INTRODUCCIÓN A LA CIENCIA DE DATOS

Estadística e Inteligencia Artificial con

ALEJANDRO RODRÍGUEZ-COLLADO

Alejandro Rodríguez-Collado

ESTADÍSTICA E INTELIGENCIA ARTIFICIAL CON R

EDICIONES Universidad Valladolid

Notas aclaratorias:

Este libro hace uso de los programas informáticos R y R Studio, ciertos paquetes de R (R Commader, R Weka, etc.), las fuentes IBM Plex Sans y Font Awesome, así como diversos conjuntos de datos (Countries of the World, Indian Pima Diabetes, etc.).

Los programas, paquetes, fuentes y conjuntos de datos son propiedad exclusiva de sus respectivos autores, y cualquier referencia a estos no implica la afirmación de propiedad por parte del autor de este libro. Así mismo, sus autores han sido debidamente citados en la sección correspondiente a la bibliografía.

Asimismo, aparecen en el libro logos de diversas empresas (Amazon, Apple, Filmin, Facebook, HBO, Instagram, Netflix, Twitter, …). El uso de estos es completamente ilustrativo, y no implica afiliación o propiedad de los mismos.

Primera edición, mayo 2024.

Contacto del autor: alejandrorodriguezcollado@gmail.com

Diseño: Alejandro Rodríguez Collado

ISBN: 978-84-128577-1-9

Índice

Índice

Índice

Índice

Índice

Prefacio

Este libro permitirá al lector adquirir ciertos conocimientos básicos de estadística e inteligencia artificial. Aprenderá a aplicar diversas técnicas y modelos con el programa R. La comprensión del texto no requiere de conocimientos previos, más allá del anhelo por aprender, así como unas nociones básicas de informática y estadística.

Se alternarán conceptos básicos de la estadística e inteligencia artificial con ejercicios prácticos, en los que se afrontarán problemas ilustrativos basados en datos reales. En un principio, se trabajará con el interfaz gráfico de R Commander y, posteriormente, se formará en el uso de R como lenguaje de programación.

Prefacio

El texto está escrito con un enfoque didáctico, sencillo y liviano. Está acompañado de notas, que facilitan su lectura y compresión, siguiendo el formato descrito a continuación:

Nota importante
Necesaria para seguir el texto.

Nota informativa
Tiene información complementaria.

Salidas de R
R lo devuelve al ejecutar comandos.

Por otro lado, una vez se comience a programar, el código de R aparecerá en bloques como este:

```
miVector<- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(miVector)
```

Prefacio

A lo largo de los diferentes ejercicios prácticos, se emplearán varios conjuntos de datos. Algunos de los mismos pertenecen a la distribución básica de R o a alguno de sus paquetes instalados por defecto. El resto deberán ser descargados del siguiente enlace:

https://drive.google.com/file/d/1z85v-Je0R3sC-6m-_gDSbP8Yi0Lep7sb/view?usp=drive_link

TEMA 1
Introducción

Estadística e IA con 

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 1: Introducción

Índice

1. El petróleo digital: Datos.
2. Conjuntos de datos.
3. El lenguaje de programación R
   - IDE RStudio.
   - Paquete R Commander.

Tema 1: Introducción

❶ El petróleo digital: Datos.

Según algunos autores, son el **activo más importante** de la economía actual. Grandes empresas tecnológicas dominan el mercado gracias a todos los datos que producimos.

El petróleo digital: Datos

COMO CIUDADANOS...

... vemos como empresas usan nuestros datos con todo tipo de propósitos.

¿Qué serie veo esta noche?

¿Me concederían un préstamo?

¿Cómo se trataría una enfermedad?

El petróleo digital: Datos

LAS EMPRESAS E INSTITUCIONES...

... buscan sacar partido a los registros de datos que tienen.

Los datos dan información acerca del pasado, el presente y... **El FUTURO**.

OBJETIVO

La toma de decisiones moderna se basa en pronósticos y análisis numéricos semiautomáticos (estadística, inteligencia artificial,...) que aprovechen el gran volumen de datos disponible (Big Data).

El petróleo digital: Datos

¿Cómo sacar provecho a los Datos?

1. Recopilación de (¿mucha?) **información en bases de datos.**	Por ejemplo, los datos de compra de una tienda online.
2. Limpieza y transformación de los datos.	¿Todos los usuarios han comprado algo? ¿Me interesa que las valoraciones de los productos sean números decimales?
3. Estudio y exploración de los datos.	¿Qué productos se venden más en función de los diferentes tipos de clientes que tengo?
4. Desarrollo de modelos para explicar los datos.	¿Puedo predecir las necesidades pasadas de mis clientes?
5. Explotación de los modelos.	Puedo predecir las **necesidades futuras** de mis clientes.

Tema 1: Introducción

❷ Conjuntos de datos

Ya hemos visto que los datos son útiles, pero ¿Cómo debo organizar los datos para sacarlos provecho? La mayoría de las técnicas asumen que los datos están en **forma tabular**.

Conjuntos de datos

Concepto base en el análisis de datos (estadística, IA,...). Asumiremos que estos se disponen en **columnas**, que representan las **variables**, y **filas**, que son las **observaciones**.

- **Variable**: ¿Qué medidas o características medimos de cada observación?
- **Observación**: ¿Cuál es el elemento o unidad al que tomamos las diferentes variables?

VARIABLE

País	Continente	Región	Población	Superficie	Índice desarrollo humano
Afganistán	2	Oriente Medio	31056997	652 864,2	Bajo
Albania	4	Europa del Este	3581655	28748,4	Alto
Angola	1	África Subsahariano	12127071	1246709,9	Medio
Argentina	3	Caribe	39921833	2766889,8	Muy alto

OBSERVACIÓN

Tabla. Ejemplo de conjunto de datos de países en el mundo.

Conjuntos de datos

Tipos de variables

- **Variables numéricas o cuantitativas**: sus valores son números reales.
  - **Discretas**: toman un conjunto de valores aislado, habitualmente números naturales.
    Por ejemplo: número de clientes de una tienda o la población de un municipio.
  - **Continuas**: toman valores en un intervalo de la recta real de forma continua.
    Por ejemplo: distancia entre dos municipios o la duración de una reacción química.

			Numérica discreta	Numérica continua	
País	**Continente**	**Región**	**Población**	**Superficie**	**Índice desarrollo humano**
Afganistán	2	Oriente Medio	31056997	652 864,2	Bajo
Albania	4	Europa del Este	3581655	28748,4	Alto
Angola	1	África Subsahariano	12127071	1246709,9	Medio
Argentina	3	Caribe	39921833	2766889,8	Muy alto

Conjuntos de datos

TIPOS DE VARIABLES

- **Variables categóricas o cualitativas**: expresan la posesión de características.
  - **Nominales**: las categorías no tienen implícito un orden. A veces, identifican a las filas. Por ejemplo, el país de origen o el sexo de una persona.
  - **Ordinales**: categorías con un orden. Por ejemplo, gravedad lesión entre {leve, grave}.

País	Continente	Región	Población	Superficie	Índice desarrollo humano
Afganistán	2	Oriente Medio	31056997	652 864,2	Bajo
Albania	4	Europa del Este	3581655	28748,4	Alto
Angola	1	África Subsahariano	12127071	1246709,9	Medio
Argentina	3	Caribe	39921833	2766889,8	Muy alto

Conjuntos de datos

Codificación de Variables

- No hay que confundir el tipo de la variable con su **codificación**, especialmente en las categóricas. Algunas codificaciones mejoran los resultados de ciertos procedimientos.
- Veamos algunas formas de codificar una variable categórica nominal. Tomaremos como ejemplo la variable "Continente", siendo cada país una observación (fila).

Codificación *natural*

España	Europa
USA	América
China	Asia
Afganistán	Asia
...	...

Codificación numérica 1

España	4
USA	3
China	2
Afganistán	2
...	...

Codificación numérica 2

	África	Asia	América	Europa	Oceanía
España	0	0	0	1	0
USA	0	0	1	0	0
China	0	1	0	0	0
...					

Tema 1: Introducción

❸ El lenguaje de programación R

Lenguaje de programación interpretado de libre distribución.

Creado en 1992 por un grupo de investigadores **ajenos a la programación**, recibe anualmente 4 grandes actualizaciones.

El lenguaje de programación R

ESTRUCTURA

R Base (distribución básica)

Funcionalidad base: tipos de datos (básicos), cálculos (básicos), gráficos (básicos), lectura y escritura de ficheros (básicos).

Se amplia mediante **Paquetes** (*library*)

- Lectura-escritura ficheros. `xlsx, arrow,…`
- Cálculos, modelos complejos. `stats, rweka,…`
- Interfaz a otro paquete. `Rcmdr, esquisse,…`
- Nuevos tipos de dato. `terra, lubridate,…`
- Técnicas de un ámbito. `psych, bioconductor,…`

FILOSOFÍA

"Un mismo objetivo se puede lograr de muchas formas"

El lenguaje de programación R

¿Por qué R?

Lenguaje de **programación interpretado de distribución libre** enfocado en el análisis de datos.

¿Qué le diferencia respecto a otras herramientas para el análisis de datos?

- **Gratuito**: no hay que pagar ningún tipo de licencia. Esto, a su vez, atrae a más usuarios.
- **Flexible**: gracias a ser un lenguaje de programación, el nº. de procedimientos aumenta diariamente. Permite implementar procedimientos propios y no limita los procedimientos ofrecidos.
- **Dinámico**: La comunidad que usa R es enorme: los paquetes se encuentran en un proceso continuo de actualización, existen muchos foros y páginas de discusión sobre el lenguaje,...

En este curso, trabajaremos inicialmente con R Commander, que nos permitirá usar R como una herramienta con ventanas, y, posteriormente, usaremos R como lenguaje de programación.

El lenguaje de programación R

¿Por qué R?

Lenguaje de programación específico más empleado en ciencia de datos (procesamiento, análisis y visualización de datos, IA, ...). Cuenta con la colección de modelos más amplia y es el único cuyo propósito es el análisis de datos (otros lenguajes, como Python, son multipropósito).

> *"Plantéate un objetivo en tu proyecto que realmente te importe. Con R, podrás lograrlo sin que realmente tengas que saber programar"*
>
> En: https://towardsdatascience.com/the-8-most-popular-coding-languages-of-2021-b3dccb004635#1a6a

¿Y Excel? Herramienta ofimática para bases de datos pequeñas-medianas. Es problemático con cálculos/datos complejos, y no escala correctamente con grandes volúmenes de datos.

Instalación: Lenguaje de programación R

La instalación varía **en función de vuestro sistema operativo**.

Windows

1) Entrad en https://cran.r-project.org/bin/windows/base/rpatched.html
2) Hacemos click en "Download R Patched build for Windows". El fichero instalará R (si saltase el antivirus/ firewall, ignorad la advertencia: el fichero es seguro). Ante la duda, dejamos la configuración por defecto.

MacOS

1) Entrad en https://cran.r -project.org/
2) Hacemos click en "Download R for macOS" y descargamos el último fichero PKG disponible. Éste instalará R. Ante la duda, lo mejor es dejar la configuración elegida por defecto.

Unix

1) Entrad en https://cran.r-project.org/ y hacemos click en "Download R for Linux".
2) Seguimos las instrucciones correspondientes, según tengamos Debian, Fedora o Ubuntu.

Instalación: IDE R Studio

Entorno de desarrollo integrado de R que facilitan el uso del lenguaje de programación.

Instalación

1) Entrad en https://www.rstudio.com/products/rstudio/download/
2) Descargamos el fichero correspondiente a nuestro sistema operativo.

Una vez finalizada la instalación, podemos acceder a RStudio desde el botón de inicio del sistema operativo. También se creará un atajo en el escritorio, si habilitamos la opción.

- Si nos pide alguna configuración adicional al abrirlo, aceptamos las opciones por defecto.
- Su interfaz se compone de cuatro paneles principales:

Scripts (oculto por defecto; Tema 3)	**Entorno de variables** (Tema 3)
Consola de comandos	**Multifunción** (gráficos, paquetes, ayuda,…)

Instalación: Paquete R Commander

Nos permite usar R mediante una interfaz por ventanas y con menús desplegables.

Instalación desde RStudio

1) En el panel inferior derecho (Multifunción), hacemos click en la pestaña Packages.
2) Después, debemos pulsar el botón “Install”.
3) Introducimos en el epígrage “Packages (separate multiple with space or comma)”: Rcmdr (**cuidado, respetad las letras en mayúsculas y minúsculas**).

Para abrirlo, debemos introducir en el panel izquierdo inferior (consola de comandos):

```
library(Rcmdr)
```

> Puede saltar, en segundo plano, un mensaje indicando que hay que instalar más paquetes. Pulsamos el botón “Sí”, y después el botón “OK”.

TEMA 2
R Commander – Manejo de datos

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 2: R Commander – Manejo de datos

Índice

Tema 2: R Commander – Manejo de datos

❶ Introducción de datos

Importar los datos es el primer paso para analizar datos en R. Con la ayuda de R Commander, podemos cargar los datos de tres orígenes diferentes: introducción manual, uso de datos de prueba, o **importación desde un fichero externo**.

Introducción de datos: Manual

- **Introducción manual**
- Datos de prueba
- Carga de fichero

Para introducir manualmente los datos en R Commander, seleccionamos el siguiente comando:

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
Nuevo conjunto de datos...						
Cargar conjunto de datos...						
Importar datos						
Conjunto de datos en paquetes						
Conjunto de datos en activo						
Modificar variables del conjunto de datos en activo						

Introducción de datos: Manual

Vamos a introducir manualmente los valores del conjunto de datos “Coches”:

	Peso (toneladas)	Consumo (l/100km)	Marca
Coche 1	1.5	11	Renault
Coche 2	2	14	Renault
Coche 3	2.5	23	Ferrari
Coche 4	1	8	Renault

En R (por defecto), los decimales se indican con un “.”

Con el botón de “Visualizar conjunto de datos”, podemos comprobar los datos introducidos.

Para modificarlos, usamos “Editar conjunto de datos”.

Introducción de datos: Manual

Comencemos a trabajar con alguna cuestión sencilla: **¿Cuál es el consumo medio?**

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes ▶		**Conjunto de datos activo**			
	Tablas de contingencia		Resúmenes Numéricos...			
	Medias		Distribución de frecuencias...			
	(....)		(...)			

El resultado del análisis aparece en la consola de R Studio:

```
Peso                Consumo             Marca
 Min.   :1.000       Min.   : 8.00       Length:4
 1st Qu.:1.375       1st Qu.:10.25       Class :charac.
 Median :1.750       Median :12.50       Mode :charac.
 Mean   :1.750       Mean   :14.00
 3rd Qu.:2.125       3rd Qu.:16.25
 Max.   :2.500       Max.   :23.00
```

Introducción de datos: Manual

¿Consumen los coches de Renault menos que los de Ferrari?

→ Estudio de una variable numérica en función de las categorías de una variable discreta.

R Commander						
Datos	**Estadísticos** ▶	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes ▶		Conjunto de datos activo			
	Tablas de contingencia		**Resúmenes Numéricos...**			
	Medias		Distribución de frecuencias...			
	(....)		(...)			

Seleccionamos la variable `Consumo` y, en "Resumir por grupos", `Marca`.

La media de Ferrari es 23, mientras que la de Renault es 11. Por lo tanto, en esta muestra, los coches de Renault consumen menos que los de Ferrari.

Algunos comandos básicos en R

Aunque R Commander nos facilita la realización de diferentes operaciones, es relevante recordar que, en el fondo, se están ejecutando comandos del lenguaje de programación R.

Para empezar, visualizaremos el conjunto de datos en la interfaz de R Studio  .
Haz click en “Coches” en la pestaña Environment del panel superior derecho.

> Si no aparece en Environment “Coches”, haced click en History y volved a Environment.

> ¿Sigue sin aparecer? Pulsa Enter con el cursor en la consola de R Studio (panel inferior izquierdo).

Vamos a ejecutar algunas órdenes sencillas para familiarizarnos con el lenguaje de R.

Algunos comandos básicos en R

En la consola, podemos visualizar el contenido del conjunto de datos con:

```
Coches
```

El operador `$` permite acceder a cada una de las variables del conjunto de datos:

```
Coches$Peso
```

```
[1] 1.5  2  2.5  1
```

Por último, calcularemos la media y varianza del consumo:

```
mean(Coches$Peso)
var(Coches$Peso)
```

```
[1] 14
[1] 42
```

Vistas estas nociones básicas de R, podemos continuar trabajando con R Commander.

Introducción de datos: Datos de prueba

- Introducción manual
- **Datos de prueba**
- Carga de fichero

El lenguaje de programación R y sus paquetes, módulos que amplían su funcionalidad, incluyen conjuntos de datos para usarse de prueba con sus diferentes procedimientos.

En R Commander, se pueden cargar los datos y leer su documentación con los comandos:

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
Nuevo conjunto de datos...						
Cargar conjunto de datos...						
Importar datos						
Conjunto de datos en paquetes ▶			**Lista de conjunto de datos en paquetes**			
Conjunto de datos en activo			**Leer conjunto de datos de paquete adjunto...**			
Modificar variables del conjunto de datos en activo						

Introducción de datos: Carga de fichero

- Introducción manual
- Datos de prueba
- **Carga de fichero**

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
Nuevo conjunto de datos...						
Cargar conjunto de datos...						
Importar datos ▶			Desde archivo de texto, portapapeles o URL...			
Conjunto de datos en paquetes			Desde datos SPSS...			
Conjunto de datos en activo			Desde un archivo SAS exportado...			
Modificar variables del conjunto de datos en activo			Desde un archivo SAS b7dat...			
			Desde datos Minitab...			
			Desde datos STATA			
			Desde un archivo de Excel...			

Conjunto de datos: Países del Mundo

Disponible en el enlace provisto en el prefacio. Contine las siguientes variables de los países:

1. Continente al que pertenecen.
2. Región dentro del continente.
3. Población.
4. Área en m^2.
5. Migración neta.
6. PIB per cápita.
7. Teléfonos por cada 1000 personas.

País	Continente	Región	Población	Superficie	(...)
Afganistán	2	Oriente Medio	31056997	652 864,2	(...)
Albania	4	Europa del Este	3581655	28748,4	(...)
Angola	1	África Subsahariano	12127071	1246709,9	(...)
(...)	(...)	(...)	(...)	(...)	

Introducción de datos: Carga de fichero

Para cargar un fichero Excel (como “Países del Mundo”) con R Commander, usamos:

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
Nuevo conjunto de datos...						
Cargar conjunto de datos...						
Importar datos ▶			Desde archivo de texto, portapapeles o URL...			
Conjunto de datos en paquetes			Desde datos SPSS...			
Conjunto de datos en activo			(...)			
Modificar variables del conjunto de datos en activo			**Desde un archivo de Excel...**			

Llamaremos al conjunto de datos “Mundo”. Mantendremos las opciones marcadas por defecto.

