una clasificación, por ejemplo, para identificar características específicas, como que pudieran ser comestibles, venenosas, etc. La clasificación de objetos es fundamental en la vida diaria de los seres vivos, específicamente, en la del ser humano, ya que ha sido necesaria para el desarrollo del lenguaje, ya que consiste en palabras que nos permiten etiquetar cosas que tiene características en común mediante su descripción.

En muchos casos, la clasificación permite resumir los datos en grupos pequeños, proporcionando una descripción concisa de similitudes y diferencias. Por ejemplo, en la investigación de mercados, agrupar a los consumidores según sus preferencias ayuda a identificar sus preferencias por algunos productos. Además de la organización, la clasificación también puede tener propósitos más fundamentales, como en medicina, donde se utiliza para predecir y tratar enfermedades, así como para investigar sus causas.

El término *clúster* se ha utilizado de manera intuitiva sin una definición formal. Definirlo resulta complicado y no siempre es necesario. Algunos autores sugieren que un *clúster* se evalúa en función del juicio de valor del usuario, es decir, si el uso del término *clúster* aporta valor al investigador, es suficiente. Sin embargo, se suele intentar definir un *clúster* en términos de cohesión interna (homogeneidad) y aislamiento externo (separación).

El término *clustering* se refiere a un conjunto muy amplio de técnicas para encontrar subgrupos o *clústeres* en un conjunto de datos. Cuando las observaciones de un conjunto de datos son agrupadas, buscamos dividir las en grupos

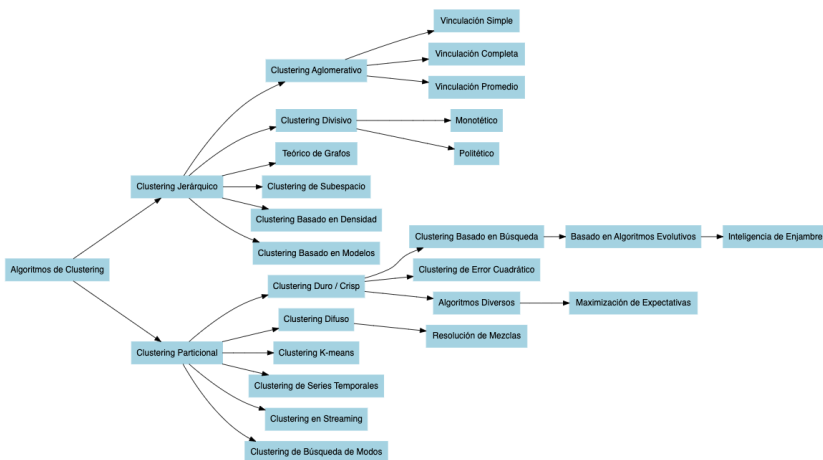
distintos, de modo que las observaciones dentro de cada grupo sean muy similares entre sí, mientras que las observaciones en diferentes grupos sean bastante diferentes. Para poder llevar a cabo esta agrupación debe definirse el significado de que dos o más observaciones sean similares o diferentes entre sí.

Por ejemplo, supongamos que tenemos un conjunto de número de observaciones, cada una con número de características. Las observaciones podrían corresponder a muestras de tejido de pacientes con cáncer de piel, y las características podrían corresponder a las mediciones realizadas a cada muestra de tejido; estas podrían ser mediciones clínicas, como el estadio clínico del paciente, o podrían ser mediciones de expresión génica. Podríamos tener razones para creer que las X muestras de tejido son heterogéneas; pudiera ser, por ejemplo, que quizás haya algunos subtipos desconocidos de cáncer de piel. El *clustering* podría usarse para encontrar estos subgrupos.

Clasificación

Los *clústeres*, al igual que muchos de los elementos que participan en las diversas ciencias en general, pueden ser clasificados. La [figura 27.1](#) muestra una clasificación de los *clústeres*, tome en cuenta que puede haber más tipos de *clústeres*.

FIGURA 27.1. CLASIFICACIÓN DE LOS CLÚSTERES



Los *clústeres* son herramientas más utilizadas en otros contextos, como la ciencia de datos y el *machine learning*, por lo que profundizar en ellos se escapa de los objetivos de este libro. Sin embargo, a continuación, se presentan algunos de los tipos de *clústeres* más comunes, sin la intención de profundizar en ellos, al menos no desde un punto de vista matemático.

Clústeres jerárquicos

En este tipo de *clústeres* los datos son particionados en niveles dentro de un formato jerárquico con un enfoque de arriba hacia abajo o de abajo hacia arriba, generando un dendograma con la estructura jerárquica de los *clústeres* formados, permitiendo explorar datos en diferentes niveles del mismo. Dentro del método algomerativo, los *clústeres* se generan a partir de objetos individuales que se fusionan iterativamente en *clústeres* más grandes que forman los distintos niveles de jerarquía hasta que se forma un único *clúster* o se cumple el criterio de parada. Por el contrario, en el método divisivo el *clúster* que contiene todos los objetos, como el nombre del *clúster* lo dice, se divide hasta que cada objeto forma un *clúster* individual o cumple el criterio de parada. Esta unión o división de datos se realiza en función de las similitudes o diferencias de los objetos contenidos en el *clúster*, determinadas por la generalización de la distancia entre puntos individuales a la distancia entre subconjuntos de puntos, utilizando la métrica de enlace, una medida de proximidad.

Clústeres divisivos

Divide cada *clúster* en partes más pequeñas comenzando con cada objeto en un solo *clúster*, hasta que se alcanza el número requerido de *clústeres* o el criterio de parada, es decir, los objetos de datos se consideran inicialmente como un *clúster* fusionado que se va dividiendo progresivamente. El método estándar para dividir un *clúster* en dos subconjuntos que contienen uno o más elementos requiere considerar cada posible bifurcación.

El *clustering* divisivo utiliza dos métodos principales: monotético o politético. Los *clústeres* divisivos monotéticos se adquieren utilizando una sola variable en cada división, partiendo objetos con valores específicos de aquellos objetos sin valor.

Un *clúster* divisivo es monotético, si la combinación de una característica lógica de cada uno que involucra una variable es esencial y suficiente para que pueda pertenecer al *clúster*. Cuanto mayor sea el número de objetos, mayor es la posible disminución del número de biparticiones mediante el método monotético.

Por otra parte, el enfoque de *clustering* divisivo politético utiliza todas las variables simultáneamente a través de valores de disimilitud o distancia, por el contrario del monotético que se basa en el orden de una sola variable.

Varios *clústeres* divisivos son politéticos, de modo que todas las variables se consideran simultáneamente en tareas para revelar las similitudes de las dos instancias.

Clústeres aglomerativos

En el método aglomerativo, los *clústeres* se originan a partir de objetos individuales, que se van fusionando iterativamente en *clústeres* más grandes que forman los diversos niveles de la jerarquía, hasta que un solo *clúster* contenga todo el conjunto de objetos o hasta que se cumple el criterio de paro. El último *clúster* en ser generado forma la raíz de la jerarquía. Para formación de los *clústeres*, se combinan los dos objetos más cercanos basado en la medida de similitud utilizada. Se requieren como máximo X iteraciones para completar la operación de *clustering*, ya que dos *clústeres* se fusionan en cada iteración para formar uno nuevo.

Dentro de este grupo, podemos encontrar los de tipo de enlace simple, enlace promedio y enlace completo.

- ♦ *Clustering* de enlace simple (método del vecino más cercano o *neighborhood-based clustering*): mide la distancia más cercana entre cualquier miembro de un *clúster* y cualquier otro miembro del *clúster*. Mide la similitud entre dos *clústeres* midiendo la distancia más corta entre un par de elementos. Tiene la tendencia a producir *clústeres* alargados.
- ♦ *Clustering* de enlace promedio (enlace de varianza mínima o *minimum-variance linkage*): este método encuentra la media o la mediana de las distancias entre todos los puntos de datos entre *clústeres*.

- ◊ Enlace completo (del vecino más lejano o *farthest neighbor*): determina la distancia entre dos *clústeres* midiendo la distancia más larga entre cualquier miembro de un *clúster* y cualquier miembro del otro *clúster*. Los *clústeres* formados por el algoritmo de enlace completo son más compactos y están más ligados que los del *clustering* de enlace simple.

Clústeres particionales

En este tipo de *clústeres* los datos se organizan en una secuencia anidada de grupos sin ninguna estructura jerárquica. Este método es utilizado para conjuntos de datos grandes, para los que la construcción de un dendograma es imposible desde el punto de vista computacional.

El criterio de error cuadrático es la función criterio sobre la cual se basa el algoritmo de *clustering* particional más utilizado. El objetivo general consiste en encontrar la partición para la cual un número fijo de *clústeres* minimiza el error cuadrático. En este caso, las desviaciones de los patrones con respecto a los centroides se representan por el error. El algoritmo de *clustering* particional comienza con una partición inicial del conjunto de datos y asigna iterativamente puntos de datos o patrones a *clústeres* para reducir el error cuadrático. El algoritmo de *clustering* particional puede categorizarse según las diversas técnicas adoptadas en la generación de los *clústeres* y la naturaleza de los *clústeres* resultantes. Estos incluyen *clustering* duro/definido (*hard/crisp clustering*), *clustering* difuso (*mixture resolving*) y *clustering* de resolución de mezclas (*fuzzy clustering*).

Algoritmo de K-means

Es una técnica basada en centroides donde los objetos se distribuyen en un número predefinido de k grupos. La calidad de la partición se evalúa con una función objetivo, buscando maximizar la similitud dentro de un grupo (intragrupo) y minimizar la similitud entre diferentes grupos (intergrupo). K-means utiliza la media para representar el centro de un grupo y agrupa los datos calculando la distancia euclidiana entre los puntos de datos y los centros de los grupos. Sin embargo, K-means tiene desventajas, como la sensibilidad a la selección inicial de los centroides, la dificultad de determinar el número adecuado de grupos y su sensibilidad a los valores atípicos.

Clústeres de uso más común en Bioestadística

Los *clústeres* de tipo jerárquico son los más utilizados en la rama de la Bioestadística, algunas de las razones por las cuales son más populares son:

- ♦ La visualización de los datos a través de dendogramas, lo que genera una representación de manera visual intuitiva y fácil de interpretar, por ejemplo, cuando se analizan similitudes, ya sea entre genes y su expresión, muestras de pacientes o proteínas, lo que permite al investigador observar la forma de agruparse de los datos.
- ♦ No requiere especificar el número de *clústeres*, ya que en el *clustering* no jerárquico no es necesario, pues en la Bioestadística el número de grupos en los datos puede no ser conocido desde el principio.
- ♦ Se utiliza en aplicaciones comunes en Bioestadística, tales como el análisis de expresión génica, lo que permite agrupar genes con patrones de expresión similar pudiendo ser asociados a diversas condiciones médicas; la clasificación de tipos de cáncer puede ser basado en su expresión génica o características moleculares; y agrupamiento de pacientes para la creación de subgrupos con características clínicas o respuestas a tratamientos similares.
- ♦ Flexibilidad en las medidas de distancia (como la euclidiana o correlación de Pearson), haciéndola adaptable a distintos tipos de datos.

Así también los *clústeres* K-means, pertenecientes a la rama de los particionales, son de amplio uso en la Bioestadística. Este tipo de *clustering* es utilizado en la clasificación de datos, como en la segmentación de pacientes en grupos con características similares, la clasificación de genes en función de su expresión, la identificación de subtipos de cáncer, entre otros.

En este libro nos enfocamos en estos tipos de *clústeres*. A continuación, se presentan ejemplos de cómo se pueden implementar en R.

Implementación del análisis de clústeres en R

- ◇ **Ejemplo 27.1:** para resolver este ejemplo de un *clúster* jerárquico divisivo vamos a utilizar el *Dataframe* “BasePCA” que se describió en [“Implementación de un análisis de componentes principales en R”](#).

Primero vamos importar las librerías necesarias, en este caso: `cluster`, `ggplot2`:

```
# Si es necesario instale las librerías
library(cluster) # Para generar los clusters
library(ggplot2) # Para visualizar los clusters
```

Posterior a esto debemos cargar nuestra base de datos a nuestro entorno de R, si guardó su *Dataframe* en su carpeta de Bases, puede cargarlo con el siguiente código:

```
data ← read.csv("Bases/BasePCA.csv") # Cargar la base de datos
```

Por cuestiones didácticas y de visualización solo vamos a utilizar las primeras 50 filas de la base de datos. Se dejó intencionalmente disponible para descargar la base completa para que pueda repetir el proceso con la base completa:

```
data ← data[1:50,] # Solo utilizaremos las primeras 50 filas
```

Con el siguiente código transformaremos la información contenida en la columna `diagnosis` a un factor:

```
data$diagnosis ← as.factor(data$diagnosis) # Convertir a factor
```

Con la siguiente línea de código nos aseguramos de usar solo las columnas numéricas y la columna `diagnosis` como primer criterio para el *clustering*:

```
data_numeric ← data[,c(-1,-2) ] # Seleccionar solo las columnas numéricas y diagnosis
```

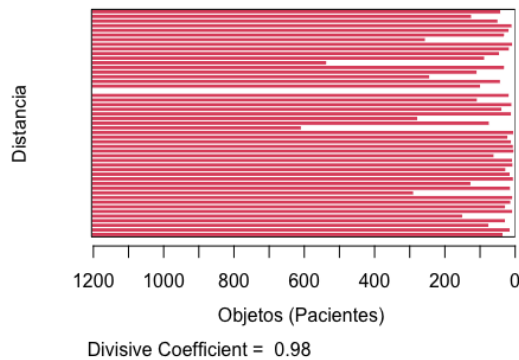
La siguiente línea ejecuta el *clustering* jerárquico divisivo:

```
diana_cluster <- diana(data_numeric)
```

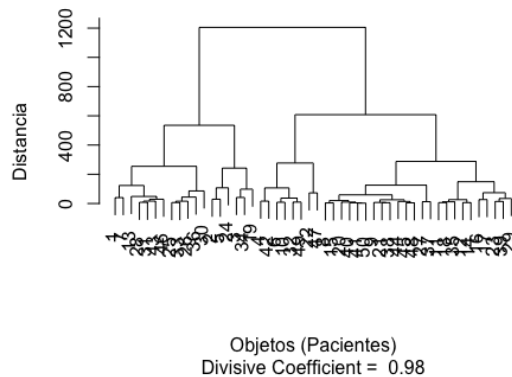
Para visualizar el dendrograma:

```
plot(diana_cluster, main = "Clustering Jerárquico Divisivo (DIANA)",  
     xlab = "Objetos (Pacientes)", ylab = "Distancia")
```

Clustering Jerárquico Divisivo (DIANA)



Clustering Jerárquico Divisivo (DIANA)



En el primer dendrograma, las líneas horizontales representan divisiones jerárquicas en el grupo de objetos. Los objetos que se encuentran más a la derecha fueron separados en un nivel temprano del proceso divisivo, lo que significa que son más diferentes de los otros objetos.

Las líneas largas indican grupos que fueron divididos en grandes “saltos”, mientras que las líneas cortas indican divisiones más pequeñas y detalladas. Al observar cómo se divide el dendrograma, puede ver cuántos grupos se formaron y en qué niveles.

En el segundo dendrograma podemos observar lo siguiente:

- ♦ Estructura de los grupos: se puede observar que el dendrograma comienza con una división principal en dos grandes grupos, y cada grupo se subdivide en subgrupos a diferentes niveles de distancia. Las divisiones más cercanas a la parte superior indican que esos objetos son más disímiles entre sí.
- ♦ Elección del número de *clústeres*: este dendrograma permite identificar visualmente cuántos *clústeres* puedes elegir para tus datos. Si quisieras dividir los datos en, por ejemplo, 3 o 4 grupos, podrías decidir cortar el dendrograma en un nivel donde se observe una clara separación.
- ♦ Variabilidad entre los grupos: a mayor altura de las líneas horizontales, mayor es la distancia entre los grupos que se forman. Las divisiones más abajo del gráfico representan objetos que son más similares entre sí.

Para este ejemplo vamos a seleccionar 4 *clúster*, la selección de la cantidad de agrupamientos no es una respuesta única. Depende de muchos factores, incluida la experiencia del investigador.

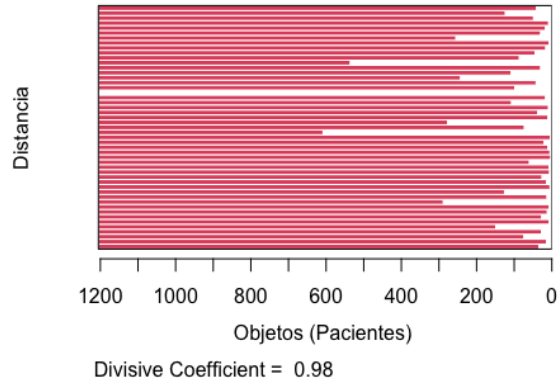
Para crear un corte basado en 4 *clústeres* utilice el siguiente código:

```
clusters_4 <- cutree(as.hclust(diana_cluster),
                     k = 4) # Crear 4 clusters
```

La última parte de este ejemplo es dibujar en el dendrograma los rectángulos que indican 4 *clústeres* que seleccionamos arbitrariamente en el código anterior. Sin embargo, para poder marcar los 4 *clústeres* es necesario correr todo el código junto, de lo contrario R con la función `rect.hclust` no será capaz de marcar los *clústeres*:

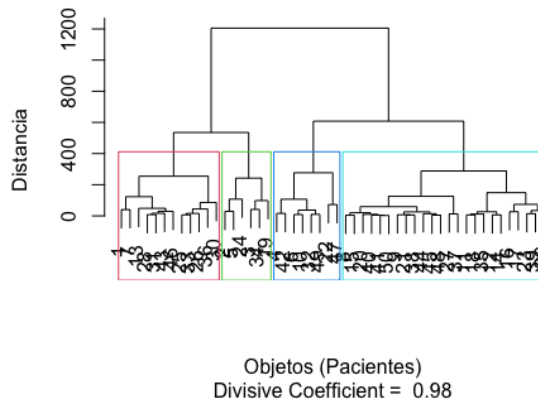

```
plot(diana_cluster, main = "Clustering Jerárquico Divisivo (DIANA)",
     xlab = "Objetos (Pacientes)", ylab = "Distancia") # Dendrograma
```

Clustering Jerárquico Divisivo (DIANA)



```
clusters_4 <- cutree(as.hclust(diana_cluster),
                     k = 4) # Crear 4 clusters
rect.hclust(as.hclust(diana_cluster), k = 4,
            border = 2:5) # Marcar 4 clusters
```

Clustering Jerárquico Divisivo (DIANA)



```
# border = 2:5 indica los colores de los bordes de los rectángulos
```

Este dendrograma muestra cómo el algoritmo DIANA ha dividido a los pacientes en 4 *clústeres* bien definidos, con una estructura jerárquica clara que refleja las similitudes y diferencias entre los pacientes. El coeficiente divisivo alto (0.98) sugiere que los grupos obtenidos son coherentes y bien diferenciados.

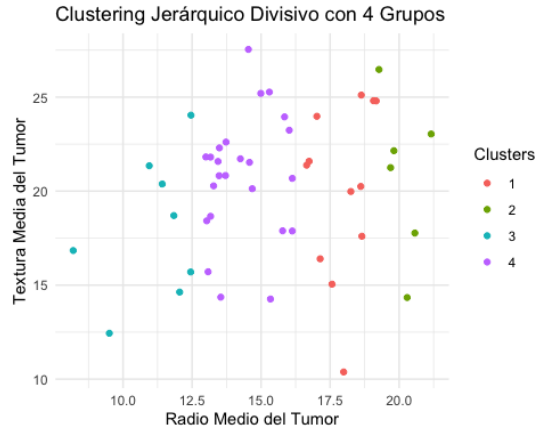
Esta línea agrega los *clústeres* al *Dataframe*:

```
data$cluster_4 ← clusters_4
```

Con los *clústeres* creados, podemos visualizarlos en un gráfico de dispersión para explorar cómo se agrupan los pacientes en función de sus características de radio y textura de los tumores.

Visualizar los *clústeres* divididos en base al *clustering* jerárquico divisivo:

```
ggplot(data, aes(x = radius_mean, y = texture_mean,
                  color = as.factor(cluster_4))) +
  geom_point() +
  ggtitle("Clustering Jerárquico Divisivo con 4 Grupos") +
  xlab("Radio Medio del Tumor") +
  ylab("Textura Media del Tumor") +
  theme_minimal() +
  theme(legend.position = "right") +
  scale_color_discrete(name = "Clusters") # Cambiar el título de la
leyenda
```



La última imagen es un gráfico de dispersión del análisis de *clustering* jerárquico divisivo (DIANA), en el que se han creado 4 grupos. La gráfica tiene como variables:

- ♦ Eje de las X: “Radio Medio del Tumor” (*radius_mean*), que indica el tamaño promedio del tumor.
- ♦ Eje de las Y: “Textura Media del Tumor” (*texture_mean*), que se refiere a las características de la textura del tumor.

Los puntos en el gráfico representan a los pacientes, y el color de los puntos indica a qué *clúster* pertenece cada paciente, donde se agruparon en 4 *clústeres* diferentes, según la leyenda de colores a la derecha del gráfico.

Interpretación: los pacientes están distribuidos en función del radio y la textura de sus tumores. Los diferentes colores representan las asignaciones a los *clústeres*:

- ♦ Rojo para el *clúster* 1
- ♦ Verde para el *clúster* 2
- ♦ Azul para el *clúster* 3
- ♦ Morado para el *clúster* 4

Separación de los *clústeres*: 1) el *clúster* 1 (rojo) parece agrupar a pacientes con valores de radio entre aproximadamente 15 y 20, y textura media variable, mayormente concentrada en el rango de 18 a 24. 2) El *clúster* 2 (verde) tiene

menos puntos, pero agrupa a pacientes con tumores de mayor tamaño (radio entre 17.5 y 22.5) y texturas en un rango diverso. 3) El *clúster* 3 (azul) incluye a pacientes con radios más pequeños (menores a 12.5) y una textura media entre 15 y 21. 4) El *clúster* 4 (morado) parece ser el grupo más grande y diverso, con radios medios que varían entre 12.5 y 17.5 y texturas más altas, entre 18 y 25.

Tendencias visibles: los *clústeres* parecen estar bien diferenciados en función del tamaño y la textura de los tumores. Los grupos 1 y 4 (rojo y morado) están más concentrados en la parte central del gráfico, mientras que los *clústeres* 2 y 3 (verde y azul) ocupan los extremos, sugiriendo que las diferencias en radio y textura juegan un papel importante en la separación de los grupos. Esto podría sugerir una clasificación de los pacientes basada en el tamaño y la textura de los tumores, lo cual es relevante para la caracterización de los tumores en términos de su potencial de malignidad o benignidad, aunque se necesitaría más análisis para confirmar cualquier conclusión clínica.

♦ **Ejemplo 27.2:** ahora realizaremos un *clúster* particional con el método de clustering K-means utilizando el mismo *Dataframe* BasePCA descrita previamente en [“Implementación de un análisis de componentes principales en R”](#).