Podemos seleccionar los datos del análisis con el botón junto a “Conjunto de datos:”.

Tema 2: R Commander – Manejo de datos

❷ Transformaciones de datos

En el análisis de datos, son necesarias **las transformaciones de observaciones** (filas) **y variables** (columnas) para trabajar con subconjuntos de filas, cambiar de escala o discretizar una variable, calcular una variable nueva a partir de otras, ...

Transformaciones

Transformaciones de observaciones (filas): modificar las filas de un conjunto, por ejemplo, filtrando las observaciones con algún criterio (normalmente, basado en una variable).

- Dividir el conjunto de datos para realizar un estudio detallado por continente, región,...
- Aunque no lo veremos, también se incluyen operaciones de agrupación de filas.

País	Continente	Región	Población	Superficie	(...)
Afganistán	2	Oriente Medio	31056997	652 864,2	(...)
Albania	4	Europa del Este	3581655	28748,4	(...)
(...)	(...)	(...)	(...)	(...)	

Transformaciones

Transformaciones de variables (columnas): obtención de medidas derivadas de los datos iniciales. En estas, es relevante tener en cuenta el tipo (numérica / categórica) de las variables.

- Densidad poblacional = Población / Superficie.
- Discretizar Superficie: países {pequeños, grandes}.
- ¿Qué países perdieron población por emigración?
- Población: rango muy amplio → ¿log(población)?

País	Continente	Región	Población	Superficie	(...)
Afganistán	2	Oriente Medio	31056997	652 864,2	(...)
Albania	4	Europa del Este	3581655	28748,4	(...)
(...)	(...)	(...)	(...)	(...)	

Transformaciones

En R Commander, la mayoría de las operaciones para transformar variables se sitúan en:

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
Nuevo conjunto de datos...						
(...)						
Modificar variables del conjunto de datos activo ▶		Recodificar variables...				
		Calcular una nueva variable...				
		Añadir nº de observaciones al conjunto de datos				
		Tipificar variables...				
		Convertir variable numérica en factor...				
		Convertir variables de caracteres en factores...				
		Segmentar variable numérica				
		(...)				
		Renombrar variables...				
		Eliminar variables del conjunto de datos...				

Transformaciones: Renombrar variables

No es como tal una transformación: sirve para cambiar la denominación de las variables.

- En diversos procedimientos, será necesario seleccionar simultáneamente varios elementos. Para ello, mantendremos pulsado la tecla Control y haremos click en cada uno de los elementos.

Vamos a **traducir los nombres de las variables a español**.

> Primera ventana: seleccionamos todas las variables del conjunto de datos.

- Area → Area
- Country → Pais
- NetMigration → Migracion
- Population → Poblacion
- Continent → Continente
- GDP → PIB
- Phones → Telefonos
- Region → Region

Cuidado: evitad en los nombres de variables espacios, acentos, eñes, y/o ciertos símbolos.

Transformaciones: Recodificar variables

Sirve para sustituir uno o varios valores concretos por otros en una variable.

- Transformación: variable categórica ⟶ categórica o numérica discreta ⟶ numérica discreta.

En continente, vamos a cambiar los valores *Near East* (Oriente Próximo) y *Asia -Ext. Near East-* (fuera de Oriente Próximo) para **agrupar todos los países de Asia** bajo un mismo valor.

Seleccionamos la variable `Region`.

En directrices de recodificación, introducimos:

```
"ASIA (EX. NEAR EAST)" = "ASIA"
"NEAR EAST" = "ASIA"
```

Nota: En la parte derecha de la ventana, podemos modificar el nombre de la variable.

Transformaciones: Segmentación

Transformación valores numéricos en categorías. También denominada discretización.

- Transformación: variable numérica ⟶ categórica.

Por ejemplo, puede resultar interesante estudiar la **población de los países** en forma de **categorías**, es decir, población {muy baja, baja, media, alta, muy alta}.

- Nombre de la nueva variable: `PoblacionCategoria`
- Número de clase: "5"

  NOTA: A veces, no es posible discretizar una variable en el nº de segmentos especificado.

Existen muchos criterios diferentes para segmentar una variable numérica.

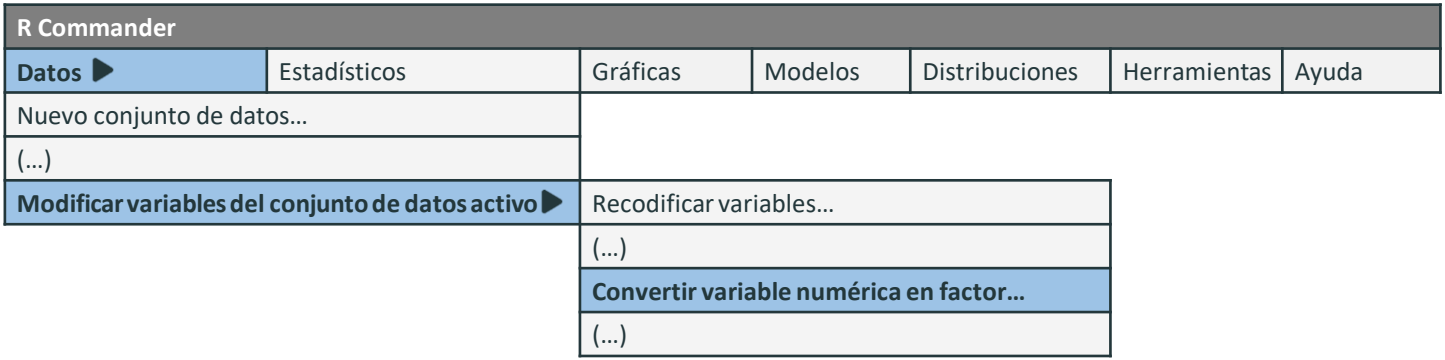
Transformaciones: Creación de factores

Sirve para cambiar la codificación numérica de una variable categórica para usar etiquetas.

- Transformación: variable categórica ⟶ categórica.

Recodifiquemos la variable continente: {1 → Africa, 2 → Asia, 3 → America, 4 → Europa, 5 → Oceania}.

En este caso, como es información que almacena como texto, los nombres podrían incluir acentos.



Transformaciones: Creación de factores

Para verificar la transformación, haremos un estudio básico de esta variable categórica:

¿Cuál es la frecuencia absoluta (n.º países) y relativa (% países) de países por continente?

En la ventana de R Studio, podemos ver que Oceanía es el continente con menos países, con algo menos del 10 % del total, mientras que África tiene el mayor número de países con 57.

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes ▶		Conjunto de datos activo			
	Tablas de contingencia		Resúmenes Numéricos...			
	Medias		**Distribución de frecuencias...**			
	Proporciones		Número de observaciones ausentes			
	(...)		(...)			

Transformaciones: Cálculo de variables

Operación con la que se definen nuevas variables (columnas) a partir de otras ya presentes en los datos. Por ejemplo, podemos calcular la **densidad poblacional** de los países:

$$\text{Densidad} = \frac{\text{Poblacion}}{\text{Area}}$$

- Nombre de la nueva variable: `DensidadPoblacional`
- Expresión a calcular: `Población / Area`

NOTA: El selector en la parte izquierda permite añadir variables de forma sencilla a la expresión.

Transformaciones: Cálculo de variables

Las variables calculadas no tienen por qué ser estrictamente numéricas. Puede interesarnos conocer si un país incrementa o disminuye su población debido a su **flujo inmigratorio**:

- Nombre de la nueva variable: `RecibeMigrantes`
- Expresión a calcular: `Migracion > 0`

La variable creada es categórica binaria (toma dos valores: 0 Falso, 1 Verdadero). En el mundo de la programación, este tipo de variables se denominan lógicas o booleans.

Operadores de comparación en R

<	>	==	<=	>=	!=
Menor que	Mayor que	Igual	Menor o igual	Mayor o igual	Desigual

Transformaciones: Cálculo de variables

Entre los valores de la nueva variable, veremos múltiples **valores faltantes** (*NA - Not Available*). En esos casos, no se ha podido hacer el cálculo al faltar el valor de migración.

¿Cuántos países reciben migración según los datos que tenemos?

Igual que hicimos para estudiar el número de países por continente, usaremos el comando *Estadísticos / Resúmenes / Distribución de frecuencias*.

```
counts:
RecibeMigrantes
   FALSE   TRUE
     154     70
```

```
percentages:
RecibeMigrantes
   FALSE   TRUE
   68.75  31.25
```

Concatenación de comparaciones en R

|	&
Disyunción (... o ...)	Adición (... y ...)

Transformaciones: Cálculo de variables

¿Qué países asiáticos reciben migración y tienen un PIB por encima de la media mundial?

Nombre de la variable: `CondicionCompleja`. Expresión a calcular:

```
RecibeMigrantes == TRUE & Continente == "Asia" & PIB>mean(PIB, na.rm=TRUE)
```

- Con el operador `&` concatenamos condiciones ("debe recibir inmigración y ser de Asia y…").
- En vez de `RecibeMigrantes == TRUE`, se puede usar directamente `Migracion>0`.
- Se pueden hacer operaciones directamente en la condición: `mean(PIB, na.rm=TRUE)`.
- ¿Por qué `mean(PIB, na.rm=TRUE)`? Ejecutad estos comandos y observad las diferencias:

```
mean(Mundo$PIB)
```

```
mean(Mundo$PIB, na.rm=TRUE)
```

Más en el Tema 3.

Transformaciones: Filtrar observaciones

Transformación de observaciones en el que se selecciona un subconjunto de datos. Lo usaremos para saber qué países cumplen la condición establecida en la anterior diapositiva.

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda

(...)
Importar datos
Conjunto de datos en paquetes
Conjunto de datos en activo ▶
Modificar vars. del conjunto de datos

(...)
Establecer el nombre de casos...
Filtrar el conjunto de datos en activo...
Ordenar el conjunto de datos activo...
(...)

```
CondicionCompleja == TRUE
```

Once países cumplen la condición: Bahrain, Brunei, Cyprus, Hong Kong, Israel, Kuwait,...

Tema 2: R Commander – Manejo de datos

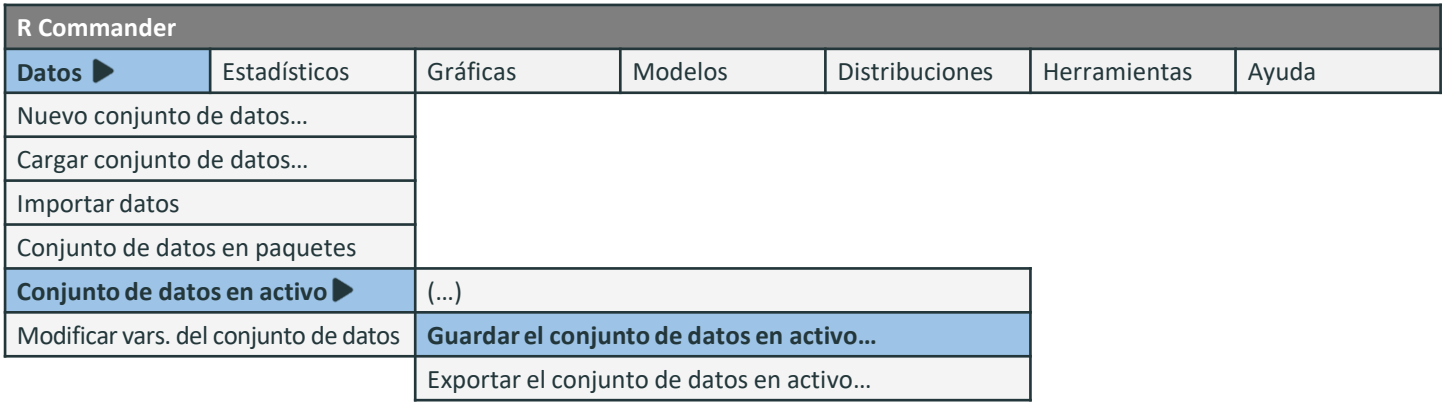
❸ Exportación de ficheros de datos

Después de importar los datos y transformarlos, nos puede interesar **guardar los datos procesados** para continuar trabajando en otro momento o para leerlo desde otro programa.

Podemos usar formatos genéricos o específicos de R.

Exportación de ficheros de datos

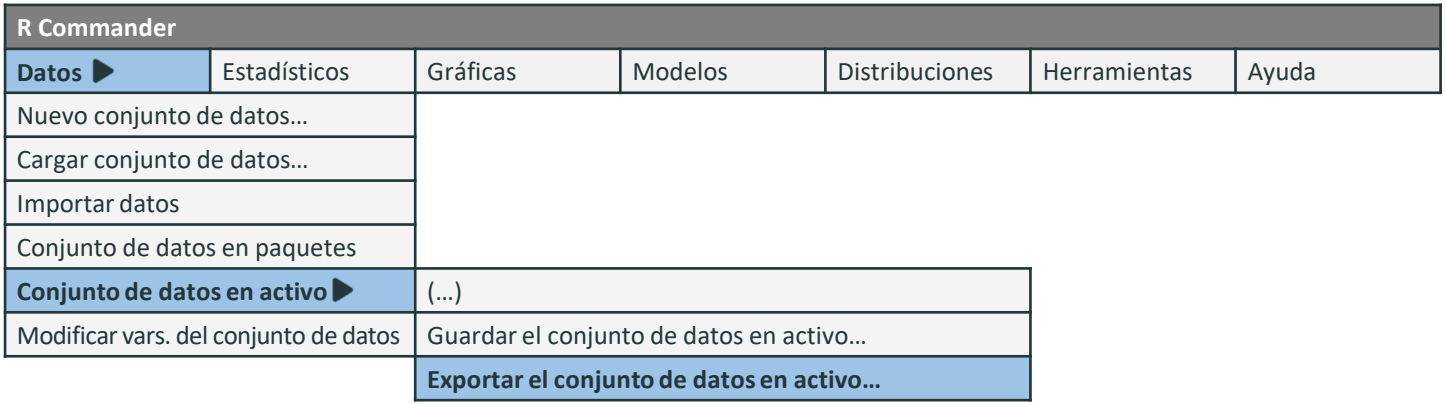
Para guardar el conjunto de datos en formato RData (.RData o .rda; ficheros nativos de R):



En este formato, la lectura de los datos se hace con *Datos / Cargar conjunto de datos...*

Exportación de ficheros de datos

Si, por el contrario, queremos guardar los datos en otro tipo de fichero, como TXT o CSV:



Como ya hemos visto, estos formatos requieren usar *Datos / Importar conjunto de datos...*

TEMA 2

R Commander – Estadística descriptiva

Estadística e IA con 

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 2: R Commander – Estadística descriptiva

Índice

1. Manejo de datos
2. Estadística descriptiva
    A. Univariante
    B. Bivariante
    C. Multivariante
2. ANOVA
4. Regresión lineal

Conjunto de datos: Covid-19

Se trata de un conjunto de datos con la **extensión CSV** (comma-separated values).

Es un formato de fichero tabulado usado de forma universal para almacenar datos.

COVID-19

Medición de diversas variables relacionadas con la pandemia en 220 países del mundo (a 13/5/21).

Variables categóricas	1. `country`: Nombre del país. Variable identificatoria.	
	2. `continent`: Continente al que pertenece el país.	
Variables numéricas	3. `total_confirmed`: Número de casos.	8. `total_cases_per_1m_population`: Casos/1M habitantes.
	4. `total_deaths`: Número de muertes.	9. `total_deaths_per_1m_population`: Muertes/1M habitantes.
	5. `total_recovered`: Número recuperados.	10. `total_tests`: Número de tests.
	6. `active_cases`: Número de casos activos.	11. `total_tests_per_1m_population`: Tests/1M de habitantes.
	7. `serious_or_critical`: Número de casos graves.	12. `population`: Población del país.

Conjunto de datos: Covid-19

Desde R Commander, podemos leer datos en formato CSV mediante el siguiente comando: *Datos / Importar datos / Desde archivo de texto, URL o portapapeles*.

En la ventana emergente, debemos modificar las siguientes opciones:

- Nombre del conjunto de datos: “Covid”
- Separador de campos: “Comas [,]”
- Dejamos el valor por defecto en el resto de las opciones.

Podemos echar un vistazo al conjunto de datos con el botón de visualizar conjunto de datos.

Estadística descriptiva

Rama de la estadística encargada del estudio de conjuntos de datos para **describir sus características y comportamientos** a través de medidas numéricas y gráficos.

En función del número de variables que se usan en el estudio, puede ser:

- **1D – Univariantes**. Estudio de una variable para describir su comportamiento aislado. Se usan medidas y gráficos diferentes en función de la naturaleza de la variable.
- **2D – Bivariantes**. Estudio de la relación existente entre dos variables. De nuevo, el tipo de las variables intervinientes determina el uso de unas medidas y gráficos concretos.
- **ND – Multivariantes**. Estudio de la interacción entre un conjunto de tres o más variables.

Tema 2: R Commander – Est. descriptiva

❶ Estadística descriptiva univariante

Estudio del comportamiento aislado de una variable. En función del **tipo de la variable** (numérica o categórica), se usa un conjunto diferente de **medidas numéricas y gráficos**.

1D: Var. categóricas – Medidas

Numéricamente, las variables categóricas se estudian con **tablas de frecuencia** con frecuencias absolutas (nº de observaciones por categoría) o relativas (porcentaje de observaciones).

En el tema 1, ya vimos cómo se creaban tablas frecuencias de una variable categórica: con *Estadísticos / Resúmenes / Distribución de frecuencias*. Para la variable `continent`, serían:

```
counts:
Africa      Asia      Australia/Oceania      Europe      North America South America
58          49        12                     48          39             14

percentages:
Africa      Asia      Australia/Oceania      Europe      North America South America
26.36       22.27     5.45                   21.82       17.73          6.36
```

1D: Var. categóricas – Gráficos

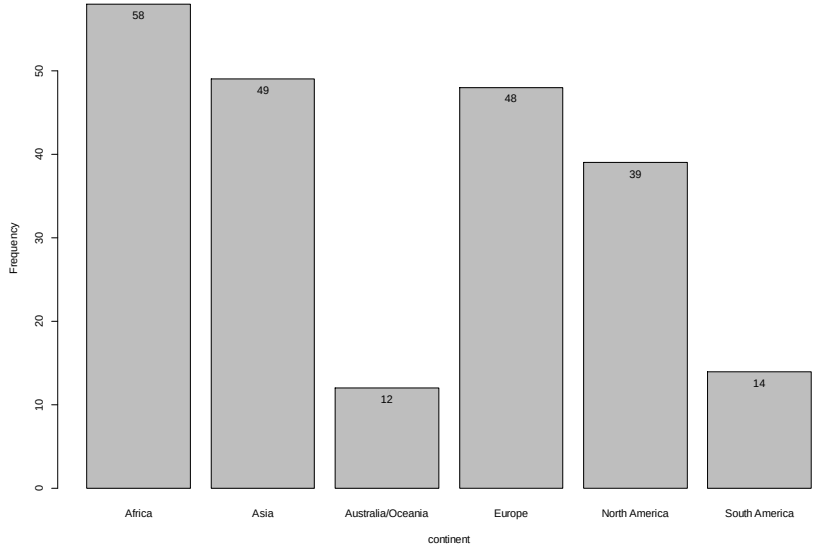
Existen diferentes gráficos para visualizar las frecuencias observadas en una variable categórica. Los principales son el **diagrama de barras** y el **diagrama de sectores**.