Vamos retomar nuestro *Dataframe* BasePCA, pero esta vez utilizando todos los datos, no solo las 50 primeras filas, por lo que es necesario volver a importar la base de datos que ahora llamaremos df, para no interferir con el *Dataframe* anterior (data):

```
df ← read.csv("Bases/BasePCA.csv") # Cargar la base de datos
```

En el caso de los K-means solo requerimos variables numéricas, por lo que vamos a seleccionar solo las columnas que cumplan con esta característica:

```
df_numeric ← df[,c(-1,-2, -3) ] # Seleccionar solo las columnas numéricas
```

Ahora nuestro *Dataframe* df_numeric solo tiene variables numéricas, por lo que podemos proceder a realizar el clustering con el método K-means

En este caso es necesario normalizar nuestras variables, al igual que lo hicimos en “Implementación de un análisis de componentes principales en R”, para esto utilizaremos la función `scale`:

```
df_scaled ← scale(df_numeric) # Normalizar las variables
```

Dado que el *Dataframe* original tiene una variable categórica (*diagnosis*), vamos a realizar un agrupamiento con dos *clústeres* para ver si se pueden diferenciar los tumores benignos de los malignos:

```
# Aplicar k-means con k = 2 (dos clústeres)
set.seed(123) # Sembrar la semilla para reproducibilidad
kmeans_result ← kmeans(df_scaled, centers = 2, nstart = 25)

# Añadir los clústeres resultantes al dataframe original
df$cluster ← as.factor(kmeans_result$cluster)
```

Procedemos a comparar los resultados con la variable original *diagnosis*, note cómo utilizamos el *Dataframe* original *df* y no el *df_numeric* o *df_scaled*:

```
table(df$diagnosis, df$cluster)
```

	1	2
B	5	352
M	165	47

El resultado anterior indica lo siguiente:

- ♦ En el *clúster* 1: hay 165 observaciones con diagnóstico maligno (M) que fueron asignadas a este *clúster*; 5 observaciones con diagnóstico benigno (B) también fueron asignadas a este *clúster*.
- ♦ En el *clúster* 2: hay 352 observaciones con diagnóstico benigno (B) que fueron asignadas correctamente a este *clúster* y 47 observaciones con diagnóstico maligno (M) también fueron asignadas aquí.

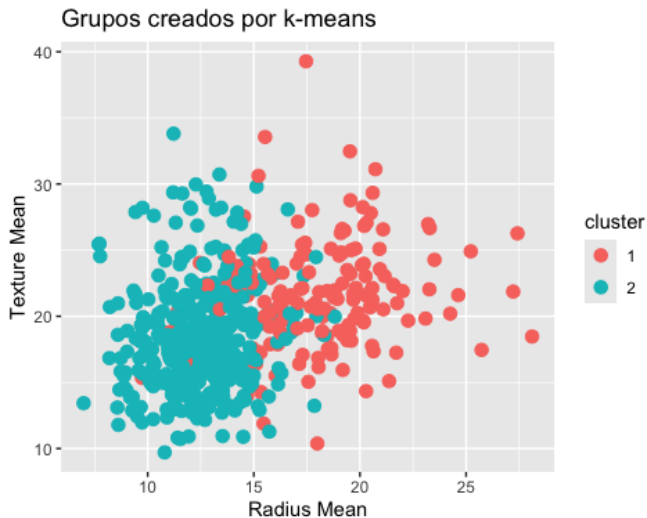
El *clúster* 1 parece representar principalmente los casos malignos, ya que contiene una gran cantidad de diagnósticos malignos (165), pero también incluye un pequeño número de casos benignos (5), lo que indica que hay algunas observaciones benignas mal clasificadas. Por su parte, el *clúster* 2 parece representar principalmente los casos benignos, ya que agrupa una gran cantidad de diagnósticos benignos (352), pero también contiene 47 casos malignos, lo que indica algunos errores de clasificación.

En conclusión, podríamos decir que el algoritmo K-means ha hecho un buen trabajo separando los casos malignos y benignos, aunque no es perfecto.

Ahora procedemos a graficar los resultados:

```
# Cargar ggplot2 si es necesario

# Gráfico de dispersión con los clústeres
ggplot(df, aes(x = radius_mean,
               y = texture_mean, color = cluster)) +
  geom_point(size = 3) +
  labs(title = "Grupos creados por k-means",
       x = "Radius Mean", y = "Texture Mean")
```



El gráfico de dispersión muestra los datos de los tumores de mama en función del radio medio y la textura media, con los puntos coloreados según el *clúster* asignado por el algoritmo K-means. Los puntos se agrupan en dos *clústeres*, que se distinguen por los colores azul y rojo. Los tumores benignos y malignos se agrupan en diferentes *clústeres*, lo que sugiere que el algoritmo K-means ha logrado separar efectivamente los dos tipos de tumores en función de sus características de radio y textura.

Selección del número de clústeres

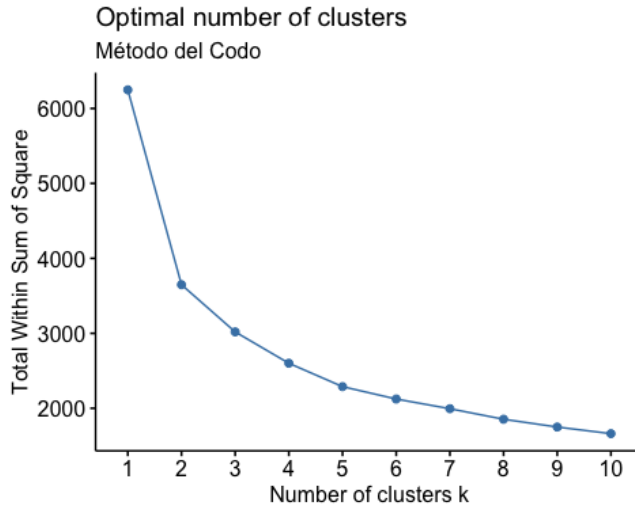
Hasta ahora hemos seleccionado arbitrariamente el número de *clústeres*, sin embargo, en la práctica, la selección del número de *clústeres* es un paso crítico en el análisis de estos. La elección del número de *clústeres* puede depender de varios factores, como el objetivo del análisis, la naturaleza de los datos y la interpretación de los resultados. A continuación, se presentan algunas técnicas comunes para seleccionar el número de *clústeres*.

Método del codo

Ya hemos hablado de él. Consiste en graficar la suma de las distancias al cuadrado de cada punto al centroide más cercano en función del número de *clústeres*. El punto en el gráfico donde la curva comienza a aplanarse se conoce como el “codo”, y se considera un buen candidato para el número de *clústeres*. Para el ejemplo anterior podemos utilizar el siguiente código:

```
# Cargar las librerías
library(factoextra)

# Aplicar el método del codo usando la función fviz_nbclust
fviz_nbclust(df_scaled, kmeans, method = "wss") +
  labs(subtitle = "Método del Codo")
```



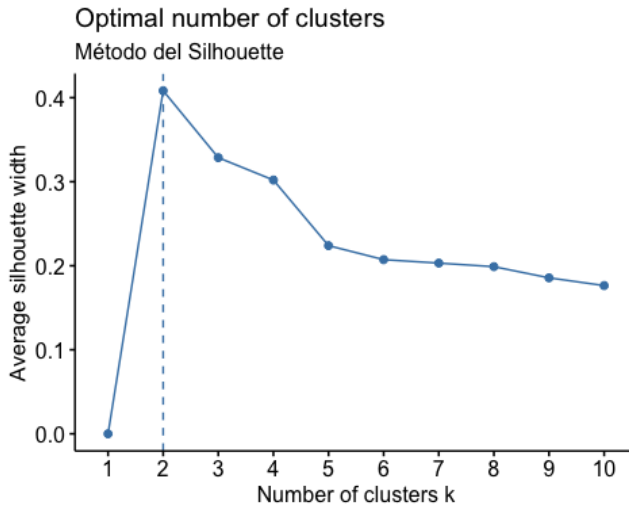
Note cómo se utiliza la función `fviz_nbclust` de la librería `factoextra` para realizar el método del codo y que el *Dataframe* utilizado es `df_scaled` que es el *Dataframe* normalizado.

Es muy evidente dónde se encuentra el doblez del codo a los 2 *clústeres*, lo que confirma la elección de 2 *clústeres* que hicimos anteriormente.

Silhouette analysis

Este análisis evalúa qué tan bien separadas están las observaciones en diferentes *clústeres*, ayudando a determinar el número adecuado de *clústeres*. Puede emplear el siguiente código para realizar el análisis de silueta:

```
fviz_nbclust(df_scaled, kmeans, method = "silhouette") +
  labs(subtitle = "Método del Silhouette")
```

También 2 *clústeres* es el número óptimo, según el análisis de silueta.

Ejercicios

- ♦ **Ejercicio 27.1:** utilizando la base de datos `bodyfat.txt`, genere un *clúster* jerárquico divisivo.

Resolución ejercicio 27.1

```
# Cargar las bibliotecas necesarias
library(cluster)
library(ggplot2)
library(dplyr)

# Cargar los datos desde un archivo .txt (separado por tabulaciones)
data <- read.table("Bases/bodyfat.txt", header = TRUE, sep = " ")
data <- data[1:50,]

# Seleccionar solo las columnas numéricas para el clustering
data_numeric <- data %>%
```

```

select(Density, Fat, Age, Weight, Height,
       Neck, Chest, Abdomen, Hip, Thigh, Knee,
       Ankle, Biceps, Forearm, Wrist)

# Ejecutar el clustering jerárquico divisivo (DIANA)
diana_cluster ← diana(data_numeric)

# Convertir a un objeto de tipo hclust para poder manipularlo con
corte
hclust_diana ← as.hclust(diana_cluster)

# Crear un corte basado en 4 clústeres
clusters_4 ← cutree(hclust_diana, k = 4)

# Agregar los clusters al dataset original
data$cluster_4 ← as.factor(clusters_4)

# Visualización del dendrograma cortado en 4 clústeres
plot(hclust_diana, main = "Dendrograma Jerárquico con 4 Clústeres",
     labels = FALSE)

# Añadir una línea de corte para los 4 clústeres
abline(h = hclust_diana$height[length(hclust_diana$height) - 3], col =
"red")

# Dibujar los rectángulos alrededor de los 4 clústeres
rect.hclust(hclust_diana, k = 4, border = 2:5)

```

- ♦ **Ejercicio 27.2:** utilizando la base de datos [bodyfat.txt] (https://drive.google.com/file/d/1Mz9N326Td-PqoMZpxPRBHS08VJpsX_o/view), genere un *clúster* particional.

Resolución ejercicio 27.2

```
df ← read.table("Bases/bodyfat.txt", header = TRUE, sep = " ")
```

```

df_scaled ← scale(df) # Normalizar las variables
# Cargar las librerías
library(factoextra)
fviz_nbclust(df_scaled, kmeans, method = "wss") +
  labs(subtitle = "Método del Codo")
fviz_nbclust(df_scaled, kmeans, method = "silhouette") +
  labs(subtitle = "Método del Silhouette")
# Aplicar k-means con k = 2 (dos clústeres)
set.seed(123)
kmeans_result ← kmeans(df_scaled, centers = 2, nstart = 25)

# Añadir los clústeres resultantes al dataframe original
df$cluster ← as.factor(kmeans_result$cluster)
# Gráfico de dispersión con los clústeres
ggplot(df, aes(x = radius_mean,
               y = texture_mean, color = cluster)) +
  geom_point(size = 3) +
  labs(title = "Grupos creados por k-means",
       x = "Radius Mean", y = "Texture Mean")

```

Referencias

- Chen, Y. T., y Witten, D. M. (2023). Selective inference for k-means clustering. *Journal of Machine Learning Research*, 24, 152. PMID: 38264325; PMCID: PMC10805457.
- Dafir, Y., Benabdellah, A., Mouline, S., y Ahouzi, A. (2021). Recent advances in clustering methods applied to big data. *Engineering Applications of Artificial Intelligence*, 110, 104743. <https://doi.org/10.1016/j.engappai.2022.104743>.
- Dalmaijer, E. S., Nord, C. L., y Astle, D. E. (2022). Statistical power for cluster analysis. *BMC Bioinformatics*, 23(1), 205. <https://doi.org/10.1186/s12859-022-04675-1>. PMID: 35641905; PMCID: PMC9158113.
- Daniel, W. W., y Cross, C. L. (2018). *Biostatistics: A foundation for analysis in the health sciences* (11th ed.). John Wiley & Sons.