En R Commander, los comandos para crear estos gráficos son los siguientes:

R Commander						
Datos	Estadísticos	**Gráficas** ▶	Modelos	Distribuciones	Herramientas	Ayuda
		Gama de colores...				
		(...)				
		Gráficas de barras				
		Gráficas de sectores				
		(....)				

1D: Var. categóricas – Gráficos

Diagrama de barras



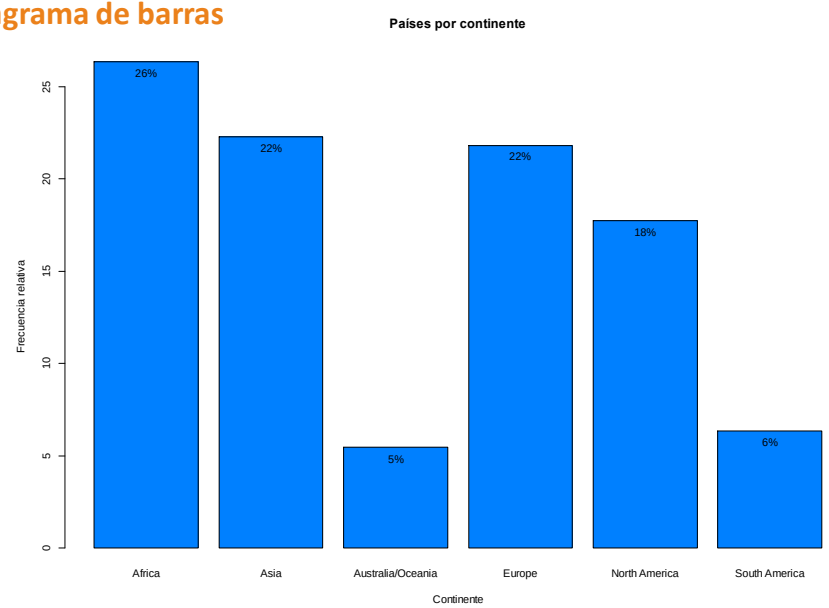
En opciones, podemos elegir:

- Frecuencias abs. o rel.
- Colores de las barras.
- Nombres de los ejes

La ventana gráfica de R Commander permite guardar el gráfico o copiarlo.

1D: Var. categóricas – Gráficos

Diagrama de barras

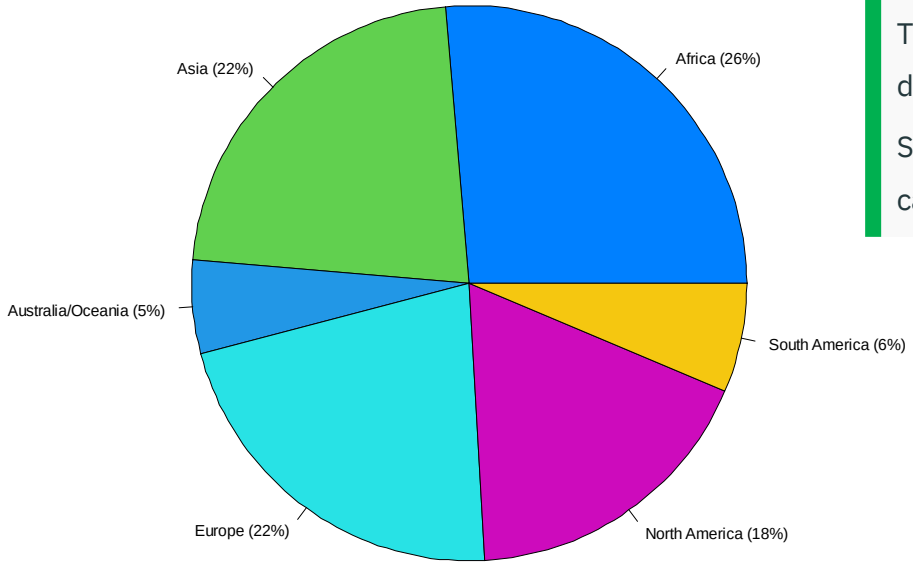


Si usamos las frecuencias relativas, suman en total 100 %.

Podemos guardar los gráficos en formatos de alta resolución con *Archivo / Guardar como...*

1D: Var. categóricas – Gráficos

Diagrama de sectores o Piechart



Tiene la misma información que el diagrama de barras.

Se debe usar sólo si el número de categorías es bajo.

1D: Var. numérica – Medidas

En el tema 1, vimos también cómo se obtenían los estadísticos resumen de una variable numérica: con *Estadísticos / Resúmenes / Conjunto de datos activo*.

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes ▶		Conjunto de datos activo			
	Tablas de contingencia		**Resúmenes Numéricos...**			
	Medias		Distribución de frecuencias...			
	(...)		(...)			

Podemos configurar las medidas en la pestaña “Estadísticos” de la ventana emergente. Antes de proseguir, realizaremos un pequeño repaso sobre las medidas o **estadísticos** más usados.

1D: Var. numérica – Medidas

Las medidas más empleadas para una variable numérica X con n observaciones son:

- **Media**: promedio de las observaciones. $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- **Varianza**: medida de la dispersión. $S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2$

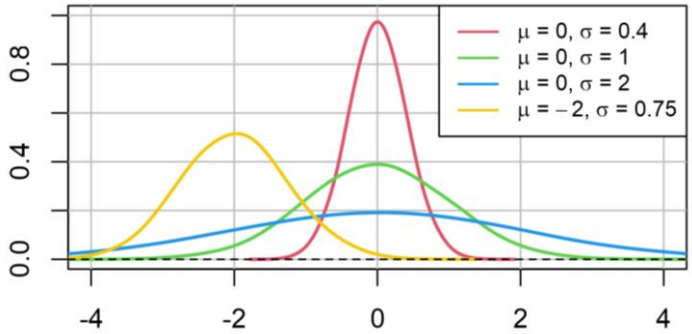


Figura. Distribución normal en función de los parámetros de media (μ) y desviación estándar (σ).

1D: Var. numérica – Medidas

Las medidas más empleadas para una variable numérica X con n observaciones son:

- **Desviación estándar**: mide la dispersión en las unidades originales. $S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2}$
- **Coeficiente de variación** (para variables positivas): Medida relativa de dispersión. $CV = {}^{S}/_{\bar{X}}$

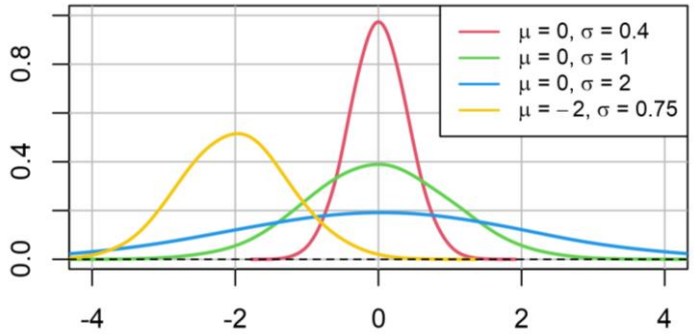


Figura. Distribución normal en función de los parámetros de media (μ) y desviación estándar (σ).

1D: Var. numérica – Medidas

- **Asimetría** (Skewness): medida de asimetría de la muestra. $A = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{X})^3}{S^3}$

$A \cong 0$	$A > 0$	$A < 0$
Distribución simétrica	D. asimétrica positiva (cola dcha. más larga)	D. asimétrica negativa (cola izq. más larga)

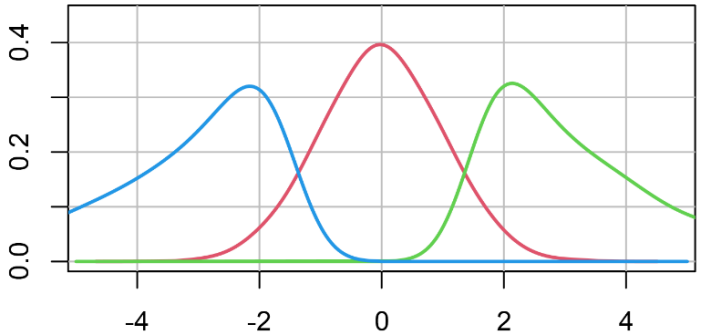


Figura. Distribuciones con diferentes asimetrías (azul: asimétrica positiva, roja: simétrica, verde: asimétrica negativa).

1D: Var. numérica – Medidas

- **Apuntamiento** (Kurtosis): Medida de las colas de una muestra. $K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{X})^4}{S^4}$

$K \cong 3$	$K > 3$	$K < 3$
Distribución normal	Distribución más apuntada	Distribución menos apuntada

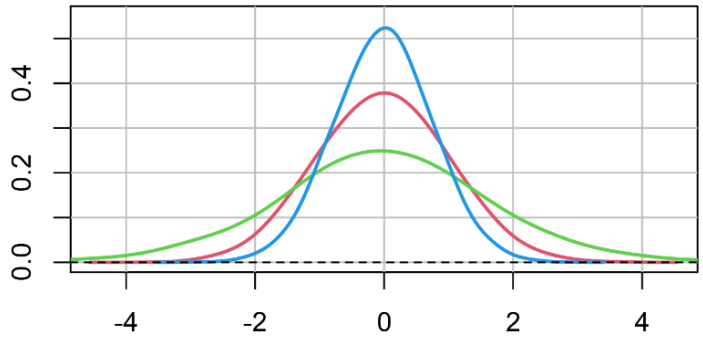


Figura. Distribuciones con diferentes valores de apuntamientos (de más apuntada a menos: azul, roja -K=3-, verde).

1D: Var. numérica – Medidas

- **Percentil** p: valor que deja a la izquierda p % de observaciones (y $100 - p$ % a la derecha). El máximo se denomina percentil 100 y, en ocasiones, el mínimo "percentil 0".

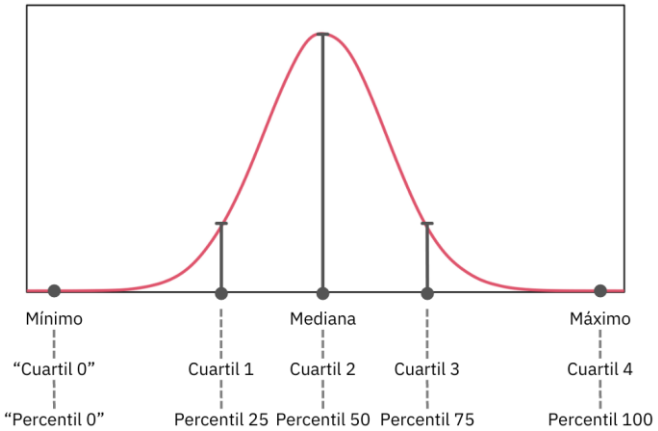


Figura. Medidas cuantiles de una distribución: percentiles y cuartiles.

1D: Var. numérica – Medidas

- **Cuartiles**: dividen la muestra en 4 trozos (Q_1, Q_2, Q_3, Q_4); corresponden con los percentiles 25 %, 50 % (mediana), 75 % y 100 % (máximo). A veces, se denomina al mínimo Q_0.

Rango intercuartílico: medida de dispersión. $IQR = Q_3 - Q_1$.

La mediana es una medida central de la distribución más robusta que la media.

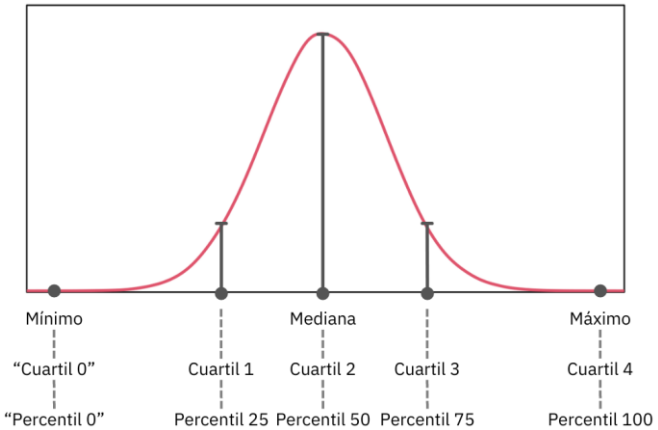


Figura. Medidas cuantiles de una distribución: percentiles y cuartiles.

1D: Var. numérica – Medidas

Es el momento de poner a prueba nuestros conocimientos estudiando la variable nº de muertes por millón de habitantes. Añadiremos el apuntamiento y la kurtosis a las medidas calculadas por defecto.

```
    mean       sd  IQR skewness kurtosis  0% 25% 50% 75% 100% n  NA
581.9353 712.1218 879 1.446576 1.355186  1  42 274 921 2987 201 19
```

- De media, han fallecido 581 personas por cada millón de habitantes en cada país.
- El número de fallecidos por cada millón es desigual entre países, nos lo indica:
  1. La diferencia entre la media y la mediana. En distribuciones simétricas, tienden al mismo valor.
  2. Existe asimetría positiva ($A > 0$): hay países con notablemente más fallecidos que otros.
  3. La kurtosis indica que la distribución no es apuntada: los fallecidos están repartidos ($K < 3$).
  4. Hay 19 valores faltantes.

1D: Var. numérica – Gráficos

El estudio gráfico de variables permite afianzar conclusiones obtenidas previamente a través de medidas numéricas, así como ayudarnos a dar una nueva visión a los datos.

Los gráficos más empleados con variables numéricas son en el **gráfico de caja** y el **histograma**. En R Commander, los comandos para crear estos gráficos son los siguientes.

R Commander						
Datos	Estadísticos	**Gráficas** ▶	Modelos	Distribuciones	Herramientas	Ayuda
		Gama de colores...				
		(...)				
		Histograma...				
		Diagrama de caja...				
		(...)				

1D: Var. numérica – Gráficos

Etiquetas para las observaciones

Muchos de los gráficos que genera R incluyen etiquetas en aquellas observaciones que resulten ser anómalas (outliers) con sus nombres de fila (que, por defecto, son los números de fila).

Para facilitar el estudio, haremos que los nombres de fila sean los de los países (variable `country`).

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda

(...)
Conjunto de datos en paquetes
Conjunto de datos en activo ▶
Modificar vars. del conjunto de datos

(...)
Variables del conjunto de datos en activo...
Establecer el nombre de casos...
Filtrar el conjunto de datos en activo...
(...)

1D: Var. numérica – Gráficos

Histograma

En este gráfico, la variable se discretiza y se calcula la frecuencia observada en cada tramo. El aspecto del histograma varía con el tipo de segmentación y el nº de segmentos.

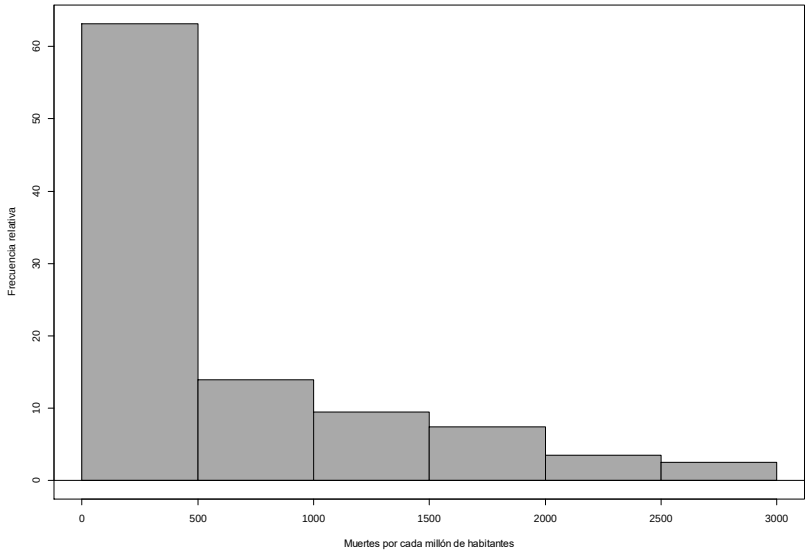
Seleccionamos la variable `total_deaths_per_1m_population` y las siguientes opciones para estudiar los porcentajes asociados a discretizar en 5 niveles (muy baja, baja, media, alta y muy alta).

- Número de clase: “5”

  NOTA: A veces, no es posible discretizar una variable en el nº de segmentos especificado.
- Escala de los Ejes: “Porcentajes”

1D: Var. numérica – Gráficos

Histograma

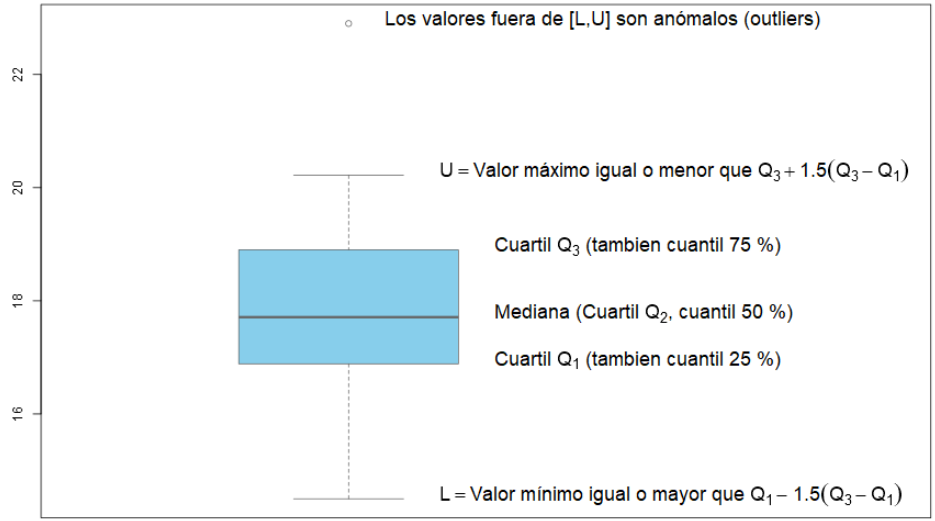


Podemos ver como cerca del 60% de los países han tenido relativamente muy pocos fallecidos.

Parece que unos pocos países acumulan más muertes por millón.

1D: Var. numérica – Gráficos

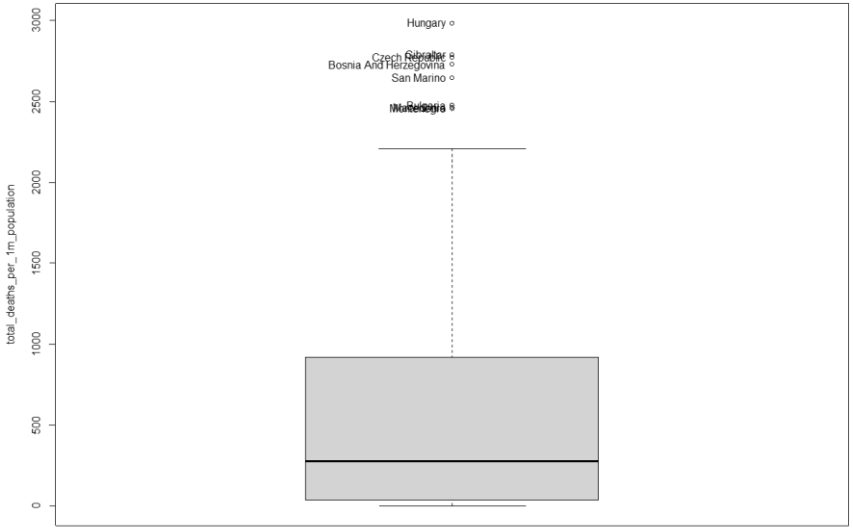
Diagramas de caja



Es un diagrama muy útil para estudiar la distribución de una variable numérica.

1D: Var. numérica – Gráficos

Diagramas de caja



Unos pocos países tienen muchas más muertes/1M de habitantes.

Esta variable puede resultar engañosa para valorar ciertos "países" (San Marino, Gibraltar,...).

Tema 2: R Commander – Est. descriptiva

❷ Estadística descriptiva bivariante

Estudio de dos variables con el objetivo de conocer la relación entre ambas. El **tipo de las variables** intervinientes determina el uso de ciertas **medidas numéricas y gráficos**.

2D: Categórica + Categórica – Medidas

Para estudiar la interacción, se usan **tablas de frecuencia de doble entrada** o de contingencia.

Frecuencias absolutas

		Continente			
		Africa	Europa	Oceania	
IDH	Bajo	38	5	1	44
	Alto	10	39	12	61
		48	44	13	105

Frecuencias relativas

		Continente			
		Africa	Europa	Oceania	
IDH	Bajo	36 %	5 %	1 %	42%
	Alto	9 %	37 %	12 %	58%
		45 %	42 %	13 %	100%

Condicionado por filas

		Continente			
		Africa	Europa	Oceania	
IDH	Bajo	86 %	12 %	2%	100%
	Alto	16 %	64 %	20 %	100%

Condicionadas por columnas

		Continente		
		Africa	Europa	Oceania
IDH	Bajo	79 %	11 %	8 %
	Alto	21 %	89 %	82 %
		100%	100%	100%

2D: Categórica + Categórica – Medidas

¿Ha tenido más impacto el Covid-19 en continentes con menos recursos como África?

Para hacer este análisis con los datos que tenemos, son necesarias dos operaciones previas:

1. Eliminar del conjunto países con poblaciones muy pequeñas o enormes (muy atípicos).

R Commander						
Datos ▶	Estadísticos	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda

(...)
Importar datos
Conjunto de datos en paquetes
Conjunto de datos en activo ▶
Modificar vars. del conjunto de datos

(...)
Establecer el nombre de casos...
Filtrar el conjunto de datos en activo...
Ordenar el conjunto de datos activo...
(...)

```
population > 1000000 &
population < 1000000000
```

2D: Categórica + Categórica – Medidas

2. Crearemos la variable `impactoCovid` como la discretización en 3 clases del nº de casos / 1M habitantes. Nuestra hipótesis es que, a mayor proporción de casos, más impacto en la vida del país.

Para segmentar una variable numérica, seguimos los pasos vistos en el primer tema: *Datos / Modificar variables del conjunto de datos activo / Segmentar variable numérica* e introducimos:

- Variable: `total_cases_per_1m_population`.
- Número de clase: 3
- Nombre de la nueva variable: `impactoCovid`.

Después de dar al botón de "Aceptar", introduciremos los segmentos:

- 1: Bajo, 2: Medio, 3: Alto.

2D: Categórica + Categórica – Medidas

Introduciremos la variable `impactoCovid` en la ventana emergente de la siguiente opción:

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia ▶		**Tabla de doble entrada**			
	Medias		Tablas de entradas múltiples			
	(….)		Introducir y analizar una tabla de doble entrada…			

¿Ha tenido más impacto el Covid-19 en continentes con menos recursos como África?

No, quizás por su población más joven, aunque… se hicieron menos tests, datos menos fiables, …

```
impactoCovid Africa Asia Australia/Oceania Europe North America South America
       Bajo     50   31                 3     10            11             4
       Medio     0   11                 0     23             3             6
       Alto      0    1                 0      2             0             0
```

2D: Categórica + Categórica – Gráficos

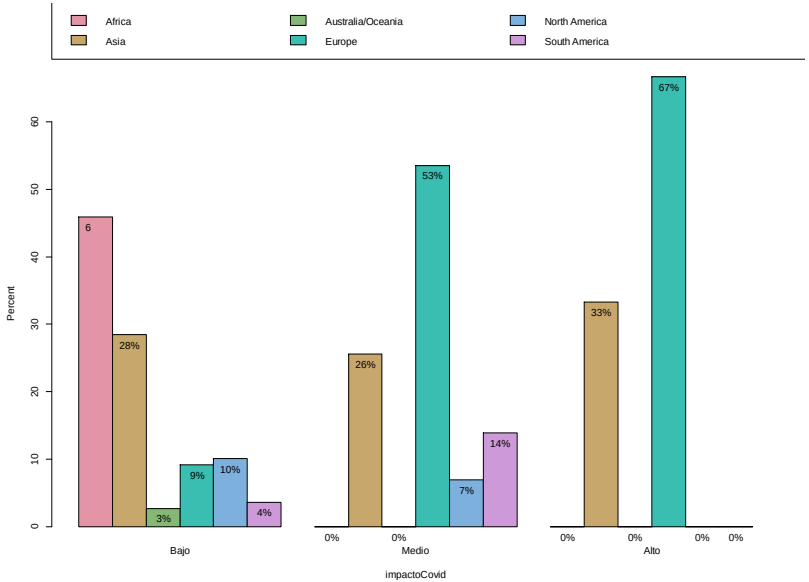
La conclusión obtenida numéricamente también se puede plasmar en forma de gráfico. En R Commander, estos gráficos se crean con los mismos comandos que los categóricas univariantes.

R Commander						
Datos	Estadísticos	**Gráficas ▶**	Modelos	Distribuciones	Herramientas	Ayuda
		Gama de colores...				
		(...)				
		Gráficas de barras				
		Gráficas de sectores				
		(....)				

Una vez que seleccionemos el gráfico, introduciremos la segunda variable en “Gráfica por grupos”. Podemos crear, por ejemplo, un diagrama de barras múltiple entre impacto Covid y continente.

2D: Categórica + Categórica – Gráficos

Diagramas de barras múltiple



Bajo nuestra hipótesis, Europa es el continente más castigado.

Podemos configurar, en las opciones:

- Porcentajes mostrados.
- Multibarra o barras apiladas.

2D: Numérica + Categórica – Medidas

Para estudiar el posible efecto de una variable categórica en otra numérica, se calculan diferentes medidas de la variable numérica para cada uno de los niveles de la categórica.

Dicho esto, podemos plantearnos la siguiente pregunta:

¿Ha tenido Europa una mortalidad significativamente mayor que la del resto de continentes?

De forma similar al caso del diagrama de barras, accedemos a los comandos univariantes *Estadísticos / Resumen / Resúmenes Numéricos* e introducimos los siguientes datos:

- Seleccionamos la variable `total_deaths_per_1m_population`.
- Al hacer click en el botón "Resumir según...", seleccionamos `continent`.

2D: Numérica + Categórica – Medidas

¿Ha tenido Europa una mortalidad significativamente mayor que la del resto de continentes?

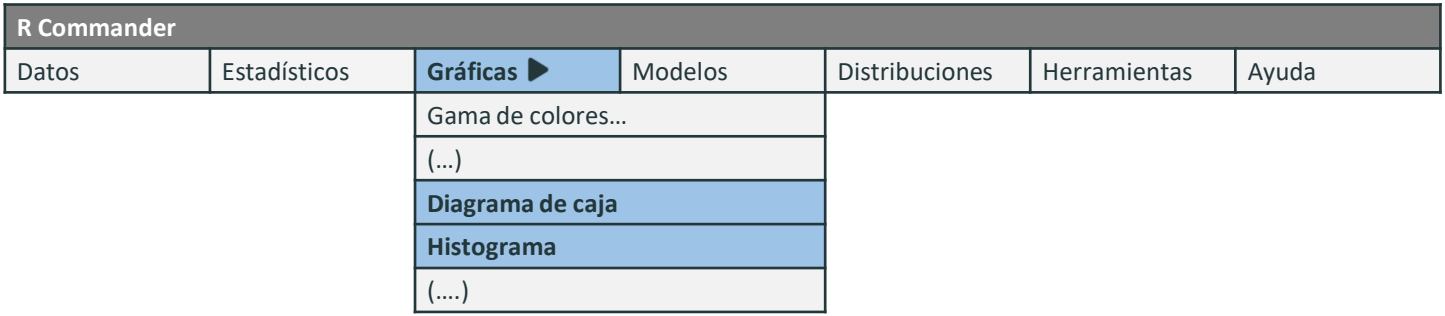
- Tanto la media como la mediana son superiores a las de otros continentes.
- Las cifras son dispares en Europa, como indican la desviación estándar y el rango intercuartílico.
- Parece que en Sudamérica también hay países muy castigados.

Parece que sí, ya que su población es la más envejecida. Pero... ¿los datos de ciertos países pertenecientes a los continentes asiático, africano o americano son fiables?

```
                  mean       sd    IQR  0%     25%    50%     75% 100%
Africa           113.3958 208.8786  70.00   3   12.50   33.0   82.50  969
Asia             296.1500 356.5628 351.25   4   50.25  165.0  401.50 1434
Australia/Oceania 18.0000  15.3948  15.00   5    9.50   14.0   24.50   35
Europe          1469.0286 741.1059 956.00 141 1001.00 1483.0 1957.00 2987
North America    603.6429 603.6380 478.75  23  188.50  379.5  667.25 1794
South America   1260.1000 554.7907 497.50  82 1028.00 1272.0 1525.50 1991
```

2D: Numérica + Categórica – Gráficos

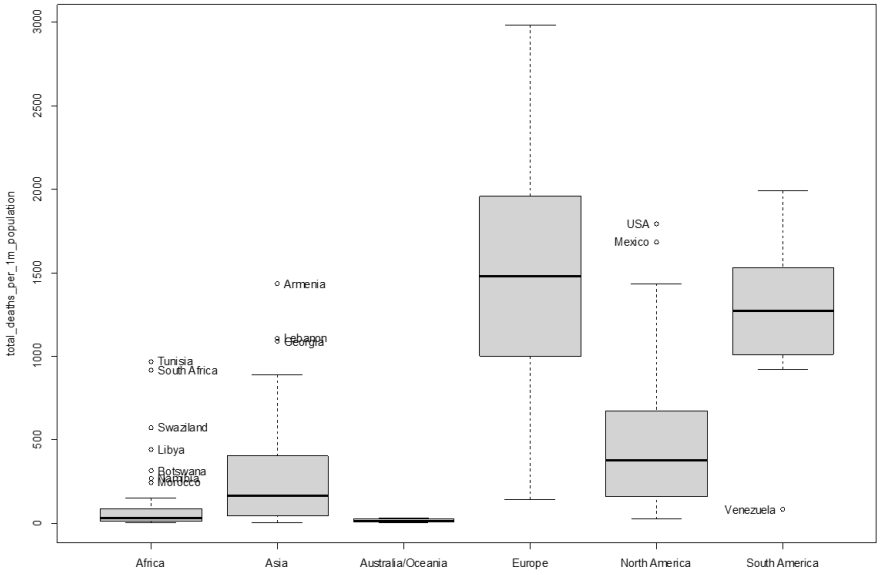
Los gráficos empleados para estudiar la relación entre una variable numérica y una categórica son los **diagramas de cajas múltiple** y los **histogramas múltiples**. El comando usado es, de nuevo:



Una vez seleccionado el gráfico, elegiremos la variable numérica (`total_deaths_per_1m_population`, en este caso) e introduciremos la categórica en "Gráfica por grupos" (`continent`).

2D: Numérica + Categórica – Gráficos

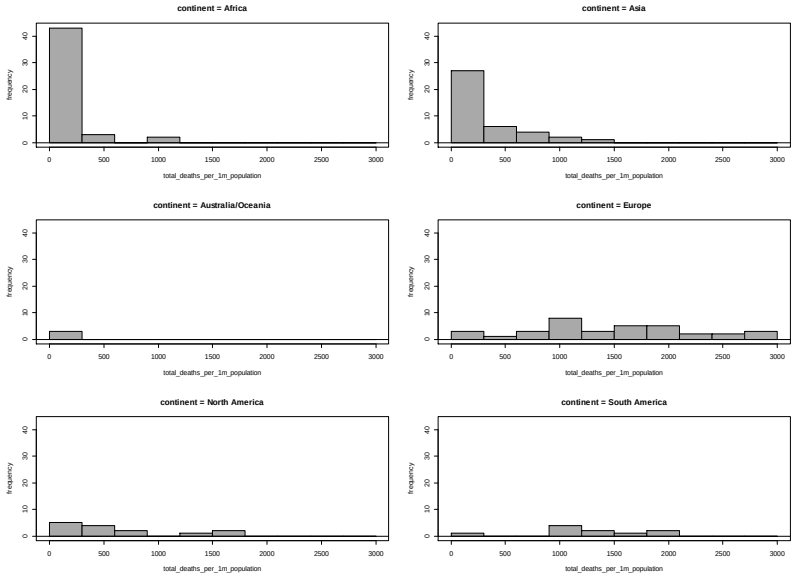
Diagramas de cajas múltiple



Europa ha tenido una mortalidad superior a la de otros continentes, pero hay otros países con cifras similares (USA).

2D: Numérica + Categórica – Gráficos

Diagramas de barras múltiple



Se observa el impacto desigual por continente (Australia vs. Europa).

En Europa, hay una gran dispersión de valores, así como unas cifras de fallecidos muy elevadas.

2D: Numérica + Numérica – Medidas

La relación entre dos variables numéricas se cuantifica, principalmente, con estadísticos que miden la existencia de relaciones lineales entre ellas. Las medidas más empleadas son:

- **Covarianza**: $\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})$

$\mathrm{Cov}(X,Y) \cong 0$	$\mathrm{Cov}(X,Y) > 0$	$\mathrm{Cov}(X,Y) < 0$
Ausencia de asociación	Asociación creciente	Asociación decreciente

- **Coeficiente de correlación**: $r_{XY} = \frac{\mathrm{Cov}(X,Y)}{S_X S_Y}$. Toma valores entre -1 y 1.

A menudo, se calculan matrices de correlaciones.

$$\begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & \dots \end{array} \\ \begin{pmatrix} 1 & r_{12} & r_{13} & r_{14} & \dots \\ r_{12} & 1 & r_{23} & r_{24} & \dots \\ r_{13} & r_{23} & 1 & r_{34} & \dots \\ r_{14} & r_{24} & r_{34} & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array}$$

$r_{XY} \cong 0$	$r_{XY} > 0$	$r_{XY} < 0$	$\lvert r_{XY} \rvert \cong 1$
Ausencia de asociación	Asociación creciente	Asociación decreciente	Asociación lineal exacta

2D: Numérica + Numérica – Medidas

Con los datos que tenemos, nos podemos preguntar:

¿Dos países que hayan tenido el mismo número de contagiados (por millón de habitantes) han tenido el mismo número de fallecidos (por millón de habitantes)?

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes ▶		Conjunto de datos activo			
	Tablas de contingencia		(...)			
	Medias		**Matriz de correlaciones...**			
	(....)		(...)			

Seleccionamos de forma simultánea (manteniendo pulsada la tecla Control) las variables `total_cases_per_1m_population` y `total_deaths_per_1m_population`.

2D: Numérica + Numérica – Medidas

En la parte inferior de la diapositiva, está el comando de R usado y el resultado de su ejecución.

La correlación es alta y positiva, por lo que los países con más casos han tenido más fallecidos.

R Commander no dispone, en su distribución básica, de una opción para calcular la covarianza.

Por lo tanto, debemos de emplear comandos de programación de R.

```
cor(Covid[,c("total_cases_per_1m_population", "total_deaths_per_1m_population")], use="complete")
```

```
                           total_cases_per_1m_population     total_deaths_per_1m_population
total_cases_per_1m_population                     1.0000000                       0.8150672
total_deaths_per_1m_population                    0.8150672                       1.0000000
```

2D: Numérica + Numérica – Medidas

Para calcular la covarianza, copiaremos el comando ejecutado por R Commander y, después de pegarlo, sustituiremos `cor` (“cor-relation”) por `cov` (“cov-ariance”).

Los elementos diagonales de la matriz resultante son estimadores de las varianzas, mientras que el resto son la covarianza entre las dos variables.

```
cov(Covid[,c("total_cases_per_1m_population", "total_deaths_per_1m_population")], use="complete")
```

```
                            total_cases_per_1m_population     total_deaths_per_1m_population
total_cases_per_1m_population                    1164809648                        20390226
total_deaths_per_1m_population                     20390226                          537282
```

2D: Numérica + Numérica – Gráficos

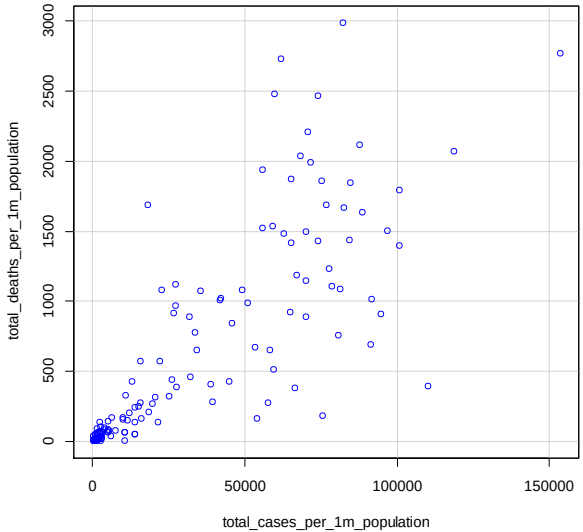
Desde un punto de vista gráfico, la relación entre variables numéricas se estudia con **diagramas de dispersión**. El comando de R Commander que se usa con este propósito es el siguiente:

R Commander						
Datos	Estadísticos	**Gráficas** ▶	Modelos	Distribuciones	Herramientas	Ayuda
		Gama de colores...				
		(...)				
		Diagrama de dispersion				
		Matriz de diagramas de dispersión				
		(....)				