- Eckhardt, C. M., Madjarova, S. J., Williams, R. J., Ollivier, M., Karlsson, J., Pareek, A., y Nwachukwu, B. U. (2023). Unsupervised machine learning methods and emerging applications in healthcare. *Knee Surgery, Sports Traumatology, Arthroscopy*, 31(2), 376-381. <https://doi.org/10.1007/s00167-022-07233-7>. Epub 2022 Nov 15. PMID: 36378293.
- Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Agushaka, J. O., Eke, C. I., y Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110, 104743. <https://doi.org/10.1016/j.engappai.2022.104743>.
- Hastie, T., Tibshirani, R., y Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
- James, G., Witten, D., Hastie, T., y Tibshirani, R. (2023). *An introduction to statistical learning with applications in R* (2nd ed.). Springer.
- Johnson, R. A., y Wichern, D. W. (2007). *Applied multivariate statistical analysis* (6th ed.). Pearson Prentice Hall.
- McLachlan, G. J., Bean, R. W., y Ng, S. K. (2017). Clustering. *Methods in Molecular Biology*, 1526, 345-362. https://doi.org/10.1007/978-1-4939-6613-4_19. PMID: 27896751.
- Nowak-Brzezińska, A., y Gaibei, I. (2022). How the Outliers Influence the Quality of Clustering? *Entropy (Basel)*, 24(7), 917. <https://doi.org/10.3390/e24070917>. PMID: 35885141; PMCID: PMC9324173.
- Peterson, A. D., Ghosh, A. P., y Maitra, R. (2018). Merging K-means with hierarchical clustering for identifying general-shaped groups. *Statistics (International Statistical Institute)*, 7(1), e172. <https://doi.org/10.1002/sta4.172>. PMID: 29736237; PMCID: PMC5935272.
- Torgo, L. (2016). *Data mining with R: Learning with case studies* (2nd ed.). Chapman and Hall/CRC.
- Wickham, H., y Grolemund, G. (2016). *R for data science: Import, tidy, transform, visualize, and model data*. O'Reilly Media.

Análisis de sobrevida

Jesús Jonathan García Galindo

Introducción

Imagine que se encuentra frente a una población con artritis reumatoide en la cual usted quiere determinar la cantidad de recaídas asociadas al tiempo de uso de metrotexate v.s el uso de metrotexate mas esteroide. Para ello determinamos un tiempo de seguimiento de 5 años, en el cual el momento de la recaída se convierte en nuestro evento de interés, por lo tanto, al final podremos determinar qué porcentaje de pacientes recayó en un tiempo determinado, así como la comparación entre los diferentes grupos y la diferencia entre los dos tratamientos.

Para poder evaluar el ejemplo anterior tenemos diferentes herramientas, la más utilizada es el análisis de supervivencia, el cual utiliza métodos, herramientas, habilidades de procesamiento y utilización de datos con distribución no paramétrica en cuyo principal modelo encontramos la estimación “curva Kaplan-Meier” (KM). Además, existe la posibilidad de comparar distintas curvas mediante la prueba de log-rank y así encontrar diferencias de supervivencia en distintos grupos. Pero para que estos modelos tengan la confiabilidad deseada se debe examinar y comunicar de forma más sofisticada los métodos estadísticos relacionados con el manejo de los datos de supervivencia, pero al mismo tiempo deben ser capaces de adaptarse a los problemas reales de las aplicaciones clínicas.

Esta herramienta estadística es utilizada principalmente para la investigación de datos recolectados en un periodo de tiempo específico, como en el ejemplo inicial. Los principales diseños de estudio en los que se utiliza son cohortes y ensayos clínicos.

Tiempo para el evento

Primero, debe identificar a qué se refiere el concepto de evento de interés; esto se refiere al desenlace de la investigación en el caso de estas curvas a “muerte” o “mejoría a través del tratamiento”. En el caso del ejemplo inicial nuestro evento de interés se refiere a la recaída de pacientes. El segundo concepto que debe identificar es el tiempo de observación definido por el investigador, como en el ejemplo que se definió a 5 años, pero que pasa si el evento no sucede en el tiempo en el cual se está observando por el investigador; esto se define como “observación censurada”, lo cual es una ventaja de este tipo de trabajos, ya que se pueden incluir a los sujetos que no completen el estudio o no alcancen el punto final del mismo.

Estas observaciones censuradas pueden ser “derechas”, en el cual solo se conoce el desenlace (evento de interés) de los sujetos mientras son observados, pero desconocemos el desenlace de los sujetos que dejamos de observar. En cambio, las observaciones censuradas “izquierdas” ocurren cuando el diagnóstico o tiempo de exposición es desconocido. Existe también la censura del tiempo para el evento cuando el sujeto se mueve dentro o fuera durante el periodo en que debe ser observado o cuando el evento ocurre para un sujeto en dos puntos distintos en el periodo de observación, pero el punto preciso del evento es desconocido. La hipótesis de este tipo de estudio presupone que las “censuras” suceden independientemente del “evento de interés”.

Funciones de sobrevida

Usted debe entender a la función de sobrevida como la existencia del evento de interés en un momento de tiempo determinado. En una gráfica de KM la curva se mantendrá horizontal mientras el evento no ocurra; en el momento en el que el evento ocurra se expresará como una caída de la curva en caso de que el evento sea “muerte” o como un ascenso en caso de que el evento sea “mejoría a través del tratamiento”.

La densidad de probabilidad de la función se define como: la frecuencia de eventos por unidad de tiempo y se consigue por la derivada de la función de sobrevida en relación con el tiempo y se expresa con la siguiente ecuación:

$$S(t) = P(T > t) \text{ con } 0 \leq S(t) \leq 1 \text{ y } 0 < t < \infty$$

La representación de la función de supervivencia es la curva de KM, la cual se mantendrá en su eje horizontal mientras el evento esperado no suceda cuando el evento sucede corresponde al cambio en la función de sobrevida.

El lector también debe recordar la función Hazard (daño). Esta función sirve para calcular la probabilidad de que suceda el evento de interés para un sujeto determinado en un intervalo infinitesimal de tiempo expresado, como delta de tiempo, y se calcula con la siguiente ecuación:

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t < T < t + \Delta t \mid T > t)}{\Delta t}$$

Curvas Kaplan-Meier

El gráfico de la curva de KM requiere que se cumplan tres consideraciones fundamentales:

- ♦ El sujeto que es censurado en cualquier momento del estudio tiene la misma posibilidad de sobrevida que los sujetos quienes no fueron censurados.
- ♦ La probabilidad de sobrevida es la misma para sujetos independientemente del momento en el cual fueron reclutados.
- ♦ El evento debe ocurrir en el tiempo establecido.

El estimado de la función de sobrevida propuesta con el modelo de la curva de KM se obtiene al dividir en tiempo total (intervalo total) entre el tiempo en donde ocurre el momento del fallo (subintervalos) y con ello calcula las probabilidades de sobrevida de cada intervalo; cada evento es representado con alteración (“descenso o salto”) del gráfico en su eje horizontal.

Los resultados obtenidos se expresan como mediana de sobrevida. Por ejemplo, en un caso hipotético la mediana de sobrevida obtenida fue de 11.04 (10.65-11.81) por año con un intervalo de confianza (IC) de 95%.

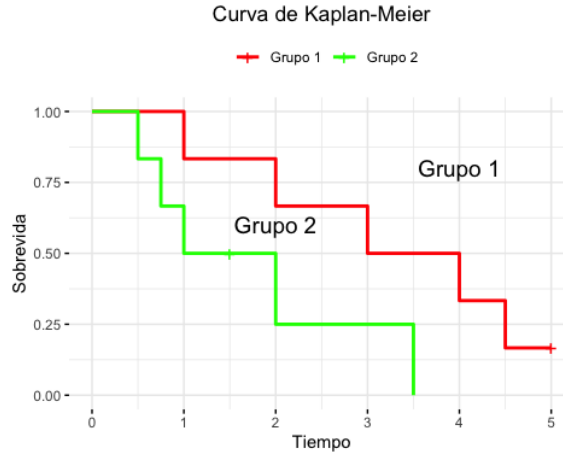
Comparación de curvas de Kaplan-Meier

Este proceso tiene la utilidad de comparar las curvas de supervivencia entre diferentes grupos, lo cual genera su significancia estadística al medir la diferencia de probabilidad de sobrevivida. La prueba estadística más usada para realizar esta función es la prueba de log-rank (LR), la cual compara el número de eventos observados con el número de eventos esperados en dos grupos independientes mediante la comparación de (X^2) correspondiente a cada curva.

En un hospital de México se siguió a una serie de sujetos con diagnóstico de neoplasia maligna de glándula parótida. Fueron separados en 2 grupos: 1) aquellos que fueron sometidos a parotidectomía total (grupo 1) y 2) aquellos que fueron sometidos a parotidectomía radical (grupo 2). Los sujetos tuvieron un seguimiento de 5 años, en total fueron observados 6 sujetos por cada grupo, y los datos y análisis se describen a continuación.

SUJETO	TIEMPO-EVENTO (AÑOS)	ESTADO DEL EVENTO (1 = MUERTE 0 = CENSURADO)	GRUPO
ABR	1	1	1
DOS	2	1	1
ASR	3	1	1
JGG	4	1	1
IMP	4.5	1	1
RAB	5	0	1
EED	0.5	1	2
LMP	0.75	1	2
BSG	1	1	2
EAG	1.5	0	2
SGS	2	1	2
JJG	3.5	1	2

Con los datos anteriores se calculó una curva de supervivencia, la cual podemos observar a continuación (ver [figura 28.1](#)).

FIGURA 28.1. CURVA DE KAPLAN MEIER

En este ejemplo hipotético los resultados obtenidos fueron que existe 83.3% de sobrevida a un año en el grupo 1, mientras en el grupo 2 es de 66.7 %; además de los datos anteriores se sabe que el grupo 2 tuvo una mortalidad del 100% a los 3.5 años, mientras el grupo 1 antes del año presenta una sobrevida del 100%. En la mitad del tiempo a los 3 años el grupo 1 presentó una sobrevida de 66.7%, mientras el grupo 2 presentó 25%. A pesar de que a simple vista se pueda inferir que existe una diferencia significativa entre los 2 grupos al momento de realizar la prueba de log-rank se obtiene una $p = 0.80$, lo cual no muestra diferencia estadísticamente significativa.

Construcción de una curva de Kaplan-Meier en R

Para este ejemplo vamos a replicar la [figura 28.1](#).