Como variable X (llamada independiente), seleccionamos `total_cases_per_1m_population`, mientras que como Y (dependiente), `total_deaths_per_1m_population`.

2D: Numérica + Numérica – Gráficos

Diagrama de dispersión



Es cierto que, a más casos, más fallecido en general.

Sin embargo, la relación entre las variables no es completamente lineal.

3D: Numérica + Numérica + Categórica

En último lugar, vamos a realizar el gráfico anterior segmentando por continente. También añadiremos algún elemento gráfico para que podamos obtener más conclusiones.

De nuevo, usamos el comando *Gráficas / Diagrama de Dispersión*.

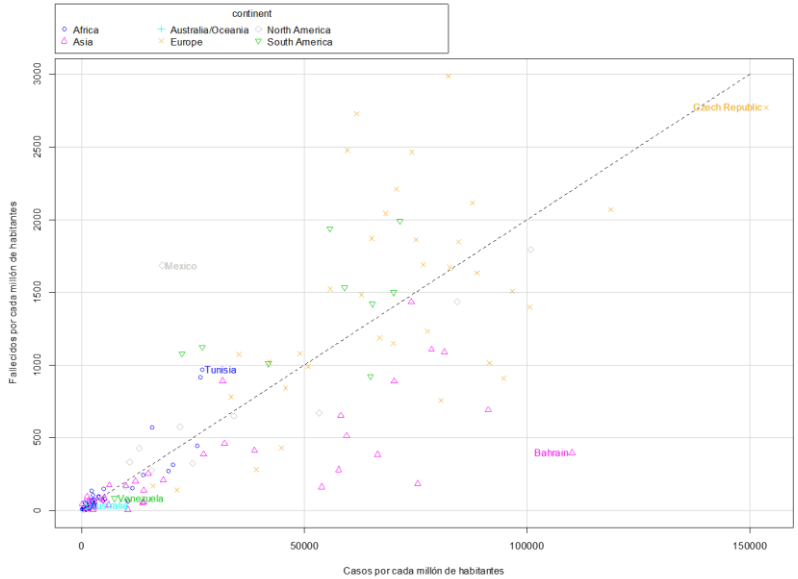
- Variable X: `total_cases_per_1m_population`
- Variable Y: `total_deaths_per_1m_population`.
- Seleccionamos en el botón "Gráfica por Grupos" la variable `continent`.

En la línea de comandos de R Studio, ejecutad la siguiente orden para añadir una línea al gráfico:

```
lines(x = c(0,150000), y = c(0,3000), lty=2)
```

3D: Numérica + Numérica + Categórica

Diagrama de dispersión con la interacción de una variable categórica



Europa, de nuevo, sale como el continente más castigado.

No tienen una relación lineal perfecta (hipótesis representada por la recta).

En la Republica Checa, parece que se han realizado pocos tests respecto al número de fallecidos.

TEMA 2
R Commander – ANOVA

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 2: R Commander – ANOVA

Índice

Analysis of Variance (ANOVA)

Modelo estadístico empleado para analizar las **diferencias existentes entre grupos o individuos**. Desde un punto de vista formal, permite cuantificar diferencias en media de una variable numérica en función de los niveles de otra u otras variables categóricas.

Existen muchas técnicas asociadas a estos modelos que permiten, entre otras cosas, estimar las diferencias entre grupos o comprobar si las diferencias entre grupos son significativas.

¿Tiene efecto un tratamiento?

¿Existen diferencias entre grupos?

¿Diferencias en la evolución?

Conjunto de datos: Waste de multcomp

Trabajaremos con este conjunto de datos. Recoge el volumen de residuos que se genera en 5 plantas de residuos industriales en función de su entorno y la temperatura de combustión.

Para utilizar un conjunto de datos de un paquete, debemos cargarlo con el comando `library()`. Debemos ejecutar la siguiente línea en la consola de R Studio (panel inferior izquierdo):

```
library(multcomp)
```

Después, podemos cargar el conjunto de datos `waste` como tal desde R Commander con la opción *Datos / Cargar conjunto de datos* y seleccionando el paquete `multcomp`.

Conjunto de datos: Waste de multcomp

El conjunto de datos `waste` se compone de las siguientes variables:

1. `temp`: Temperatura a la que se ha generado el residuo (variable categórica, valores {low, medium, high}).
2. `envir`: Entorno al que pertenece la planta (variable categórica, valores {env1, env2, env3, env4, env5}).
3. **`waste`**: Volumen de residuos generados por la planta en un entorno específico y a una temperatura concreta. Se trata de una variable numérica.

Podéis echar un vistazo a los datos desde RStudio. Interesa saber si el volumen de residuos varía con la temperatura y el entorno. Una primera cuestión para afrontar el problema sería:

¿Una mayor temperatura implica generar menos residuos?

Tema 2: R Commander – ANOVA

❶ ANOVA de un factor

Modelo ANOVA más sencillo que asume que el valor de una variable numérica está determinado por el **grupo al que pertenece**, determinado por una variable categórica.

ANOVA de un factor

Sea Y una variable numérica cualquiera y X una variable categórica que toma $j = \{1, \dots, J\}$ valores.

Un **modelo ANOVA de un solo factor** tiene la siguiente expresión:

$$y_i = \mu_j + \varepsilon_{i,j} \qquad \text{si } x_i = j$$

Para una observación, se asume que el valor de Y (var. respuesta) está determinado por el valor de X ("grupo al que pertenece") más un efecto aleatorio ("diferencia de la observación respecto al grupo").

Se estima μ_j como la media de las observaciones pertenecientes a la clase j.

La formulación vista se denomina modelo de medias. Es equivalente al **modelo de efectos**:

$$y_i = \mu + \tau_j + \varepsilon_{i,j} \qquad \text{si } x_i = j$$

"¿Cuánto varía la media general al pertenecer al grupo j?"

ANOVA de un factor

Podemos crear un modelo ANOVA de un factor usando el siguiente comando:

R Commander						
Datos	**Estadísticos** ▶	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia					
	Medias ▶		Test t para una muestra			
	Proporciones		(...)			
	Varianzas		**ANOVA de un factor...**			
	(...)		(...)			

En la venta emergente, seleccionamos `temp` en grupos y `waste` en variable explicada.

ANOVA de un factor

Medias de los Grupos

Si temp = “high”, más residuos. En el caso de “baja” y “media”, no hay diferencias significativas.

La dispersión por clase es similar y el n.º de observaciones de cada clase es el mismo.

	Media	Desv. estándar	Nº obs.
High	9.88	1.77	10
Medium	7.62	1.31	10
Low	7.87	1.53	10

Si voy a poder calcular una media siempre, **¿Qué relevancia tiene el nº de observaciones?**

Uno de los fundamentos de la estadística es que, cuantas más observaciones tengamos (en la muestra), conoceremos mejor la verdadera naturaleza de los datos (de la población subyacente).

➔ ¿Y si un grupo tiene $n = 1$?

➔ Más en el tema sobre Inferencia Estadística.

ANOVA de un factor

TABLA ANOVA

Simplificación p-valor.

	Grados de libertad (DF)	Suma de cuadrados	Cuadrado medio	Estadístico F	p-valor	Significancia
temp	2	30.69	15.346	6.353	0.00548	**
Residuo (Error)	27	65.22	2.416			

Variabilidad no explicada (el modelo asume que hay una media común por grupo).

Codificación de la significancia:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Valores entre [0,1]. Cuanto más alto, más probable es $\mu_1 = \mu_2 = \cdots = \mu_J$
(Contraste: H_0: $\mu_1 = \mu_2 = \cdots = \mu_J$ vs. H_1: no todas μ_j las son iguales).

Se genera un volumen diferente de residuos según la temperatura.

ANOVA de un factor

TABLA ANOVA

NOTA: se asume que la muestra es equilibrada (cada grupo → mismo n.º de observaciones).

	Grados de libertad (DF)	Suma de cuadrados	Cuadrado medio	Estadístico F	p-valor	Significancia
temp	2 $J - 1$	30.69 $SSB = \sum_{j \in temp} (\hat{\mu}_j - \hat{\mu})$	15.346 $MSB = {}^{SSB}/_{J-1}$	6.353 $Fval = {}^{MSB}/_{MSE}$	0.00548 $Fval \sim F(J-1, n-J)$	**
Residuo (Error)	27 $n - J$	65.22 $SSE = \sum_{i,j \in temp} (y_{ij} - \hat{\mu}_j)$	2.416 $MSE = {}^{SSE}/_{n-J}$			

Codificación de la significancia:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Siendo $\hat{\mu}$ la media de todas las observaciones y $\hat{\mu}_j$ la media de las observaciones de la clase j.

Es interesante recalcar que $SST = \sum_i (y_i - \hat{\mu}) = SSB - SSE$.

ANOVA de un factor: Método de Tukey

Parece que varía el volumen de residuos según la temperatura, pero...

¿Entre qué temperaturas (baja, media, alta) hay diferencias significativas?

El método de Tukey sirve para estudiar diferencias en media entre grupos.

R Commander						
Datos	**Estadísticos** ▶	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia					
	Medias ▶		Test t para una muestra			
	Proporciones		(...)			
	Varianzas		**ANOVA de un factor...**			
	(...)		(...)			

Después de seleccionar las variables, habilitamos “Comparación dos a dos de las medias”.

ANOVA de un factor: Método de Tukey

Las cuentas subyacentes son similares al ANOVA.

Simplificación p-valor.

	Estimador	Error estándar	Estadístico T	p-valor	Significancia
low - high == 0	-2.015	0.6951	-2.899	0.01962	*
medium - high == 0	-2.256	0.6951	-3.246	0.00859	**
medium - low == 0	-0.241	0.6951	-0.347	0.93603	

Codificación de la significancia:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimación de las diferencias (más cerca de 0, menos diferencia)

Toma valores entre [0,1] y, cuanto más alto sea, más probable es que $\mu_1 = \mu_2$. (Contraste: H_0: $\mu_1 - \mu_2 = 0$ vs. H_1: $\mu_1 - \mu_2 \neq 0$).

Por otro lado, R Commander también hace una estimación en intervalo de la diferencia.

Hay diferencias entre la temperatura alta y el resto.

Tema 2: R Commander – ANOVA

❷ ANOVA de dos factores

Modelo que asume que el valor de la variable numérica respuesta está determinado por los valores tomados en otras **dos variables categóricas** (grupos pertenecientes).

ANOVA de dos factores

Sea Y una variable numérica y X_1, X_2 dos var. categóricas que toman $j = \{1, \dots, J\}$ y $k = \{1, \dots, K\}$ valores.

Un **modelo ANOVA de dos factores** tiene la siguiente expresión:

$$y_i = \mu_{j,k} + \varepsilon_{i,j,k} \qquad \text{si } x_{1,i} = j \text{ y } x_{2,i} = k$$

Para una observación, se asume que el valor de Y (var. respuesta) depende de los valores de X_1, X_2 ("grupos a los que pertenece") más un efecto aleatorio ("algo que diferencia a la observación del grupo").

Se estima $\mu_{j,k}$ como la media de todas las observaciones pertenecientes a las clases j y k.

A veces, interesa estudiar el equivalente **modelo de efectos**:

$$y_i = \mu + \alpha_j + \beta_k + \gamma_{j,k} + \varepsilon_{i,j,k} \qquad \text{si } x_{1,i} = j \text{ y } x_{2,i} = k$$

Interacción entre pertenecer a los grupos j y k

"¿Cuánto varía la media general al pertenecer al grupo j (o k)?"

ANOVA de dos factores

Podemos crear un modelo ANOVA de dos factores usando el siguiente comando:

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia					
	Medias ▶		Test t para una muestra			
	Proporciones		(...)			
	Varianzas		ANOVA de un factor...			
	Tests no paramétricos		**ANOVA de múltiples factores...**			
	(...)		(...)			

Después, seleccionamos `temp` y `envir` en factores, y `waste` en variable explicada.

ANOVA de dos factores

Tablas de Medias (y Desviaciones Estándar)

MEDIAS	Env1	Env2	Env3	Env4	Env5
High	7.76	9.58	9.08	11.56	11.41
Medium	6.42	6.68	7.09	8.58	9.34
Low	6.49	8.54	9.54	6.58	8.16

Se genera más residuo en el entorno 5 y menos en el 1. Hay cierta interacción entre factores: con temp = low, el volumen de residuos varía según entorno. En otros casos, no es clara la interacción.

Por otro lado, hemos obtenido las tablas de nº. de observaciones, que nos indica que la muestra es equilibrada, y de desviación estándar, que no admite mucho comentario al ser $n = 2$.

ANOVA de dos factores

TABLA ANOVA

	Grados de libertad (DF)	Suma de cuadrados	Estadístico F	p-valor	Significancia
Envir	4	24.68	5.25	0.0075	**
Temp	2	30.69	13.06	0.0005	***
Envir:Temp	8	22.91	2.44	0.0650	.
Residuo (Error)	15	17.62			

Codificación de la significancia:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Como podéis apreciar, la salida obtenida es bastante similar a la del ANOVA de un factor, añadiéndose el **término de interacción** entre variables ("¿Guardan los grupos alguna relación?").

Hay diferencias en los residuos generados según entorno y temperatura.

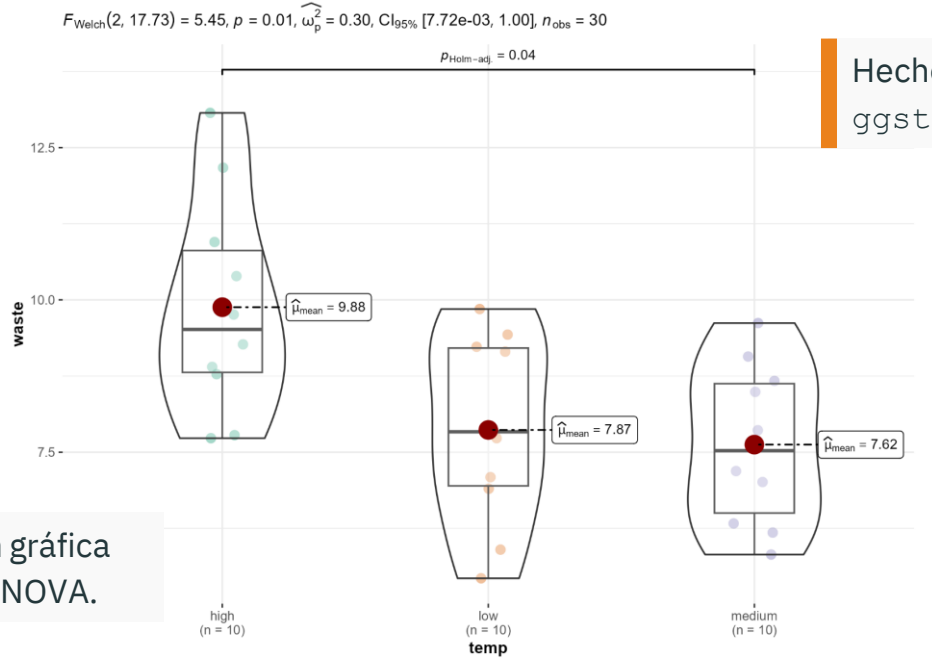
Conclusión

Estáis listos para trabajar con todo tipo de ANOVAs, ya que supone generalizar las ideas vistas (3 factores: 4 interacciones: `fact1:fact2`, `fact1:fact3`, `fact2:fact3` y `fact1:fact2:fact3`).

- En el tema de inferencia, profundizaremos sobre otros conceptos que hemos visto de soslayo: ¿Qué es un **p-valor**? ¿Un intervalo (de confianza)? ¿Y un contraste (de hipótesis)?
- Hay **modelos ANOVA especializados** en ciertos tipos de datos: más de una medida por individuo, el comportamiento de cada grupo no es homogéneo, variaciones intra-grupo,... Muchos de estos modelos están implementados en paquetes adicionales de R.

Por otro lado, otro tema de sumo interés es la presentación de las conclusiones obtenidas a partir de aplicar un modelo ANOVA. Existen diferentes formatos para ello.

Conclusión



Hecho con el paquete `ggstatsplot`.

Representación gráfica de un modelo ANOVA.

ANEXOS

Tema 2: R Commander – ANOVA

Estadística e Inteligencia Artificial con R

ANEXO: Comandos de modelos ANOVA

En este anexo, se reproducen los comandos que generan las salidas de R Commander del tema. Suponemos haber cargado los datos:

```
data(waste, package = "multcomp")
```

- ANOVA de un solo factor (temperatura):

```
aov(formula = waste ~ temp, data = waste)
```

- Método de Tukey para ANOVA de un solo factor (temperatura):

```
library("multcomp")
summary(glht(AnovaModel.1, linfct = mcp(temp = "Tukey")))
```

ANEXO: Comandos de modelos ANOVA

En este anexo, se reproducen los comandos que generan las salidas de R Commander del tema. Suponemos haber cargado los datos:

```
data(waste, package = "multcomp")
```

- ANOVA de dos factores (temperatura, entorno) con interacción:

```
AnovaModel.2 <- lm(waste ~ envir * temp, data=waste)
Anova(AnovaModel.2)
```

- ANOVA de dos factores (temperatura, entorno) sin interacción:

```
AnovaModel.3 <- lm(waste ~ envir + temp, data=waste)
Anova(AnovaModel.3)
```

TEMA 2
R Commander – Regresión lineal

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 2: R Commander – Regresión lineal

Índice

Tema 2: R Commander – Regresión lineal

❶ Regresión lineal simple

Modelo de regresión más sencillo en el que se busca **predecir una variable numérica a partir de otra, también, numérica**. Se presupone que la relación entre ellas es lineal.

Regresión lineal simple

Modelo en el que se relaciona una variable numérica (Y), llamada variable **respuesta o explicada**, con otra, también, numérica (X), denominada variable **independiente o regresor**. Se presupone que la relación entre estas variables es **lineal**.

El modelo cumple la siguiente expresión:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Intercept o término independiente
Valor de Y cuando $X = 0$.

Pendiente
Aumento esperado de Y por cada aumento en una unidad de X.

Conjunto de datos: Marketing

Afrontaremos el problema de predecir el beneficio de una empresa en función de lo que invierte en diferentes medios. El conjunto de datos se puede cargar con *Datos / Cargar conjunto de datos*.