Primero vamos a cargar las librerías necesarias:

```
# Si es necesario instalar previamente
library(survival)
library(survminer)
library(ggplot2)
```

Ahora vamos a generar los datos del ejemplo:

```
# Datos de ejemplo
set.seed(123) # Sembrar semilla
time <- c(1, 2, 3, 4, 4.5, 5, 0.5, 0.75, 1, 1.5, 2, 3.5) #Tiempo
status <- c(1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1) # Evento
group <- c(rep("Grupo 1", 6), rep("Grupo 2", 6))# Objeto con los grupos
data_sobrevida <- data.frame(group, status, time)# Creación de un DataFrame
```

Puede visualizar los primeros 6 datos de su base de datos:

```
head(data_sobrevida)
  group status time
1 Grupo 1     1  1.0
2 Grupo 1     1  2.0
3 Grupo 1     1  3.0
4 Grupo 1     1  4.0
5 Grupo 1     1  4.5
6 Grupo 1     0  5.0
```

Para crear la curva de sobrevivencia puede utilizar el siguiente código, recuerde que siempre puede copiar y pegar su código:

```
# Crear el objeto de supervivencia
surv_object <- Surv(time, status)

# Ajustar el modelo de Kaplan-Meier
fit <- survfit(surv_object ~ group)

# Generar la gráfica
plot <- ggsurvplot(fit, data = data.frame(time, status, group),
                  pval = FALSE, conf.int = FALSE,
                  risk.table = FALSE,
                  legend.title = "",
```

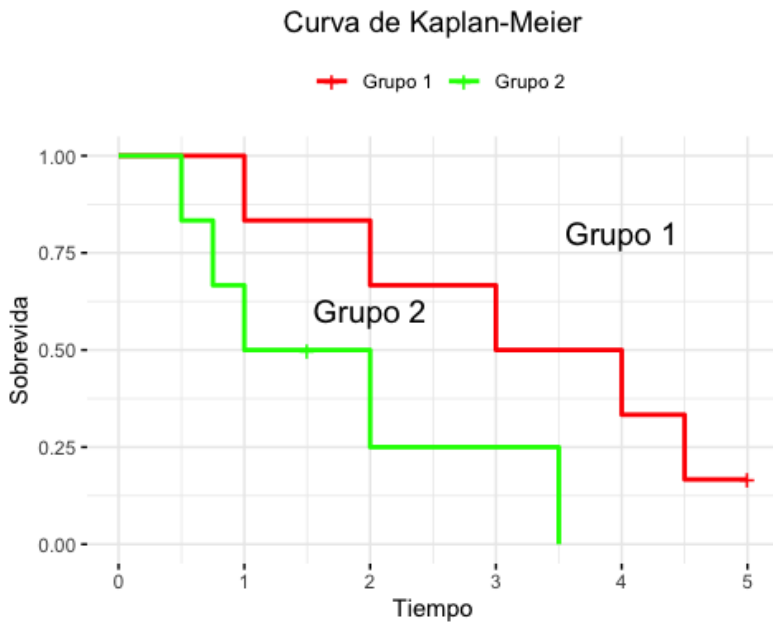
```

legend.labs = c("Grupo 1", "Grupo 2"),
xlab = "Tiempo", ylab = "Supervivencia",
ggtheme = theme_minimal(),
palette = c("red", "green"),
surv.median.line = "none")

# Añadir anotaciones y título
plot$plot <- plot$plot +
  annotate("text", x = 2, y = 0.6, label = "Grupo 2", size = 5) +
  annotate("text", x = 4, y = 0.8, label = "Grupo 1", size = 5) +
  ggtitle("Curva de Kaplan-Meier") +
  theme(plot.title = element_text(hjust = 0.5))

# Mostrar la gráfica
print(plot)

```



En el código anterior lo primero que realizamos fue crear el objeto `surv_object`. Para ello, utilizamos la función `survfit()`, la cual se alimenta de dos vectores: 1) `time`: vector que contiene los tiempos de seguimiento o tiempo hasta el evento y 2) `status`: vector que indica si el evento de interés (por ejemplo, muerte, recaída) ocurrió (generalmente codificado como 1) o si los datos están censurados (generalmente codificado como 0).

Posteriormente, ajustamos nuestro objeto al modelo de Kaplan-Meier utilizando la función `fit()`. Finalmente, graficamos nuestra curva.

En la gráfica que creamos podemos observar lo siguiente: el grupo 1 muestra un mejor perfil de supervivencia en comparación con el grupo 2 durante todo el periodo observado, el grupo 2 muestra mayor número de fallecimientos en los primeros intervalos en comparación al grupo 1, a simple vista el grupo 1 mantiene una probabilidad de supervivencia más alta en cada punto de tiempo comparado con el grupo 2, indicando que los individuos en el grupo 1 tienen una mayor probabilidad de sobrevivir más tiempo en comparación con los del grupo 2, pero cuando se aplica el test de log rank sabemos que no existe diferencia estadística significativa.

Ejercicios

Para la resolución se requiere, además del paquete base de R, las siguientes librerías:

- ♦ `survival`
 - ♦ `survminer`
 - ♦ `ggplot2`
 - ♦ `dplyr`
 - ♦ `readxl`
- ♦ **Ejercicio 28.1:** para este ejercicio vamos a utilizar una adaptación de la base de datos `myeloma` de la librerías `survminer` para establecer si existe o no diferencia entre la supervivencia de dos grupos: el primero con tratamiento 1 (TTO1) y el segundo con tratamiento 2 (TTO2). En cada grupo existen 80 sujetos con diferentes números de eventos y diferente tiempo de seguimiento. Puede descargar esta base [aquí](#).

Resolución ejercicio 28.1**Paso 1. Cargar las librerías**

```

if (!requireNamespace("survival", quietly = TRUE)) {
  install.packages("survival")
}
if (!requireNamespace("survminer", quietly = TRUE)) {
  install.packages("survminer")
}
if (!requireNamespace("readxl", quietly = TRUE)) {
  install.packages("readxl")
}

library(survival)
library(survminer)
library(ggplot2)
library(readxl)
library(dplyr)

```

Paso 2. Cargar la base de datos

Si al descargar su base de datos la guardó en la carpeta de Bases, puede utilizar el siguiente código para cargarla:

```

data1 ← read_excel("Bases/Base_Sobrevida_1.xlsx", sheet = "Hoja1") #Grupo 1
head(data1)

str(data1)
function (... , list = character(), package = NULL, lib.loc = NULL,
verbose = getOption("verbose"),
  envir = .GlobalEnv, overwrite = TRUE)
data2 ← read_excel("Bases/Base_Sobrevida_1.xlsx", sheet = "Hoja2") #Grupo 2
head(data2)

# A tibble: 6 × 3
  Tratamiento Evento Tiempo

```

```

  <chr>      <dbl> <dbl>
1 TT2        1    1.6
2 TT2        0   57.7

3 TT2        0   51.0
4 TT2        1   49.4
5 TT2        0   57.8
6 TT2        0   56.2
str(data2)
tibble [80 × 3] (S3: tbl_df/tbl/data.frame)
 $ Tratamiento: chr [1:80] ""T"" ""T"" ""T"" ""T2" ...
 $ Evento      : num [1:80] 1 0 0 1 0 0 0 0 0 0 ...
 $ Tiempo      : num [1:80] 1.6 57.7 51 49.4 57.8 ...

```

A continuación, crearemos las curvas de Kaplan-Meier para cada grupo:

```

#Grupo 1
# Crear el data frame con las columnas necesarias
data1 <- data.frame(
  Tiempo = data1$Tiempo,
  Evento = data1$Evento,
  Tratamiento = data1$Tratamiento
)
# Crear el objeto de supervivencia
surv_object1 <- Surv(data1$Tiempo, data1$Evento)

# Ajustar el modelo de Kaplan-Meier
fit1 <- survfit(surv_object1 ~ data1$Tratamiento, data = data1)
#Grupo 2
# Crear el data frame con las columnas necesarias
data2 <- data.frame(

```

```

    Tiempo = data2$Tiempo,
    Evento = data2$Evento,
    Tratamiento = data2$Tratamiento
  )

# Crear el objeto de supervivencia
surv_object2 <- Surv(data2$Tiempo, data2$Evento)

# Ajustar el modelo de Kaplan-Meier
fit2 <- survfit(surv_object2 ~ Tratamiento, data = data2)

```

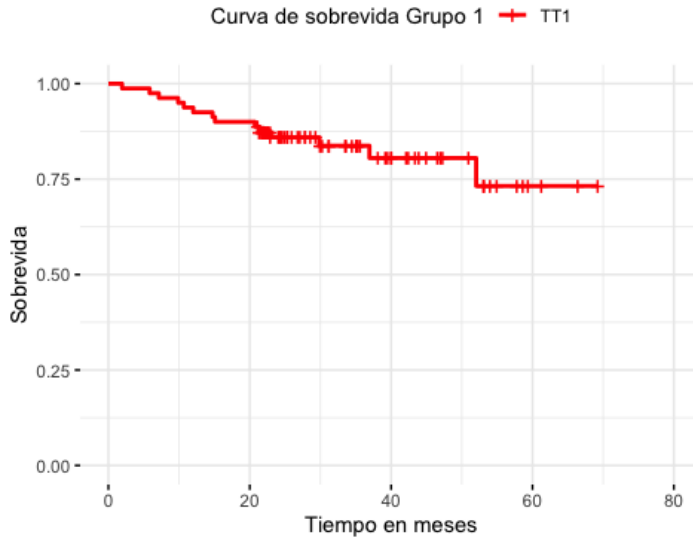
Una vez preparado todo comenzamos con la creación de las gráficas:

```

plot <- ggsurvplot(
  fit1,
  data = data1,
  pval = FALSE,
  conf.int = FALSE,
  risk.table = FALSE,
  legend.title = "Curva de sobrevida Grupo 1",
  legend.labs = c("TT1"),
  xlab = "Tiempo en meses",
  ylab = "Sobrevida",
  ggtheme = theme_minimal(),
  palette = c("red"),
  surv.median.line = "none"
)

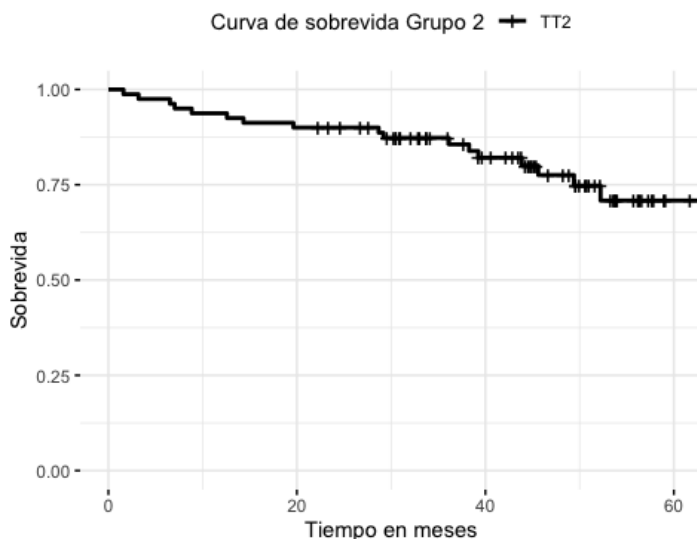
print(plot)

```



```
plot ← ggsurvplot(
  fit2,
  data = data2,
  pval = FALSE,
  conf.int = FALSE,
  risk.table = FALSE,
  legend.title = "Curva de sobrevida Grupo 2",
  legend.labs = c("TT2"),
  xlab = "Tiempo en meses",
  ylab = "Sobrevida",
  ggtheme = theme_minimal(),
  palette = c("black"),
  surv.median.line = "none"
)

print(plot)
```

Usted ahora puede proceder a realizar las tablas donde se represente la estadística de las curvas que generó anteriormente:

```
#Tabla de datos del Grupo 1
```

```
#Generar tabla de contenido
```

```
linelistsurv_fit <- survival::survfit(surv_object1 ~ 1)
```

```
summary(linelistsurv_fit)
```

```
Call: survfit(formula = surv_object1 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
1.93	80	1	0.988	0.0124	0.963	1.000
5.83	79	1	0.975	0.0175	0.941	1.000
7.13	78	1	0.963	0.0212	0.922	1.000
9.87	77	1	0.950	0.0244	0.903	0.999
10.67	76	1	0.938	0.0271	0.886	0.992
12.00	75	1	0.925	0.0294	0.869	0.985
14.73	74	1	0.913	0.0316	0.853	0.977
15.08	73	1	0.900	0.0335	0.837	0.968

```
20.73    72      1    0.888  0.0353      0.821    0.960
21.37    70      1    0.875  0.0370      0.805    0.950
22.97    57      1    0.859  0.0394      0.786    0.940
29.83    39      1    0.837  0.0442      0.755    0.929
36.93    26      1    0.805  0.0529      0.708    0.916
52.03    11      1    0.732  0.0848      0.583    0.919
```

#con la siguiente opción puedo estimar el por sección de tiempo