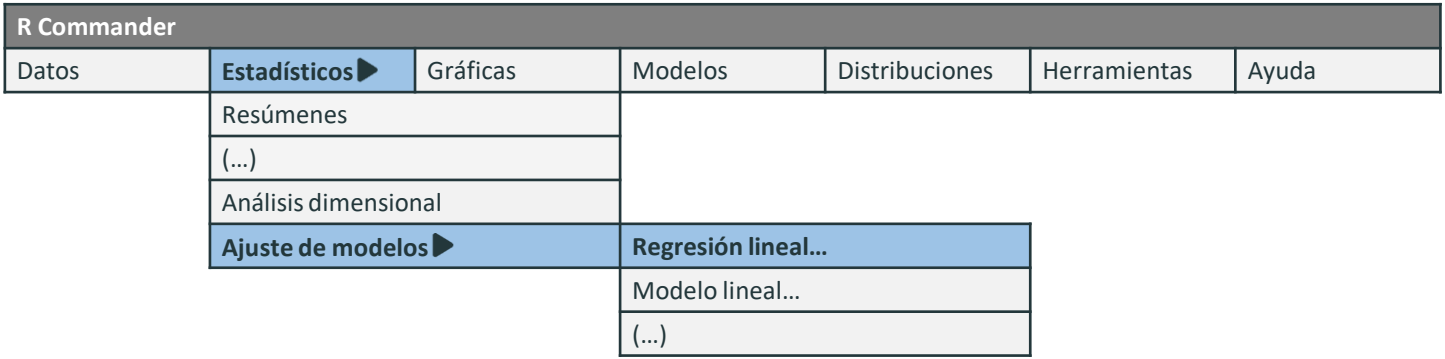
1. `youtube`: Millones de $ invertidos en Youtube.
2. `facebook`: Millones de $ invertidos en Facebook.
3. `newspaper`: Millones de $ invertidos en periódicos.
4. `sales`: Ventas generadas.

Nuestra primera aproximación al problema será afrontando la siguiente pregunta:

¿Podemos predecir los beneficios de una empresa en función del dinero que invierte en Youtube?

Regresión lineal simple

El comando de R Commander para ajustar un modelo de regresión lineal es:



En la venta emergente correspondiente a este comando, seleccionamos `sales` en la variable explicada y `youtube` en las variables explicativas.

Regresión lineal simple

En la consola, podemos ver los comandos que ajustan el modelo y el resultado del mismo.

```
RegModel <- lm(sales~youtube, data=marketing)
summary(RegModel)
```

```
Residuals:      Min       1Q       Median       3Q      Max
                -10.0632  -2.3454  -0.2295   2.4805   8.6548

Coefficients:                Estimate Std. Error t value  Pr(>|t|)
               (Intercept) 8.439112   0.549412   15.36   <2e-16   ***
               youtube     0.047537   0.002691   17.67   <2e-16   ***
               ---
               Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.91 on 198 degrees of freedom
Multiple R-squared:  0.6119,         Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

$$\text{sales} = 8.439 + 0.0475 \times \text{youtube}$$

Regresión lineal simple

TABLA REGRESIÓN

Simplificación p-valor.

	Estimador	Error estándar	Valor T	p-valor	Significancia
(Intercept)	8.439	0.549	15.360	$< 2 \times 10^{-16}$	***
youtube	0.047	0.002	17.670	$< 2 \times 10^{-16}$	***

Codificación de la significancia:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Valores estimados para los parámetros β_0, β_1.

Toma valores entre [0,1]. Cuanto más bajo sea, la variable es más significativa, es decir, es menos probable es que su parámetro sea 0. (El contraste realizado es: H_0: $\beta_i = 0$ vs. H_1: $\beta_i \neq 0, i = \{0,1\}$).

Hay una relación lineal entre `sales` y `youtube`. Cuando `youtube` $= 0$, el valor de `sales` no es 0.

Regresión lineal simple

Bondad de Ajuste de la Regresión

```
Residuals:      Min       1Q       Median       3Q      Max
                -10.0632  -2.3454  -0.2295   2.4805   8.6548

Coefficients:                Estimate Std. Error t value  Pr(>|t|)
               (Intercept) 8.439112   0.549412   15.36   <2e-16   ***
               youtube     0.047537   0.002691   17.67   <2e-16   ***
               ---
               Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.91 on 198 degrees of freedom
Multiple R-squared:  0.6119,         Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

El R cuadrado (R^2) nos indica la variabilidad explicada por el modelo de la total en la variable respuesta. Cuanto más cerca esté de 1 (máximo), mejor es la bondad de ajuste.

Regresión lineal simple

Bondad de Ajuste de la Regresión

```
Residuals:      Min       1Q       Median       3Q      Max
                -10.0632  -2.3454  -0.2295   2.4805   8.6548

Coefficients:                Estimate Std. Error t value  Pr(>|t|)
               (Intercept) 8.439112   0.549412   15.36   <2e-16   ***
               youtube     0.047537   0.002691   17.67   <2e-16   ***
               ---
               Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.91 on 198 degrees of freedom
Multiple R-squared:  0.6119,         Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

- R^2 ajustado: medida de bondad de ajuste que penaliza modelos con muchas variables.
- Estadístico F: cuanto más bajo es el p-valor, más fuerte es la relación lineal.

Regresión lineal simple

Representación del Modelo

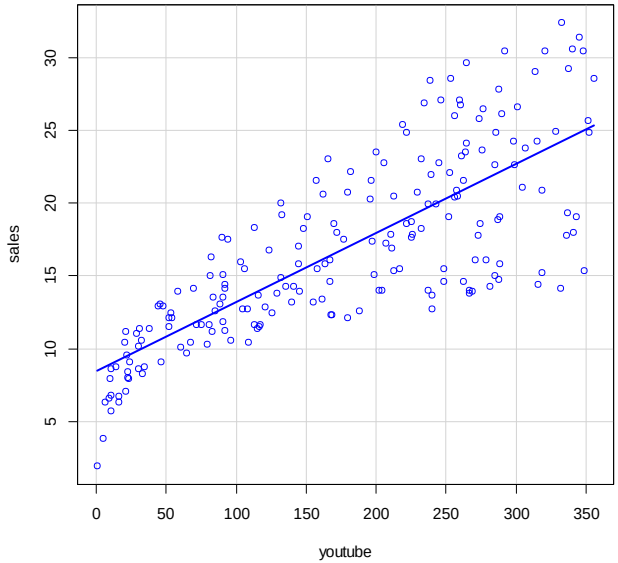
El modelo se ha añadido a la ventana de R Commander, en la misma zona que el conjunto de datos.

Para visualizar el ajuste, vamos a crear un diagrama de dispersión. En la anterior lección, vimos que se usaba el comando *Gráficas / Diagrama de Dispersión*. Realizamos la siguiente selección:

- Variable X: `youtube`
- Variable Y: `sales`.
- En la pestaña opciones, habilitamos “Línea de mínimos cuadrados”.
- En la pestaña opciones, seleccionamos en Identificar observaciones, “No identificar”.

Regresión lineal simple

Representación del Modelo



Existe una relación lineal creciente: a más inversión en Youtube, más beneficio para la empresa.

Aun así, hay variabilidad en la variable respuesta que no explica la Youtube.

¿Cuál es el beneficio para una empresa que invierte 200 millones de dólares en Youtube?

```
predict(RegModel.1, data.frame(youtube=200))
```

Regresión lineal simple

Errores de Predicción (Residuos o Residuales)

```
Residuals:      Min       1Q       Median       3Q      Max
                -10.0632  -2.3454  -0.2295   2.4805   8.6548

Coefficients:                 Estimate Std. Error t value  Pr(>|t|)
(Intercept) 8.439112   0.549412   15.36   <2e-16   ***
youtube     0.047537   0.002691   17.67   <2e-16   ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.91 on 198 degrees of freedom
Multiple R-squared:  0.6119,         Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

Modelo: $Y = \beta_0 + \beta_1 X + \varepsilon$

Se usan para medir la bondad de ajuste, así como para diagnosticar problemas en el modelo.

Dado un modelo, cada observación tiene asociada un error de predicción:

$$\varepsilon_i = y_i - \hat{y}_i = y_i - (\beta_0 + \beta_1 x_i)$$

Regresión lineal simple: Diagnóstico

El **estudio de los residuos del modelo** se usa para el diagnóstico del ajuste. Desde un punto de vista formal, se presupone que se distribuyen como una normal: $\varepsilon \sim N(0, \sigma^2)$.

- "Las predicciones, en promedio, ni infra- ni sobrestiman el valor real" (el error es, en media, 0).
- "La dispersión del error de predicción es constante, similar a lo largo de la variable respuesta".

A parte de medir la bondad de ajuste, su estudio permite afrontar otras cuestiones:

- ¿Hay algún punto que está influyendo mucho en la predicción (**punto "palanca"**)?
- ¿Hace falta transformar alguna variable ($\log X, X^2, \sqrt{X}, \ldots$)?
- ¿Se cumplen las **hipótesis del modelo lineal**?

1. Linealidad.
2. Normalidad.
3. Independencia (errores entre observaciones son independientes).
4. Homocedasticidad (σ^2 común).

Regresión lineal simple: Diagnóstico

En R Commander, podremos obtener diversos gráficos diagnósticos en *Modelos / Gráficas / Gráficas de diagnóstico*. Veremos algunas nociones básicas a partir de los gráficos provistos.

R Commander						
Datos	Estadísticos	Gráficas	**Modelos ▶**	Distribuciones	Herramientas	Ayuda
			Selecciona el modelo activo...			
			Resumir el modelo			
			(...)			
			Test de hipótesis			
			Diagnósticos numéricos			
			Gráficas ▶		**Gráficas básicas de diagnóstico**	
					(...)	

Regresión lineal simple: Diagnóstico

Estos dos gráficos tienen una interpretación similar.

¿Existen errores de predicción mucho más altos que otros?

Observaciones anómalas o erróneas tienen residuos altos (lejos del 0, marcados con el índice de obs.), que pueden ser influencia negativa para el modelo.

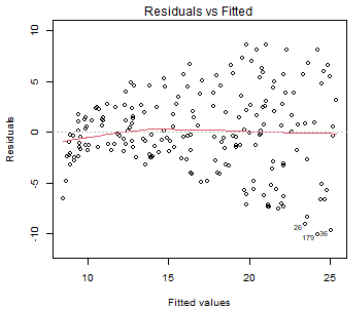
→ Estudio detallado: ¿Descartar alguna?

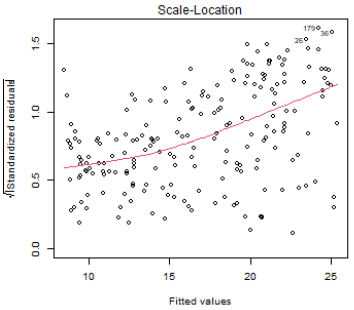
¿Se cometen errores de forma "similar" a lo largo del eje?

En caso contrario, hay más error en valores bajos o altos (extremos eje) o la disposición de los residuos no es aleatoria.

→ ¿Homocedasticidad, independencia, normalidad?

→ ¿Transformar alguna variable?





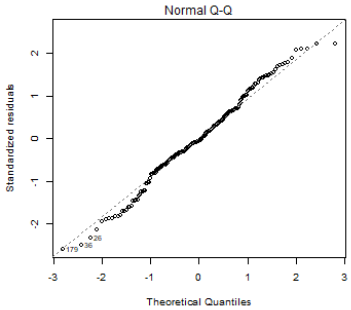
Regresión lineal simple: Diagnóstico

Estos gráficos representan la normalidad e influencia de puntos.

Normal Q-Q Plot: ¿Los residuos son normales?

Los residuos (puntos) deben situarse próximos a los cuantiles teóricos de una normal (línea discontinua).

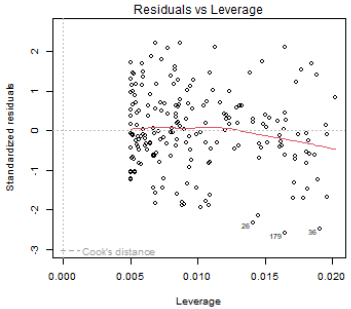
→ ¿Transformar alguna variable?



Residuos vs. apalancamiento: ¿Hay puntos demasiado influyentes?

Los residuos más altos (lejos del cero, marcados con el índice de observación) son sospechosos de ser “palanca”.

→ Estudio detallado: ¿Descartar alguno?



Tema 2: R Commander – Regresión lineal

❷ Regresión lineal múltiple

Generalización del modelo de regresión lineal simple.

Relaciona linealmente una variable numérica (Y; denominada variable respuesta, dependiente o explicada) con un **conjunto de regresores** ($X_1, X_2, \ldots, X_p$; variables independientes o explicativas).

Regresión lineal múltiple

El modelo tiene la siguiente formulación:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

Pendiente de X_1

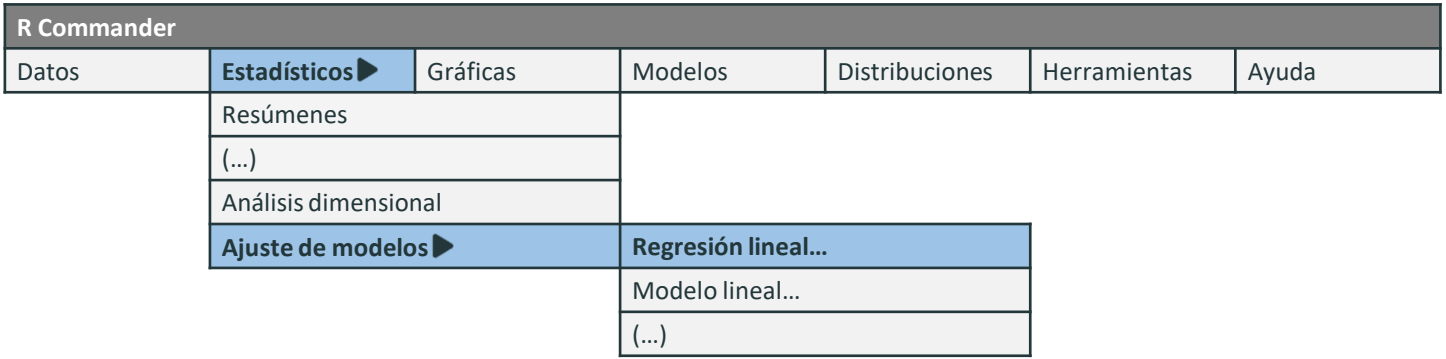
Aumento esperado de Y por cada aumento unitario de X_1, siendo 0 el resto de variables.

En nuestro caso, podemos enriquecer el modelo lineal simple al incluir más regresores.

¿Podemos predecir los beneficios de una empresa en función del dinero que invierte en diferentes medios (Youtube, Facebook y periódicos)?

Regresión lineal múltiple

El comando de R Commander es el mismo que para una regresión lineal simple:



En la venta emergente, seleccionamos `sales` en la variable explicada y el resto de las variables en las variables explicativas (manteniendo la tecla control pulsada al hacer click).

```
RegModel.1 <- lm(sales~facebook+newspaper+youtube, data=marketing)
summary(RegModel.1)
```

Regresión lineal múltiple

```
Residuals:       Min        1Q    Median        3Q       Max
                -10.5932   -1.0690    0.2902    1.4272    3.3951

Coefficients:                 Estimate Std. Error t value Pr(>|t|)
               (Intercept)   3.526667   0.374290   9.422   <2e-16 ***
               facebook      0.188530   0.008611  21.893   <2e-16 ***
               newspaper    -0.001037   0.005871  -0.177     0.86
               youtube       0.045765   0.001395  32.809   <2e-16 ***
               ---
               Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,         Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

$$\text{sales} = 3.52 - 0.01\,\text{newspaper} + \\ + 0.19\,\text{facebook} + 0.046\,\text{youtube}$$

La variabilidad explicada ha mejorado, de $R^2 = 0.611 \rightarrow 0.897$.

La significancia de `newspaper` es muy baja (alto p-valor): no aporta información al modelo (invertir dinero en periódicos no garantiza más beneficios). **Creamos un nuevo modelo sin la variable `newspaper`.**

Regresión lineal múltiple

$$\text{sales} = 3.51 + 0.19\ \text{facebook} + 0.046\ \text{youtube}$$

```
Residuals:
     Min       1Q   Median       3Q      Max
-10.5572  -1.0502   0.2906   1.4049   3.3994

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.50532    0.35339   9.919   <2e-16 ***
facebook     0.18799    0.00804  23.382   <2e-16 ***
youtube      0.04575    0.00139  32.909   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.018 on 197 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8962
F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

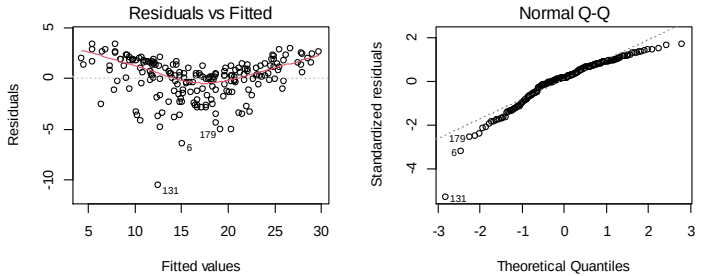
La variabilidad explicada no ha disminuido al quitar `newspaper` ($R^2 = 0.897 \rightarrow 0.897$), lo que indica que no aportaba información al modelo. Todos los términos de la regresión son significativos (p-valores bajos).

Regresión lineal múltiple

Podemos obtener los valores predichos como hicimos en la regresión simple:

```
predict(RegModel.3, data.frame(youtube=200, facebook=50))
```

Por otro lado, podemos echar un vistazo a los gráficos diagnóstico de la regresión múltiple (podemos cambiar el modelo seleccionado desde la interfaz de R Commander).



Ciertas observaciones son muy anómalas: ¿anotaciones erróneas u outliers?, ¿perjudiciales para el ajuste?

Conclusión

Una vez vistos los fundamentos de regresión, os invito a reflexionar sobre ciertas cuestiones:

- Formulación similar a ANOVA → Regresión lineal: generalización del ANOVA.
- ¿Qué es un p-valor? ¿Y un contraste (de hipótesis)? → Más en el tema de inferencia.
- Variables categóricas como regresores → ¿Cómo usarlas? ¿Cuáles son más útiles?
- Diagnóstico de residuos en profundidad → ¿Qué variables transformar y con qué función?
- ¿Y si la relación con los regresores no es lineal? → ¿Interacciones, términos cuadráticos...?
- ¿Modelos de regresión no-lineales? → Más en el tema de IA.

Con esto, terminan nuestras lecciones con R Commander. Toca comenzar a **programar con R**.