```
summary(linelistsurv_fit, times = c(6,12,24,36,48,60))
```

```
Call: survfit(formula = surv_object1 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
6	78	2	0.975	0.0175	0.941	1.000
12	75	4	0.925	0.0294	0.869	0.985
24	54	5	0.859	0.0394	0.786	0.940
36	26	1	0.837	0.0442	0.755	0.929
48	12	1	0.805	0.0529	0.708	0.916
60	3	1	0.732	0.0848	0.583	0.919

imprimir el objeto linelistsurv_fit con el tiempo medio de supervivencia y su se.

```
print(linelistsurv_fit, print.rmean = TRUE)
```

```
Call: survfit(formula = surv_object1 ~ 1)
```

```
      n events rmean* se(rmean) median 0.95LCL 0.95UCL
[1,] 80      14  58.2      2.71    NA      NA      NA
      * restricted mean with upper limit = 69.2
```

#Tabla de datos del Grupo 2

#Generar tabla de contenido

```
linelistsurv_fit <- survival::survfit(surv_object2 ~ 1)
```

```
summary(linelistsurv_fit)
```

```
Call: survfit(formula = surv_object2 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
1.60	80	1	0.988	0.0124	0.963	1.000
3.20	79	1	0.975	0.0175	0.941	1.000

6.53	78	1	0.963	0.0212	0.922	1.000
7.03	77	1	0.950	0.0244	0.903	0.999
8.84	76	1	0.938	0.0271	0.886	0.992
12.58	75	1	0.925	0.0294	0.869	0.985
14.33	74	1	0.913	0.0316	0.853	0.977
19.63	73	1	0.900	0.0335	0.837	0.968
28.67	67	1	0.887	0.0356	0.819	0.959
29.18	66	1	0.873	0.0375	0.803	0.950
36.13	51	1	0.856	0.0405	0.780	0.939
38.27	49	1	0.839	0.0433	0.758	0.928
39.25	48	1	0.821	0.0458	0.736	0.916
43.85	40	1	0.801	0.0490	0.710	0.903
45.61	32	1	0.776	0.0535	0.677	0.888
49.40	28	1	0.748	0.0583	0.642	0.871
52.22	19	1	0.708	0.0672	0.588	0.853

#con la siguiente opción puedo estimar el por sección de tiempo

```
summary(linelistsurv_fit, times = c(6,12,24,36,48,60))
```

```
Call: survfit(formula = surv_object2 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
6	78	2	0.975	0.0175	0.941	1.000
12	75	3	0.938	0.0271	0.886	0.992
24	70	3	0.900	0.0335	0.837	0.968
36	52	2	0.873	0.0375	0.803	0.950
48	30	5	0.776	0.0535	0.677	0.888
60	4	2	0.708	0.0672	0.588	0.853

imprimir el objeto linelistsurv_fit con el tiempo medio de supervivencia y su se.

```
print(linelistsurv_fit, print.rmean = TRUE)
```

```
Call: survfit(formula = surv_object2 ~ 1)
```

```
      n events rmean* se(rmean) median 0.95LCL 0.95UCL
[1,] 80      17 55.5      2.18    NA      NA      NA
* restricted mean with upper limit = 65.5
```

Por ultimo, usted debe realizar la comparación entre grupos con el test de Log rank:

```
# crear grupos para comparación
Grupo1 ← data1
Grupo2 ← data2

#Añade una columna de grupo e integra ambos dataframes:
Grupo1$grupo ← 1
Grupo2$grupo ← 2
datos_comb ← rbind(Grupo1, Grupo2)

resultado ← survdiff(Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
print(resultado)
Call:
survdiff(formula = Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
```

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
grupo=1	80	14	13.5	0.0154	0.0281
grupo=2	80	17	17.5	0.0119	0.0281

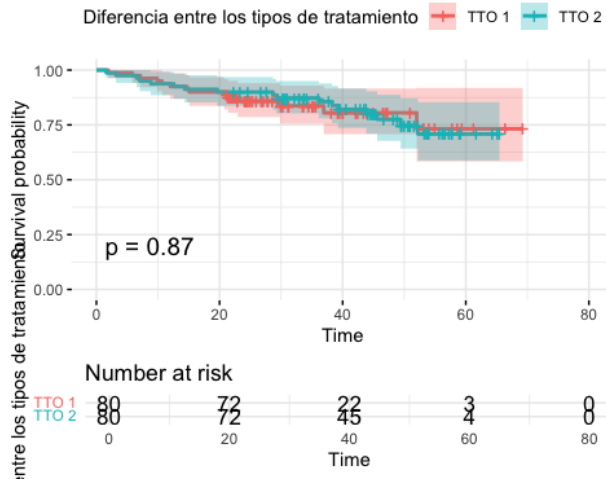
```

Chisq= 0 on 1 degrees of freedom, p= 0.9
#Crear gráfica
surv_obj3 ← Surv(datos_comb$Tiempo, datos_comb$Evento)
fit ← survfit(surv_obj3 ~ grupo, data = datos_comb)

ggsurvplot(fit, data = datos_comb, pval = TRUE,
  conf.int = TRUE, # Intervalos de confianza
  risk.table = TRUE, # Tabla de riesgo
  ggtheme = theme_minimal(), # Tema minimalista
  legend.labs = c("TTO 1", "TTO 2"), # Etiquetas de leyenda
  legend.title = "Diferencia entre los tipos de tratamiento")
# Título de la leyenda

```





- ♦ **Ejercicio 28.2:** para este ejercicio importe a R la base de datos [Sobrevida.xlsx](#) y guárdela en su carpeta de Bases.

La base de datos contiene información de un grupo de pacientes que fueron sometidos a una cirugía para la colocación de prótesis en una de sus rodillas. Se realizó un seguimiento de los pacientes hasta por 20 años. En este estudio el evento es la revisión de la prótesis por complicaciones (independientemente de la naturaleza).

Realice:

- Curvas de Kaplan-Meier
- Realice la prueba de log rank ratio

Resolución [ejercicio 28.2](#)

Recuerde guardar su archivo en su carpeta de bases.

A continuación, crearemos las curvas de Kaplan-Meier para cada grupo:

```
data3 <- read_excel("Bases/Base_Sobrevida_2.xlsx", sheet = "Hoja1")
#Osteoartritis crónica
head(data)
```

```

1 function (..., list = character(), package = NULL, lib.loc = NULL,
2   verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
3 {
4   fileExt ← function(x) {
5     db ← grepl("\\\\.([.]+)\\\\\\.(gz|bz2|xz)$", x)
6     ans ← sub(".*\\\\\\.", "", x)
7   }
8   str(data)
9   function (..., list = character(), package = NULL, lib.loc = NULL,
10    verbose = getOption("verbose"),
11    envir = .GlobalEnv, overwrite = TRUE)
12     data4 ← read_excel("Bases/Base_Sobrevida_2.xlsx", sheet = "Hoja2")
13     #Osteoartritis
14
15     data5 ← read_excel("Bases/Base_Sobrevida_2.xlsx", sheet = "Hoja3")
16     #Osteoartrosis post traumática
17
18     head(data3)
19     # A tibble: 6 × 3
20       Causa          Evento Tiempo
21   <chr>          <dbl>   <dbl>
22 1 Osteoartrosis  crónica     1    19
23 2 Osteoartrosis  crónica     1     4
24 3 Osteoartrosis  crónica     1     7
25 4 Osteoartrosis  crónica     1    19
26 5 Osteoartrosis  crónica     1     5
27 6 Osteoartrosis  crónica     1    19
28
29     head(data4)
30     # A tibble: 6 × 10
31       Causa          Evento Tiempo ...4 ...5 ...6 ...7 ...8 ...9 ...10
32   <chr>          <dbl>   <dbl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <chr>
33 1 Osteoartritis     1    20 NA   NA   NA   NA   NA   NA   NA   <NA>
34 2 Osteoartritis     1     5 NA   NA   NA   NA   NA   NA   NA   <NA>
35 3 Osteoartritis     1    20 NA   NA   NA   NA   NA   NA   NA   <NA>
36 4 Osteoartritis     1    20 NA   NA   NA   NA   NA   NA   NA   <NA>

```

```

5 Osteoartritis      1      20 NA    NA    NA    NA    NA    NA    <NA>
6 Osteoartritis      1       5 NA    NA    NA    NA    NA    NA    <NA>
head(data5)
# A tibble: 6 × 3
  Causa                      Evento Tiempo
  <chr>                      <dbl>  <dbl>
1 Osteoartrosis post traumática      1      4
2 Osteoartrosis post traumática      1     11
3 Osteoartrosis post traumática      1     11
4 Osteoartrosis post traumática      1     11
5 Osteoartrosis post traumática      0     18
6 Osteoartrosis post traumática      0     18
str(data3)
tibble [474 × 3] (S3: tbl_df/tbl/data.frame)
 $ Causa : chr [1:474] "Osteoartrosis crónica" "Osteoartrosis crónica" "Osteoartrosis crónica" "Osteoartrosis crónica" "Osteoartrosis crónica" ...
 $ Evento: num [1:474] 1 1 1 1 1 1 1 1 1 1 ...
 $ Tiempo: num [1:474] 19 4 7 19 5 19 15 7 15 15 ...
str(data4)
tibble [535 × 10] (S3: tbl_df/tbl/data.frame)
 $ Causa : chr [1:535] "Osteoartritis" "Osteoartritis" "Osteoartritis" "Osteoartritis" "Osteoartritis" ...
 $ Evento: num [1:535] 1 1 1 1 1 1 1 1 0 0 ...
 $ Tiempo: num [1:535] 20 5 20 20 20 5 20 7 11 17 ...
 $ ...4 : logi [1:535] NA NA NA NA NA NA ...
 $ ...5 : logi [1:535] NA NA NA NA NA NA ...
 $ ...6 : logi [1:535] NA NA NA NA NA NA ...
 $ ...7 : logi [1:535] NA NA NA NA NA NA ...
 $ ...8 : logi [1:535] NA NA NA NA NA NA ...
 $ ...9 : logi [1:535] NA NA NA NA NA NA ...
 $ ...10: chr [1:535] NA NA NA NA ...
str(data5)
tibble [62 × 3] (S3: tbl_df/tbl/data.frame)
 $ Causa : chr [1:62] "Osteoartrosis post traumática" "Osteoartrosis

```

```

post traumática" "Osteoartritis post traumática" "Osteoartritis post
traumática" ...
$ Evento: num [1:62] 1 1 1 1 0 0 0 0 0 ...
$ Tiempo: num [1:62] 4 11 11 11 18 18 16 5 16 15 ...
# Osteoartritis crónica
# Crear el data frame con las columnas necesarias
data3 <- data.frame(

  Tiempo = data3$Tiempo,
  Evento = data3$Evento,
  Causa = data3$Causa
)

# Crear el objeto de supervivencia
surv_object3 <- Surv(data3$Tiempo, data3$Evento)

# Ajustar el modelo de Kaplan-Meier
fit3 <- survfit(surv_object3 ~ data3$Causa, data = data3)

#Osteoartritis
# Crear el data frame con las columnas necesarias
data4 <- data.frame(
  Tiempo = data4$Tiempo,
  Evento = data4$Evento,
  Causa = data4$Causa
)

# Crear el objeto de supervivencia
surv_object4 <- Surv(data4$Tiempo, data4$Evento)

# Ajustar el modelo de Kaplan-Meier
fit4 <- survfit(surv_object4 ~ data4$Causa, data = data4)