ANEXOS

Tema 2: R Commander – Regresión lineal

Estadística e Inteligencia Artificial con R

ANEXO: Comandos de modelos de Regresión

En este anexo, se reproducen los comandos empleados en el tema. Para carga los datos, usamos (recuerda que el fichero debe de estar en el directorio de trabajo actual):

```
load("marketing.rda") # Si tienes problemas, revisa la seccion del tema 3
```

- Modelo de regresión lineal simple:

```
RegModel.1 <- lm(sales~youtube, data=marketing)
summary(RegModel.1)
plot(RegModel.1)
predict(RegModel.1, data.frame(youtube=200))
```

- Modelo de regresión simple sin término independiente:

```
RegModel.2 <- lm(sales~youtube-1, data=marketing)
```

ANEXO: Comandos de modelos de Regresión

- Modelo de regresión múltiple con todas las variables, menos la respuesta:

```
RegModel.3 <- lm(sales~., data=marketing)
summary(RegModel.3)
plot(RegModel.3)
```

- Modelo de regresión múltiple con dos variables:

```
RegModel.4 <- lm(sales~facebook+youtube, data=marketing)
predict(RegModel.4, data.frame(youtube=200, facebook=50))
```

- Modelo de regresión múltiple con dos variables e interacción entre ellas:

```
RegModel.5 <- lm(sales~facebook+youtube+facebook*youtube, data=marketing)
summary(RegModel.5)
```

TEMA 3
Programación básica con R

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 3: Programación básica con R

Índice

Tema 3: Programación básica con R

1 Introducción a R

¿Qué utilidad puede tener aprender a programar en R si ya controlo R Commander? Para ampliar los procedimientos a nuestra disposición: podemos **crearlos nosotros mismos** o aprovechar los procedentes de los **+20.000 paquetes** de R.

Interfaz de R Studio

Su interfaz se compone de cuatro paneles principales:

<table>
<tr><td>Scripts
(oculto por defecto)</td><td>Entorno de variables</td></tr>
<tr><td rowspan="2">Consola de comandos</td><td>Multifunción
(gráficos, paquetes, ayuda,…)</td></tr>
</table>

Programación básica con R

Supongamos que queremos hacer un cálculo trivial, como 4×10. El comando usado será:

```
4*10
```

Podemos obtener el resultado del comando de dos formas diferentes:

- Podemos usar la **línea de comandos**, como hasta ahora, escribiendo el comando y pulsando Enter. La línea de comandos es inmediata y ofrece el historial de comandos.
- Sin embargo, para guardar los comandos empleados, podemos escribirlos en un **Script** ("guion"). Se crean con *File / New File / R Script*, y se guardan con el botón disquete.

Para ejecutarlo, se puede hacer click en el botón "Run" o Control+Enter (a veces Control+R).

Programación básica con R

Los siguientes ejemplos os permitirán usar R como una simple calculadora:

```
5 + 5
5 - 5
3 * 5
(5 + 5) / 3
3 ^ 5
5 %% 3 # Operador módulo
```

Comentarios (precedidos por #): texto que explica el código.

En la programación, el operador fundamental es la **asignación**:

Los dos operadores de asignación (<- y =) son completamente equivalentes.

Podemos probar a hacer las operaciones previas con las variables `a` y `b`.

Programación básica con R

Una variable puede contener todo tipo de valores: números, palabras, lógicos (*booleans*),...

```
miNumero <- 5
```

```
miPalabra <- "Datos"
```

```
miLogico <- TRUE
```

Para imprimir su valor, bastará con escribir solo su nombre: `miNumero`. Otra opción es usar la función `print()`. Las **funciones** realizan operaciones sobre variables (o sobre valores).

```
print(miPalabra)
cos(miNumero)
cosMiNumero <- cos(miNumero)
tolower(miPalabra) # Texto a minusculas
toupper(miPalabra) # Texto a mayusculas
```

> **Buena práctica**: Denominación semántica de las variables.
> Básicamente, denomina a las variables en función de lo que contienen.

Programación básica con R

La programación es una destreza que se adquiere de forma paulatina. Es muy importante saber buscar ayuda. En R, es sencillo: cada función tiene asociada un **manual**.

```
?mean
```

```
help(mean)
```

Las funciones pertenecen a **paquetes** que suelen tener una temática común. En este tema, usaremos funciones de la distribución básica de R. Sin embargo, si nos interesa obtener más información sobre un paquete o sobre sus funciones, podemos usar:

```
help(package="esquisse")
```

https://cran.r-project.org/ → Packages → ... → Manual

Por otro lado, como la comunidad de R es muy amplia, encontrareis muchos recursos en internet.

Tema 3: Programación básica con R

❷ Estructuras de datos en R

En la mayoría de las ocasiones, las variables no contienen un único número, palabra o lógico. En R, existen muchos **tipos de datos complejos** que pueden contener más de un valor: vectores, matrices, data.frames y otros como las listas.

Vectores

Estructura de datos de un único tipo básico. El comando básico asociado es `c` (***combine***):

```
vectorNumerico1 <- c(1,3,5,7)
vectorNumerico2 <- c(1,1,1,1)
vectorCaracter <- c("Estadistica", "ia", "analisis de datos")
```

Con vectores, la aritmética y las funciones (vistas hasta ahora) se aplican elemento a elemento:

```
2*vectorNumerico1+3
sin(vectorNumerico1)
tolower(vectorCaracter)
paste("A",c(1,2,3,4,5),sep="-")
vectorNumerico1 + vectorNumerico2
```

Precaución: "... de un único tipo básico": ejecutad `miVector<- c(1,2,"hola")` y `miVector+1`

Vectores

Otra forma de crear vectores es usar comandos que crean sucesiones de valores:

- El operador `:` crea un vector de enteros que empieza en el primer valor y acaba en el segundo.

```
1:5
```

```
5:1
```

- La función `seq(from,to,...)` crea un vector entre los valores de dos formas diferentes.

```
seq(1, 5, by=0.5)
```

```
seq(1, 5, length.out=6)
```

- La función `rep(x,times)` repite el primer argumento tantas veces como indique el segundo.

```
rep(1,5)
```

```
rep(c(1,2),5)
```

Para conocer los argumentos de los que dispone una función, echad un ojo a su manual: `?seq`

Vectores

Consideremos este conjunto de datos:

```
peso <- c(1.5, 2, 2.5, 1)
consumo <- c(11, 14, 23, 8)
```

	Coche 1	Coche 2	Coche 3	Coche 4
Peso	1.5	2	2.5	1
Consumo	11	14	23	8

Algunas funciones resumen los vectores en un único valor (sumas, cálculo de estadísticos,...):

```
max(peso)       # Maximo
min(peso)       # Minimo
sum(peso)       # Suma
mean(consumo)   # Media
var(consumo)    # Varianza
```

```
sd(consumo)     # Desv. std .
median(consumo) # Mediana
quantile (peso,0.75) # Perc. - 75
cov ( peso,consumo ) # Covarianza
cor ( peso,consumo ) # Correlación
```

Por otro lado, podemos encadenar funciones: `sqrt(var(consumo)) # Desv. std.`

Vectores: Acceso a elementos

Para extraer elementos de un vector, podemos usar **índices**. Estos pueden ser un único número o un vector numérico, como podemos ver en los siguientes ejemplos:

```
peso[1]
peso[c(1,3,4)]
```

```
peso[-2] #Todo menos elemento 2
peso[-c(1,3,4)]
```

Una de las facilidades que ofrece R en la manipulación de vectores (y otras estructuras de datos) es que las operaciones de acceso a elementos permiten su **modificación**:

```
peso[1] <- 4
peso[3] = peso[3] * 1000
```

¿Cómo puedo ampliar un vector? `pesoAmpliado <- c(peso, 2, 1, 3)`

Vectores: Acceso a elementos

Otra opción para acceder a elementos de un vector es usar **condiciones**. Por ejemplo:

```
condicion <- peso>2
peso[condicion]
```

La condición puede ser introducida directamente entre los corchetes, depender de otro vector (**con el mismo número de elementos**) o ser una condición compuesta (&: "_ y _", |: "_ o _"):

```
peso[peso==2]
peso[peso!=2]
peso[peso>=2]
```

```
peso[consumo<10]
peso[peso>=mean(peso)]
peso[(peso>1) & (peso<2)]
```

Cualquier acceso a elementos puede ser usado para modificar un vector:

```
peso[consumo<10] = 2
```

Matrices

Estructura de datos **bidimensional** de un único tipo básico. Su comando básico es `matrix()`:

```
miMatriz <- matrix(1:12,nrow=3)
miMatriz <- matrix(1:12,ncol=4) # Equivalente
miMatriz
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Las operaciones aritméticas, de nuevo, se aplican elemento a elemento. La extracción de elementos es análoga a vectores, teniendo en cuenta que las matrices son bidimensionales:

```
2*miMatriz
2*miMatriz+1
cos(miMatriz)
```

```
miMatriz[2,3]   # 8
miMatriz[2,]    # Fila: 2 5 8 11
miMatriz[,3]    # Columna: 7 8 9
miMatriz[c(1:2),c(1,3)]
```

Para conocer otras operaciones habituales usando matrices, lee el **Anexo A.**

Tema 3: Programación básica con R

❸ Data.frames (*marcos de datos*)

Estructura de datos **bidimensional** en la que se miden variables (columnas), cada una de un **tipo independiente**, a diferentes individuos (filas). Se crean con `data.frame()`. Os deberían ser familiares: los hemos usado con R Commander.

Data.frames: Importación de datos

Para aprender a manejar data.frames, usaremos los datos de “destinosTuristicos.csv”. Podemos cargarlos con R Commander o comandos de R, donde será fundamental el concepto de **directorio de trabajo** (*working directory*): ¿En qué carpeta está trabajando R?

Para cargar el fichero que está, por ejemplo en "K:\Curso\", tengo dos opciones:

A. Indicar directamente la localización del fichero (atentos: es necesario cambiar las “\” por “/”):

```
destinosTuristicos<-read.csv("K:/Curso/destinosTuristicos.csv")
```

B. Realizar el **cambio de directorio de trabajo** al de mi Script (los datos deben de estar en la misma carpeta que el Script). La lectura de los datos se reduce a un sencillo comando:

```
destinosTuristicos<-read.csv("destinosTuristicos.csv")
```

Data.frames: Importación de datos

Cambiar el directorio de trabajo actual

Con la configuración predeterminada, "Documentos" será el directorio de trabajo de R. Allí ubicará los ficheros de entrada y salida del programa. La función `getwd()` sirve para saber el directorio de trabajo. Para cambiarlo, podemos usar la interfaz de **R Studio** (fácil) o comandos:

R Studio			
Session ▶	Build	Debug	(...)
New Session..			
(...)			
Restart R			
Set Working Directory...		**To Source File Location...**	
Load Workspace...		To Files Pane Location	
(...)		(...)	

En cualquiera de los casos, ejecutamos:

```
destinosTuristicos <-
read.csv("destinosTuristicos.csv")
```

Data.frames: Importación de datos

Cambiar el directorio de trabajo actual

Con la configuración predeterminada, "Documentos" será el directorio de trabajo de R. Allí ubicará los ficheros de entrada y salida del programa. La función `getwd()` sirve para saber el directorio de trabajo. Para cambiarlo, podemos usar la interfaz de R Studio (fácil) o **comandos**:

```
setwd("K:/Curso R/")
# Atentos a cambiar "\" por "/"
```

En cualquiera de los casos, ejecutamos:

```
destinosTuristicos <-
read.csv("destinosTuristicos.csv")
```

Data.frames: Importación de datos

Si no hemos tenido problemas en el proceso, ya deberíamos haber cargado en R el conjunto de datos `destinosTuristicos`. Debería aparecer en el panel superior derecha del entorno.

Cada columna tiene un tipo diferente.

VARIABLE

Nombre de la población	Tipo de población	Habitantes	Temperatura media anual	¿Tiene playa?	¿Dispone de ocio?	Preferencia personal
Madrid	Ciudad	3334730	14.5	FALSE	TRUE	3
Barcelona	Ciudad	1664182	15.5	TRUE	TRUE	4
Nueva York	Ciudad	8398748	11.9	TRUE	TRUE	5
Paris	Ciudad	2240621	11.7	FALSE	TRUE	1
Aguasal	Pueblo	24	11.5	FALSE	FALSE	2
Medina del Campo	Pueblo	20679	12.3	FALSE	TRUE	3

OBSERVACIÓN

Tabla. Conjunto de datos de destinos turísticos.

Data.frames: Importación de datos

Ya sabemos leer ficheros CSV (comma-separated values). Si en la carga diese un error, podría solucionarse cambiando algún argumento de `read.csv` (por ejemplo, `header=FALSE`).
Los ficheros más sencillos de abrir con R son los que usan su formato nativo (RDA/RData):

```
load(file = "MisDatos.RData")
```

Sin embargo, la importación de datos resulta tediosa ante la variedad de formatos. A veces, "bastará" con aprender una función (`read.xlsx()`, `read.table()`,...). Algunos consejos son:

- Buscad en internet, por ejemplo, "Open TSV with R".
- Un fichero XLSX se puede transformar en CSV (programa de hoja de cálculo: *Guardar como / Otros formatos*) y cargarse con `read.csv2()`, que usa como separador de valores ";".
- ¡Recordad que **R Commander** ofrece un interfaz amigable para la lectura de datos!

Data.frames: Acceso a elementos

En data.frames, se pueden extraer elementos igual que como hicimos con matrices:

```
destinosTuristicos[2,1]
```

```
[1] "Barcelona"
```

```
destinosTuristicos[4,]
```

```
Nombre Tipo   Habitantes Temperatura Playa Ocio Preferencia
Paris  Ciudad 2240621    11.7        FALSE TRUE 1
```

```
destinosTuristicos[,c(1,3)]
```

```
  Nombre          Habitantes    Nombre            Habitantes
1 Madrid          3334730     4 Paris             2240621
2 Barcelona       1664182     5 Aguasal           24
3 Nueva York      8398748     6 Medina del Campo  20679
```

Por otro lado, los data.frames ofrecen operadores de acceso a variables por su nombre. Si quiero acceder a la variable “Nombre”, no tengo por qué usar `destinosTuristicos[,1]`:

```
destinosTuristicos[,"          Nombre"]
```

```
destinosTuristicos$Nombre
```

Data.frames: Acceso a elementos

Vamos a poner en práctica los operadores de acceso para responder a algunas preguntas:

- ¿Cuántos habitantes tiene Paris?

```
destinosTuristicos[4, "Habitantes"]
destinosTuristicos[destinosTuristicos$Nombre == "Paris", "Habitantes"]
destinosTuristicos$Habitantes[4] # El operador $ devuelve un vector
```

- ¿Cuál es el nombre y temperatura de las ciudades con una temperatura mayor que 15?

```
destinosTuristicos[destinosTuristicos$Temperatura>15, c("Nombre","Temperatura")]
destinosTuristicos[destinosTuristicos[,4]>15, c(1,4)]
```

- ¿Y de aquellos cuya temperatura es mayor que la media?

```
destinosTuristicos[destinosTuristicos$Temperatura >
                   mean(destinosTuristicos$Temperatura), c(1,4)]
```

Los operadores de acceso a elementos permiten modificaciones: `destinosTuristicos[4,3]=225000`

Data.frames: Funciones

Una columna extraída de un data.frame es un vector; podemos hacer cálculos sobre él:

```
mean(destinosTuristicos$Temperatura)
var(destinosTuristicos[,"Habitantes"])
cor(destinosTuristicos$Temperatura, destinosTuristicos$Preferencia)
```

R es *case-sensitive* (distingue mayúsculas y minúsculas en nombres de funciones, objetos y elementos).

Se puede obtener un resumen numérico de las variables del conjunto con `summary()`. En R Commander, obteníamos esta salida con *Estadísticos / Resúmenes / Conjunto de datos activo*.

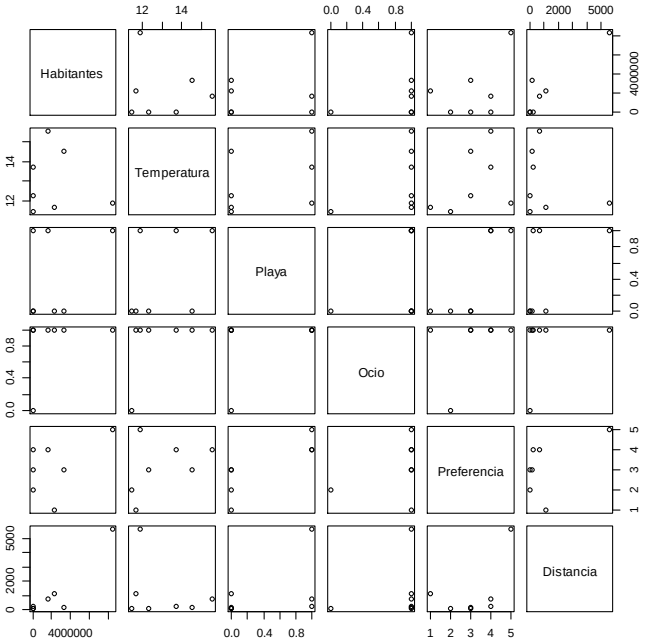
```
 Habitantes        Temperatura         Playa            Ocio           Preferencia
Min.   :     24   Min.   :11.50   Mode :logical   Mode :logical   Min.   :1.000
1st Qu.:  14698   1st Qu.:11.80   FALSE:4         FALSE:1         1st Qu.:2.500
Median :1664182   Median :12.30   TRUE :3         TRUE :6         Median :3.000
Mean   :2238243   Mean   :13.01                                    Mean   :3.143
3rd Qu.:2787676   3rd Qu.:14.10                                    3rd Qu.:4.000
Max.   :8398748   Max.   :15.50                                    Max.   :5.000
```

Data.frames: Funciones

Otro comando útil para echar un vistazo rápido a los datos es `pairs()`.

Una precondición de esta función es que solo se le pasen variables que no sean tipo carácter.

```
pairs(destinosTuristicos[,3:7])
```



Data.frames: Funciones

Adición de observaciones (filas)

1. Creamos otro data.frame con los datos nuevos. La función `data.frame()` crea conjuntos con tantas columnas como elementos antes de los iguales, y tantas filas como valores después de los iguales.
2. Añadimos el pueblo al conjunto de datos con `rbind()`.

```
nuevoPueblo<-data.frame(Nombre="Suances", Tipo="Pueblo", Habitantes=8716,
                        Temperatura=13.7, Playa=TRUE, Ocio=TRUE, Preferencia=4)
destinosTuristicos<-rbind(destinosTuristicos,nuevoPueblo)
```

Adición de variables (columnas)

1. Introducimos en un vector la nueva variable (por ejemplo, la distancia desde Valladolid).
2. Podemos añadir la columna al data.frame inicial con el operador `$` o la función `cbind()`.