```



```
#Osteoartrosis post traumática
# Crear el data frame con las columnas necesarias
data5 <- data.frame(
  Tiempo = data5$Tiempo,
  Evento = data5$Evento,
  Causa = data5$Causa
)

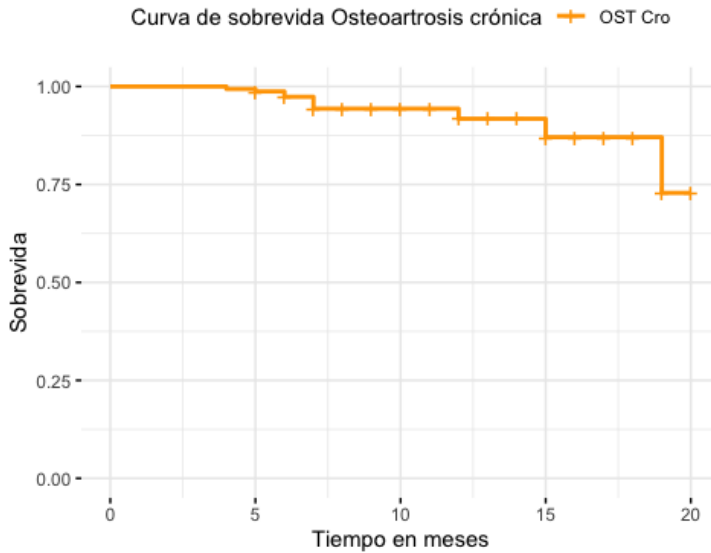
# Crear el objeto de supervivencia

# Ajustar el modelo de Kaplan-Meier
fit5 <- survfit(surv_object5 ~ data5$Causa, data = data5)
```

Una vez preparado todo comenzamos con la creación de las gráficas:

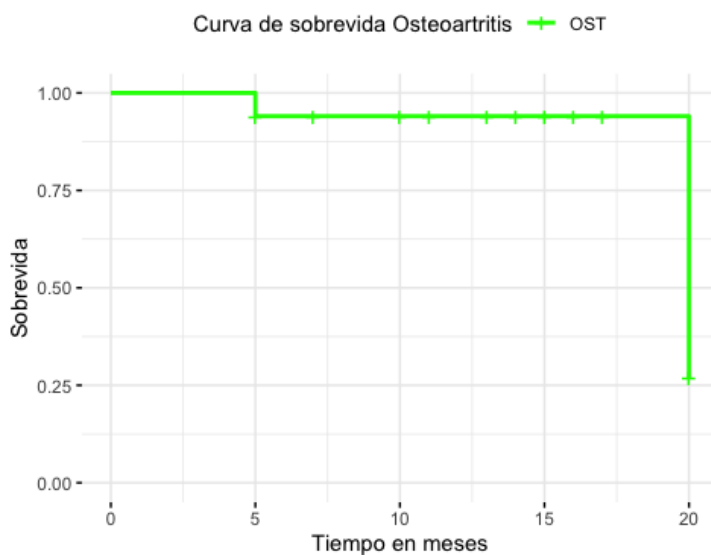
```
plot <- ggsurvplot(
  fit3,
  data = data3,
  pval = FALSE,
  conf.int = FALSE,
  risk.table = FALSE,
  legend.title = "Curva de sobrevida Osteoartrosis crónica",
  legend.labs = c("OST Cro"),
  xlab = "Tiempo en meses",
  ylab = "Sobrevida",
  ggtheme = theme_minimal(),
  palette = c("orange"),
  surv.median.line = "none"
)

print(plot)
```



```
plot <- ggsurvplot(
  fit4,
  data = data4,
  pval = FALSE,
  conf.int = FALSE,
  risk.table = FALSE,
  legend.title = "Curva de supervivencia Osteoartritis",
  legend.labs = c("OST"),
  xlab = "Tiempo en meses",
  ylab = "Supervivencia",
  ggtheme = theme_minimal(),
  palette = c("green"),
  surv.median.line = "none"
)

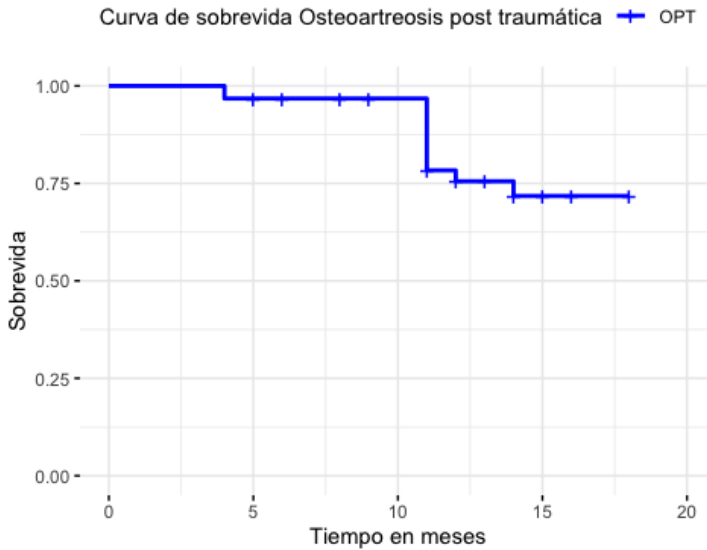
print(plot)
```



```
surv_object5 <- Surv(data5$Tiempo, data5$Evento)

plot <- ggsurvplot(
  fit5,
  data = data5,
  pval = FALSE,
  conf.int = FALSE,
  risk.table = FALSE,
  legend.title = "Curva de sobrevida Osteoartreosis post traumática",
  legend.labs = c("OPT"),
  xlab = "Tiempo en meses",
  ylab = "Sobrevida",
  ggtheme = theme_minimal(),
  palette = c("blue"),
  surv.median.line = "none"
)

print(plot)
```



Proceda a realizar las tablas donde se represente la estadística de las curvas que generó anteriormente:

```
#Tabla de datos del Grupo 3
#Generar tabla de contenido
linelistsurv_fit3 <- survival::survfit(surv_object3 ~ 1)
summary(linelistsurv_fit3)
Call: survfit(formula = surv_object3 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
4	474	3	0.994	0.00364	0.987	1.000
5	471	3	0.987	0.00513	0.977	0.997
6	434	6	0.974	0.00750	0.959	0.989
7	386	12	0.943	0.01126	0.922	0.966
12	220	6	0.918	0.01508	0.889	0.948
15	156	8	0.871	0.02162	0.829	0.914
19	49	8	0.728	0.04940	0.638	0.832

#con la siguiente opción puedo estimar por sección de tiempo

```
summary(linelistsurv_fit3, times = c(6,12,24))
```

```
Call: survfit(formula = surv_object3 ~ 1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
6	434	12	0.974	0.0075	0.959	0.989
12	220	18	0.918	0.0151	0.889	0.948

imprimir el objeto linelistsurv_fit con el tiempo medio de supervivencia y su se.

```
print(linelistsurv_fit3, print.rmean = TRUE)
```

```
Call: survfit(formula = surv_object3 ~ 1)
```

	n events	rmean*	se(rmean)	median	0.95LCL	0.95UCL
[1,]	474	46	18.6	0.191	NA	NA

* restricted mean with upper limit = 20

#Tabla de datos del Grupo 4

#Generar tabla de contenido

```
linelistsurv_fit4 ← survival::survfit(surv_object4 ~ 1)
```

```
summary(linelistsurv_fit4)
```

```
Call: survfit(formula = surv_object4 ~ 1)
```

468 observations deleted due to missingness

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
5	67	4	0.940	0.0289	0.885	0.999
20	14	10	0.269	0.1138	0.117	0.616

#con la siguiente opción puedo estimar el por sección de tiempo

```
summary(linelistsurv_fit4, times = c(6,12,24))
```

```
Call: survfit(formula = surv_object4 ~ 1)
```

468 observations deleted due to missingness

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
6	59	4	0.94	0.0289	0.885	0.999

```

12      37      0      0.94 0.0289      0.885      0.999
# imprimir el objeto linelistsurv_fit con el tiempo medio de super-
vivencia y su se.
print(linelistsurv_fit4, print.rmean = TRUE)
Call: survfit(formula = surv_object4 ~ 1)

468 observations deleted due to missingness
      n events rmean* se(rmean) median 0.95LCL 0.95UCL
[1,] 67      14  19.1      0.434      20      20      NA
      * restricted mean with upper limit = 20

#Tabla de datos del Grupo 5
#Generar tabla de contenido
linelistsurv_fit5 <- survival::survfit(surv_object5 ~ 1)
summary(linelistsurv_fit5)
Call: survfit(formula = surv_object5 ~ 1)

time n.risk n.event survival std.err lower 95% CI upper 95% CI
  4      62      2    0.968  0.0224    0.925    1.000
 11      42      8    0.783  0.0614    0.672    0.913
 12      28      1    0.755  0.0653    0.638    0.895
 14      20      1    0.718  0.0721    0.589    0.874

#con la siguiente opción puedo estimar por sección de tiempo

summary(linelistsurv_fit5, times = c(6,12,24))
Call: survfit(formula = surv_object5 ~ 1)

time n.risk n.event survival std.err lower 95% CI upper 95% CI
  6      56      2    0.968  0.0224    0.925    1.000
 12      28      9    0.755  0.0653    0.638    0.895
# imprimir el objeto linelistsurv_fit con el tiempo medio de super-
vivencia y su se.
print(linelistsurv_fit5, print.rmean = TRUE)
Call: survfit(formula = surv_object5 ~ 1)

```

```

      n events rmean* se(rmean) median 0.95LCL 0.95UCL
[1,] 62      12  15.9      0.523      NA      NA      NA

```

```

* restricted mean with upper limit = 18

```

Por último, debe realizar la comparación entre grupos con el test de log rank:

```

# crear grupos para comparación

```

```

Grupo1 <- data3

```

```

Grupo2 <- data4

```

```

#Añade una columna de grupo e integra ambos data frames:

```

```

Grupo1$grupo <- 1

```

```

Grupo2$grupo <- 2

```

```

datos_comb <- rbind(Grupo1, Grupo2)

```

```

resultado <- survdiff(Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
print(resultado)

```

Call:

```

survdiff(formula = Surv(Tiempo, Evento) ~ grupo, data = datos_comb)

```

```

n=541, 468 observations deleted due to missingness.

```

```

      N Observed Expected (O-E)^2/E (O-E)^2/V
grupo=1 474      46      48    0.0861    0.527
grupo=2  67      14      12    0.3458    0.527

```

```

Chisq= 0.5  on 1 degrees of freedom, p= 0.5

```

```

#Crear gráfica

```

```

surv_obj3 <- Surv(datos_comb$Tiempo, datos_comb$Evento)

```

```

fit <- survfit(surv_obj3 ~ grupo, data = datos_comb)

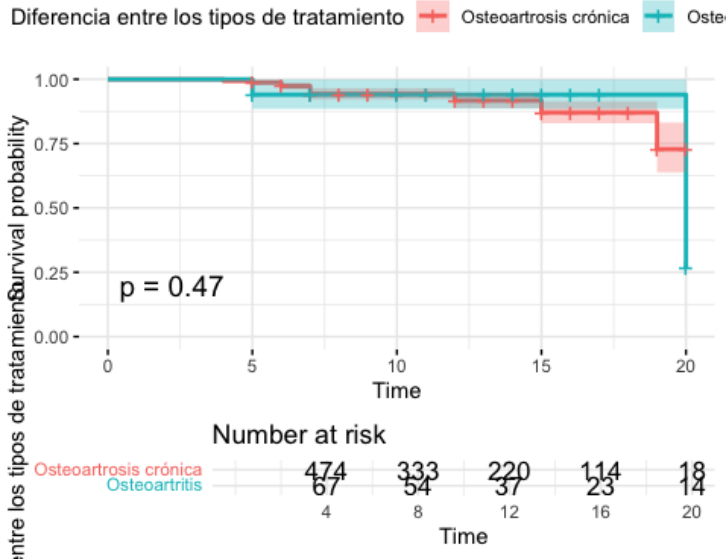
```

```

ggsurvplot(fit, data = datos_comb, pval = TRUE,
            conf.int = TRUE, # Intervalos de confianza
            risk.table = TRUE, # Tabla de riesgo

```

```
ggtheme = theme_minimal(), # Tema minimalista
legend.labs = c("Osteoartrosis crónica", "Osteoartritis"),
# Etiquetas de leyenda
legend.title = "Diferencia entre los tipos de tratamiento")
# Título de la leyenda
```



```
# crear grupos para comparación
Grupo1 ← data3
Grupo2 ← data5

#Añade una columna de grupo e integra ambos dataframes:
Grupo1$grupo ← 1
Grupo2$grupo ← 2
datos_comb ← rbind(Grupo1, Grupo2)

resultado ← survdiff(Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
print(resultado)

Call:
survdiff(formula = Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
```


	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
grupo=1	474	46	52.11	0.716	7.36
grupo=2	62	12	5.89	6.338	7.36

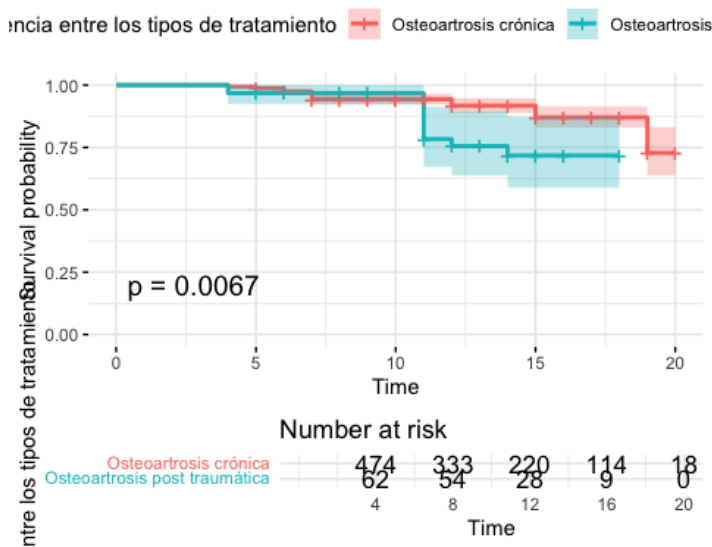
Chisq= 7.4 on 1 degrees of freedom, p= 0.007

#Crear gráfica

```
surv_obj3 <- Surv(datos_comb$Tiempo, datos_comb$Evento)
```

```
fit <- survfit(surv_obj3 ~ grupo, data = datos_comb)
```

```
ggsurvplot(fit, data = datos_comb, pval = TRUE,
  conf.int = TRUE, # Intervalos de confianza
  risk.table = TRUE, # Tabla de riesgo
  ggtheme = theme_minimal(), # Tema minimalista
  legend.labs = c("Osteoartritis crónica", "Osteoartritis
post traumática"), # Etiquetas de leyenda
  legend.title = "Diferencia entre los tipos de tratamiento")
# Título de la leyenda
```



```
# crear grupos para comparación
Grupo1 ← data4
Grupo2 ← data5

#Añade una columna de grupo e integra ambos dataframes:
Grupo1$grupo ← 1
Grupo2$grupo ← 2
datos_comb ← rbind(Grupo1, Grupo2)

resultado ← survdiff(Surv(Tiempo, Evento) ~ grupo, data = datos_comb)
print(resultado)
Call:
survdiff(formula = Surv(Tiempo, Evento) ~ grupo, data = datos_comb)

n=129, 468 observations deleted due to missingness.
```

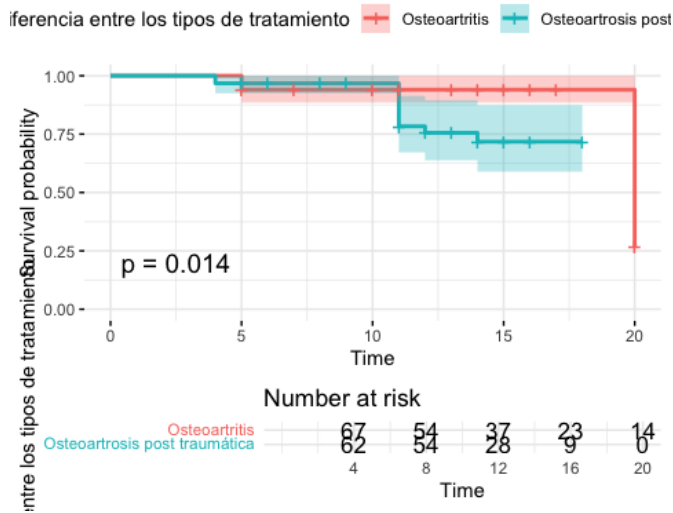
	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
grupo=1	67	14	18.79	1.22	6.07
grupo=2	62	12	7.21	3.18	6.07

```

Chisq= 6.1 on 1 degrees of freedom, p= 0.01
#Crear gráfica
surv_obj3 ← Surv(datos_comb$Tiempo, datos_comb$Evento)
fit ← survfit(surv_obj3 ~ grupo, data = datos_comb)

ggsurvplot(fit, data = datos_comb, pval = TRUE,
            conf.int = TRUE, # Intervalos de confianza
            risk.table = TRUE, # Tabla de riesgo
            ggtheme = theme_minimal(), # Tema minimalista
            legend.labs = c("Osteoartritis", "Osteoartrosis post trau-
mática"), # Etiquetas de leyenda
            legend.title = "Diferencia entre los tipos de tratamiento")
# Título de la leyenda

```



Referencias

- Bender, A. y Berhane, K. (2020). *Survival Analysis in Public Health Research*. Springer.
- Blanche, P., Dartigues, J. F. y Jacqmin-Gadda, H. (2021). Estimating and comparing time-dependent areas under receiver operating characteristic curves for censored event times with competing risks. *Statistics in Medicine*, 40(15), 3564-3576.
- Bradburn, M. J., Clark, T. G., Love, S. B. y Altman, D. G. (2017). Survival analysis part II: multivariate data analysis—an introduction to concepts and methods. *British Journal of Cancer*, 89(3), 431-436.
- Cox, C., Chu, H., Schneider, M. F. y Muñoz, A. (2015). Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. *Statistics in Medicine*, 26(23), 4352-4374.
- Create Elegant Data Visualisations Using the Grammar of Graphics (s. f.). <https://ggplot2.tidyverse.org/>
- Crowther, M. J. y Lambert, P. C. (2016). Parametric multistate survival models: Flexible modelling allowing transition-specific distributions with application to estimating clinically useful measures of effect differences. *Statistics in Medicine*, 36(29), 4719-4742.

- Crowther, M. J. y Lambert, P. C. (2021). Simulating complex survival data. *Statistics in Medicine*, 40(15), 3413-3435.
- Gerds, T. A., Kattan, M. W., Schumacher, M. y Yu, C. (2018). Estimating a time-dependent AUC for survival prediction models with censored data. *Machine Learning*, 102(2), 231-247.
- Gillaspy, S. R. y Campbell, S. L. (2016). *Introduction to survival analysis: Censored data and truncation*. SAGE Publications.
- Hoh, B. P. y Peto, T. (2021). Statistical models for studying survival and disease progression in genetic association studies. *Journal of Genetics and Genomics*, 48(2), 123-137.
- Hougaard, P. y Cook, R. J. (2021). Modelling multivariate survival data. *Lifetime Data Analysis*, 27(2), 197-227.
- Kleinbaum, D. G. y Klein, M. (2021). *Survival Analysis: A Self-Learning Text*. Springer.
- Lambert, P. C. y Royston, P. (2017). Further development of flexible parametric models for survival analysis. *Stata Journal*, 9(2), 265-290.
- Li, Y. y Taylor, J. M. (2019). Joint modeling of longitudinal and survival data: Current methods and issues. *Report on Progress in Physics*, 82(2), 066801.
- O'Quigley, J., Xu, R. y Stare, J. (2017). Explained randomness in survival analysis. *Statistics in Medicine*, 24(1), 479-489.
- Royston, P. y Lambert, P. C. (2016). *Flexible parametric survival analysis using Stata: Beyond the Cox model*. Stata Press.
- Therneau, T. M. (2015). A package for survival analysis in S. R package version 2.38.
- Uno, H., Claggett, B., Tian, L., Inoue, E., Gallo, P., Miyata, T. y Wei, L. J. (2016). Moving beyond the hazard ratio in quantifying the between-group difference in survival analysis. *Journal of Clinical Oncology*, 32(22), 2380-2385.
- Upton, G. y Cook, I. (2014). *A Dictionary of Statistics* (3ª ed.). Oxford University Press. <https://www.oxfordreference.com/display/10.1093/acref/9780199679188.001.0001/acref-9780199679188>
- Wickham, H., Navarro, D. y Perderse, T. L. (2016). *ggplot2: Elegant Graphics for Data Analysis* (3e). <https://ggplot2-book.org/>
- Zhang, M. J. y Klein, J. P. (2020). A new approach for estimating the restricted mean survival time for competing risks data. *Lifetime Data Analysis*, 26(4), 598-614.

Semblanzas

Edsaúl Emilio Pérez Guerrero

Licenciado en Químico Farmacéutico Biólogo, maestro en Farmacología y doctor en Farmacología por la Universidad de Guadalajara. Profesor titular de las materias de Bioestadística y Bioestadística Avanzada en diversos posgrados de la Universidad de Guadalajara. Su línea de investigación se centra en la evaluación de biomarcadores asociados a falla terapéutica en enfermedades autoinmunes y cáncer. Apasionado de las ciencias de datos, el código abierto y de la programación en R.

Alberto Beltrán Ramírez

Médico general con doctorado en Farmacología y experiencia postdoctoral en la Universidad de Guadalajara. Ha colaborado en la creación de programas académicos y tiene amplio conocimiento en bioestadística aplicada a las ciencias de la salud.

Miryam Rosario Guillén Medina

Licenciada en Químico Farmacéutico Biólogo y estudiante del doctorado en Farmacología de la Universidad de Guadalajara. Su investigación se enfoca en el papel de las proteínas transportadoras de membrana en un ambiente proinflamatorio.

Daniel Osmar Suárez Rico

Especialista en Farmacología, con amplia experiencia en docencia e investigación en áreas científicas. Sus investigaciones se centran en el reposicionamiento de fármacos, microbiota y enfermedades metabólicas. Actualmente, desempeña roles como profesor y coordinador en distintas instituciones académicas y hospitalarias.

Ana Gabriela Flores Vargas

Licenciada en Biología, estudiante del doctorado en Farmacología de la Universidad de Guadalajara. Investiga la asociación de las células supresoras derivadas mieloides con el tratamiento farmacológico en pacientes con artritis reumatoide.

Jesús Jonathan García Galindo

Doctor en Farmacología con experiencia en docencia e investigación. Su línea de investigación se centra en enfermedades metabólicas e inflamación. Actualmente se desempeña como profesor y jefe de enseñanza en distintas instituciones y hospitales.

Gerencia editorial

Sol Ortega Ruelas

Cuidado de la edición

Mariana Hernández Alvarado

Diseño de portada

Pablo Ulises Ontiveros Pimienta

Maquetación

Maritzel Aguayo Robles

Bioestadística en R para ciencias de la salud. Guía práctica

se terminó de editar en noviembre de 2024

en Guadalajara, Jalisco, México.

Para su composición se utilizaron las tipografías **Fira Sans** y **Andada Pro**.

Diseñada para estudiantes e investigadores sin experiencia previa en programación, la obra cubre desde fundamentos básicos hasta técnicas avanzadas, como regresión logística y análisis de componentes principales.

Cada capítulo incluye ejemplos prácticos y ejercicios que facilitan el aprendizaje, con secciones dedicadas al lenguaje R. Esta guía busca cerrar la brecha entre el análisis estadístico y la programación, proporcionando herramientas útiles para el análisis de datos en el ámbito de la salud.



ALFABÉTICA