```
distanciaValladolid<-c(188, 728, 5630, 1153, 45, 53, 236)
destinosTuristicos$Distancia <- distanciaValladolid
destinosTuristicos2 <- cbind(destinosTuristicos, Distancia=distanciaValladolid)
```

Data.frames: Exportación de datos

Para guardar objetos de R como RDA/RData (ficheros nativos de R), empleamos la función:

```
save(destinosTuristicos, file = "DestinosTuristicos.RData")
```

Si queremos guardar los datos en un tipo de fichero universal, como es CSV, usamos:

```
write.csv(destinosTuristicos, file = "DestinosTuristicosConDistancia.csv")
```

En internet, podéis encontrar información sobre cómo guardar datos en otros formatos. Muchos formatos tienen paquetes específicos para su tratamiento. Sin embargo, el formato CSV debería ser suficientemente universal para poderse leer por cualquier programa.

- ¡Recordad que **R Commander** ofrece un interfaz amigable para exportar los datos!

Tema 3: Programación básica con R

❹ Otras funciones y paquetes

¿Cómo puedo leer en R un formato de datos específico?

¿Cómo puedo aplicar a mis datos una técnica concreta?

¿Cómo creo un procedimiento (reutilizable) para leer de datos?

Creando **funciones propias** o usando funciones de **paquetes**.

Otras funciones

Las **funciones de control de flujo** sirven para aplicar una función u otra dependiendo de una condición o para repetir una operación.
El ANEXO B.1 detalla las principales funciones de control de flujo.

- En función del formato de mis ficheros, uso una función de lectura.
- Quiero comprobar, fila a fila, si hay duplicados en mis datos.

Por otro lado, **programar una función propia** es, en el fondo, aplicar sucesivamente otras funciones (ANEXO B.2).

Las funciones toman una serie de argumentos (datos de entrada), realizan operaciones con estos, y devuelven un valor.

```
> IF(...){}
  ELSE{}
> FOR(...){}
> WHILE(...){}
```

```
> FUNCTION(...) {}
```

Paquetes (*library*)

Igual que podemos crear funciones, miles de usuarios de R comparten sus funciones. Éstas se engloban en **paquetes**, que suelen tener una temática concreta (por ejemplo, técnicas que se usan en psicología, lectura de ficheros Excel, modelos de regresión, ...).

En la parte inferior derecha de R Studio, podéis encontrar un panel para administrar paquetes:

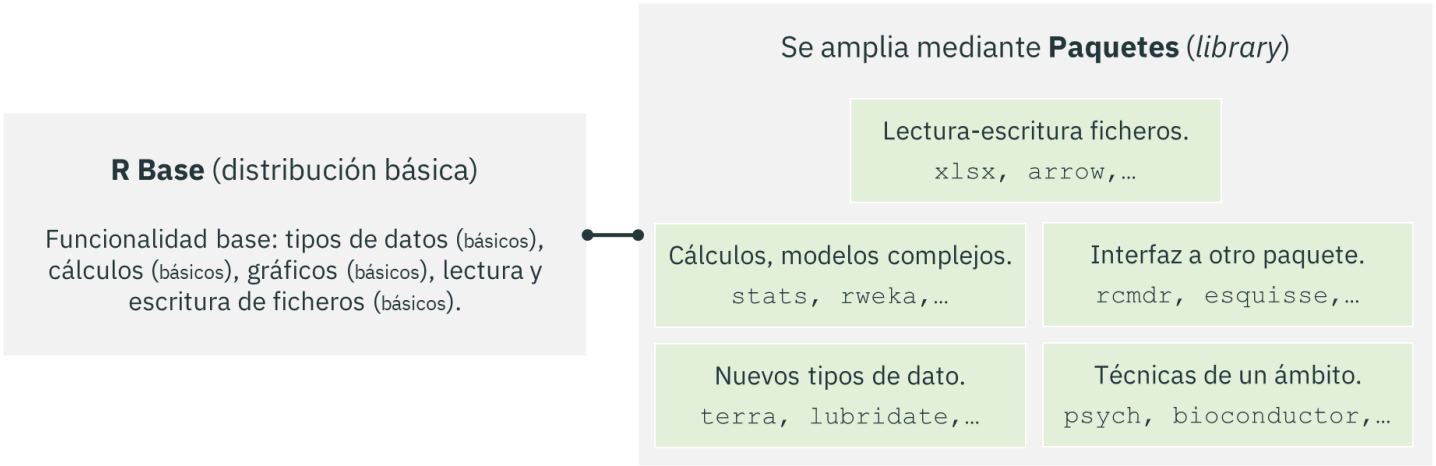
<table>
<tr><td rowspan="2">Scripts
(oculto por defecto)</td><td>Entorno de variables</td></tr>
<tr><td rowspan="2">Multifunción
→ Pestaña “Packages”</td></tr>
<tr><td>Consola de comandos</td></tr>
</table>

Con el botón *Install*, podremos instalar nuevos paquetes introduciendo sus nombres.

Paquetes (*library*)

El repositorio de paquetes más relevante de R es el **CRAN** (gratis, open code, peer-reviewed).

Tiene **+20.000** paquetes que amplían la funcionalidad de R (https://cran.r-project.org/).



ANEXOS

Tema 3: Programación básica con R

Estadística e Inteligencia Artificial con R

ANEXO A: Matrices

Para añadir nuevos elementos a una matriz, utilizamos las funciones `rbind()` y `cbind()`:

```
miMatriz <- matrix(1:20,nrow=4)

# Añadir nuevas filas
nuevaFila<- 1:5
rbind(miMatriz, nuevaFila)

# Añadir nuevas columnas
nuevaColumna<- 1:4
cbind(miMatriz, nuevaColumna)

# Se pueden añadir matrices
rbind(miMatriz, miMatriz)
```

ANEXO A: Matrices

Los operadores matriciales más empleados se realizan con los siguientes comandos de R:

```
miMatriz1 <- matrix(c(0,1,0,1,0,1),nrow=3)
miMatriz2 <- matrix(c(1,1,1,0,0,0,1,1,1),nrow=3)

# Transposicion de matriz (cambiar filas por columnas)
t(miMatriz1)

# Producto matricial
miMatriz1 %*% miMatriz2

# Determinante de una matriz
det(miMatriz1)

# Inversa de una matriz
solve(miMatriz1)
```

ANEXO A: Matrices

Los operadores matriciales más empleados se realizan con los siguientes comandos de R:

```
miMatriz1 <- matrix(c(0,1,0,1,0,1),nrow=3)
miMatriz2 <- matrix(c(1,1,1,0,0,0,1,1,1),nrow=3)
termino <- c(1,2,3)

# Producto vector – matriz
miMatriz1 %*% termino

# Resolucion sistema ecuaciones
solve(miMatriz1, termino)

# Obtencion de autovalores y autovectores
eigen(miMatriz1)
```

Anexo B: Control de flujo y funciones propias

Los **condicionales** nos permiten variar el conjunto de órdenes que se ejecutan en función del valor de una condición. Tienen la siguiente estructura:

```
if(condicion){
      # Ordenes a ejecutar si condicion es verdadera
}else{
      # Ordenes a ejecutar si condicion es falsa
}
```

Ejemplo:

```
notaExamen=6
if(notaExamen<5){
      print("Suspendiste el examen…")
} else{
      print("Felicidades! Has aprobado.")
}
```

Anexo B: Control de flujo y funciones propias

Los **bucles** sirven para repetir una serie de órdenes un número limitado de veces. El primer tipo de bucle son los `for()`, que repiten tantas veces como se indique una operación.

```
for(variable in vector){
      # Ordenes a ejecutar en cada iteracion
      # Pueden usar la variable definida en el for()
}
```

Ejemplo:

```
nDias<-5
beneficios <- c(100, 20, 30, 40, 200)
costes <- c(20, 20, 20, 20, 30)

for (i in 1:nDias) {
      print(beneficios[i]-costes[i])
}
```

Anexo B: Control de flujo y funciones propias

El segundo tipo de **bucle** son los `while()`, que repiten una operación hasta que se deje de cumplir una condición. Es importante que recordéis actualizar la condición en cada iteración para que pare en algún momento y que no quede atrapado en un bucle infinito.

```
while(condicion){
      # Ordenes a ejecutar en cada iteracion
      # Actualizar condicion
}
```

Ejemplo:

```
cont <- 1
while(cont<=10){
      print(cont^2)
      cont<-cont+1
}
```

Anexo B: Control de flujo y funciones propias

Con las funciones provistas por R y las estructuras de control de flujo, podemos definir **funciones propias**. Una función sirve para calcular un valor a partir de una serie de valores de entrada (argumentos). Tiene la siguiente estructura:

```
nombreFuncion <- function(argumentos) {
      valor <- ...# Calculos con los argumentos
      return(valor)
}
```

Ejemplo:

```
helloWorld <- function(name){
      message <- paste("Hola mundo", name, "!")
      return(message)
}
helloWorld("Juan")
```

TEMA 4
Gráficos y visualizaciones con R

Estadística e IA con 

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 4: Gráficos y visualizaciones con R

Índice

Tema 4: Gráficos y visualizaciones con R

❶ Código autogenerado por R Commander

El **uso de comandos** facilita personalizar las visualizaciones o crear gráficos complejos. En esta lección, veremos el código usado por R Commander para crear algunos gráficos.

Conjunto de datos: Covid-19

Ya hemos trabajado con anterioridad con este fichero de datos CSV (**valores separados mediante comas**). En concreto, de cada país, se han tomado las siguientes variables:

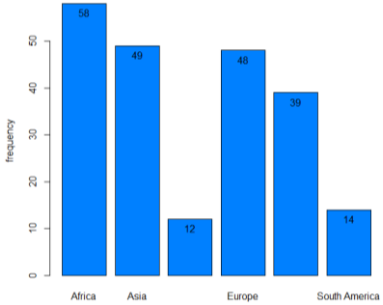
1. Continente al que pertenece el país.
2. Número de casos Covid-19.
3. Número de muertes por Covid-19.
4. Número recuperados de Covid-19.
5. Número de casos activos (a 13/5/21).
6. Número de casos graves.
7. Casos por millón de habitantes.
8. Muertes por millón de habitantes.
9. Número de tests realizados.
10. Tests por cada millón de habitantes.
11. Población del país.

Leemos los datos mediante comandos de R (`read.csv`) o con R Commander (*Datos / Importar datos / desde archivo de texto, URL o portapapeles*). Llamad al data.frame resultante "**Covid**".

Código autogenerado por R Commander

R Commander genera, a parte de las salidas, el código de R que ejecuta por detrás. Repasemos alguno de los gráficos que creamos en el tema 2:

La función `with()` me permite acceder a la variable `continent` sin especificar que se halla en `Covid`.



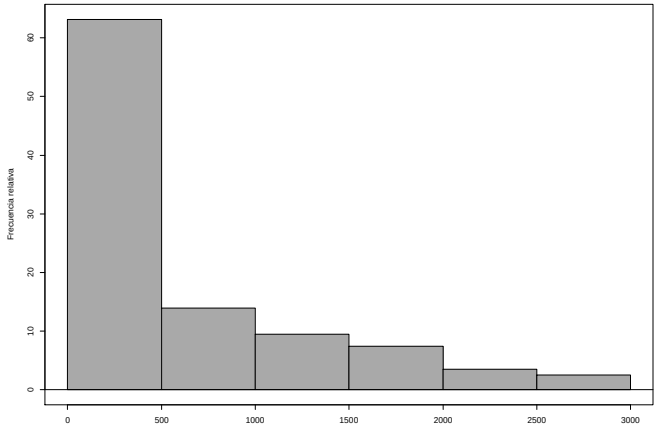
`palette()[2]= "#DF536B"`
Color azul en hexadecimal.

`label.bars=TRUE`
Añade las etiquetas a las barras.

```
with(Covid, Barplot(continent, col=palette()[2], label.bars=TRUE))

# Equivalente, sin usar with:
Barplot(Covid$continent, col=palette()[2], label.bars=TRUE)
```

Código autogenerado por R Commander



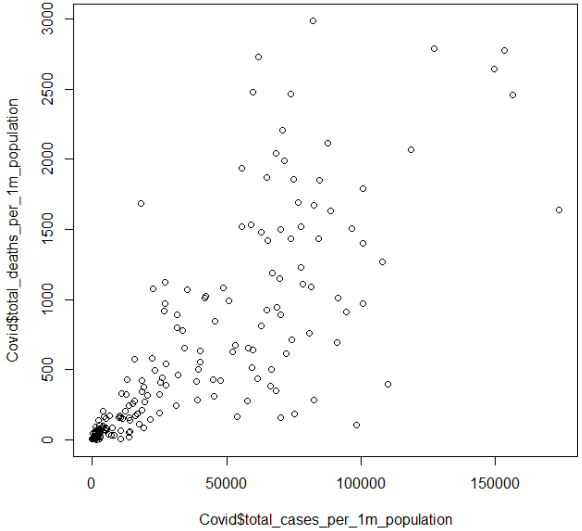

```
Hist(Covid$total_deaths_per_1m_population,
 scale="percent",                          # Frecuencias relativas
 breaks=6, col="darkgray",                 # N° de barras y color
 xlab="Muertes por cada millón de habitantes",
 ylab="Frec. relativa")                    # Etiquetas ejes
```

Tema 4: Gráficos y visualizaciones con R

❷ Algunos comandos gráficos básicos

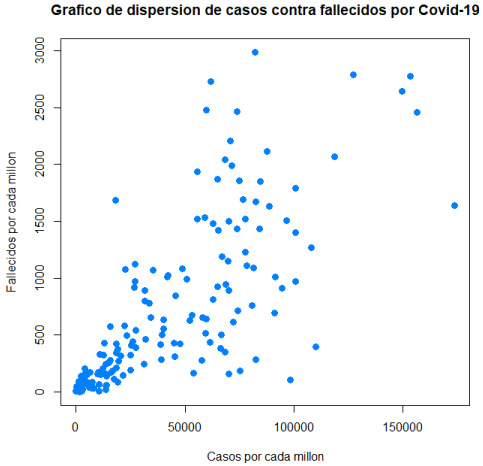
En esta sección, repasaremos algunos comandos básicos para hacer gráficos. Usaremos la función `plot()` para crear gráficos de dispersión. De forma similar a la importación de datos, veremos unas **nociones básicas**.

Gráficos de dispersión




```
plot(Covid$total_cases_per_1m_population, Covid$total_deaths_per_1m_population)
```

Gráficos de dispersión




```
plot(Covid$total_cases_per_1m_population, Covid$total_deaths_per_1m_population,
 pch=19, cex=1.2,      # Puntos: forma (hay 25 formas) y tamaño relativo
 col=2,                # Puntos: color (números, nombre color o código hex.)
 main = "Grafico de dispersion de casos contra fallecidos por Covid-19",
 xlab = "Casos por cada millon", ylab = "Fallecidos por cada millon")
```

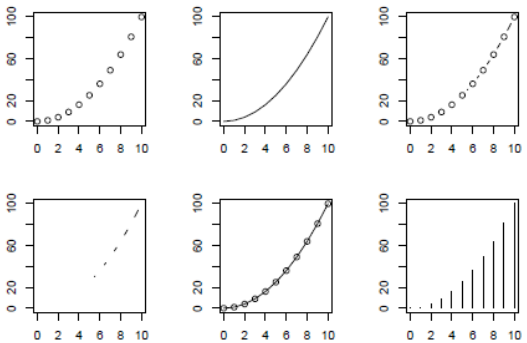
Otras funciones para hacer gráficos

La mayoría de las funciones gráficas de la distribución básica de R tienen una interfaz similar (es decir, tienen argumentos con el mismo nombre que sirven para el mismo propósito).

Como ocurría con la importación de datos, lo mejor es aprender las órdenes según necesidad. Dos lecturas recomendadas son https://r-coder.com/plot-en-r/ y el capítulo 4 del libro "Data Science with R".

Funciones gráficas de alto nivel: `plot, hist, boxplot, pairs`

Otras funciones de interés: `par, legend, expresión, jpeg`



Funciones gráficas de bajo nivel: `points, lines, text,...`

Tema 4: Gráficos y visualizaciones con R

❸ Paquetes gráficos de R

Cada día, más usuarios usan R para crear gráficos científicos, didácticos, infografías,... debido a la (relativa) sencillez del lenguaje y su flexibilidad para crear todo tipo de gráficos.

Para ello, se suele recurrir a utilizar **paquetes gráficos**.

Paquetes gráficos de R

De momento, (casi) todos los gráficos hechos se han creado con la distribución básica de R. Sin embargo, otros paquetes permiten crear gráficos personalizables más complejos con comandos.

Algunos de los paquetes gráficos más populares son los siguientes:

`lattice`	`ggplot2`	`plotly`
Paquete gráfico más veterano.	Paquete gráfico más usado.	Paquete gráfico reciente.
✗ Únicamente en R.	✓ Disponible en varios lenguajes.	✓ Disponible en varios lenguajes.
✗ "Rígido", similar a base.	✓ Muchas extensiones.	✓ Gráficos en 3D, animados, ...
✓ Sintaxis más sencilla.	✗ Cierta curva de aprendizaje.	✗ Cierta curva de aprendizaje.
https://lattice.r-forge.r-project.org/	https://ggplot2.tidyverse.org/	https://plotly.com/r/

Paquetes gráficos de R

Muchos usuarios (como tú) crean y publican paquetes que extienden la funcionalidad de `ggplot2`:

- Aumentar los tipos de gráficos disponibles (por ejemplo, radarplots o gráficos de redes).
- Personalizar gráficos (representar los puntos de mi gráfico, en vez de con ●, con ).
- Crear gráficos compuestos (combinar varios gráficos en el mismo lienzo).
- Dar animaciones a los elementos visuales del gráfico.
- Facilitar su uso, en concreto, permitir su uso mediante ventanas.

El paquete `esquisse` es la extensión de `ggplot2` que permite su uso mediante ventanas.

Esquisse: Instalación del paquete

Las siguientes instrucciones sirven para la **instalación de cualquier paquete**. Bastará con sustituir la palabra "`esquisse`" con el nombre del paquete que se desee instalar.

1. Instala el paquete (Nota: esta parte solo es necesaria la primera vez que se usa un paquete, a no ser que se quiera actualizar la versión del mismo).
   A. Ejecutando el siguiente comando: `install.packages("esquisse")`
   B. Mediante la interfaz de RStudio, panel inferior derecha: *Packages / Install*: "`esquisse`".
2. Para cargar el paquete, una vez instalado, usamos la función `library()`:

```
library("esquisse")
```

Esquisse

Es un paquete que permite crear gráficos de forma sencilla e intuitiva, facilitando la exploración de los datos y sirviendo como herramienta didáctica de `ggplot2`.

El último paso que nos resta para trabajar con `esquisse` es cargar el conjunto de datos en el paquete. Como se ha comentado previamente, usaremos el conjunto de datos "`Covid`" en esta lección. Con el siguiente comando, se cargará `esquisse` en nuestro navegador:

```
esquisser(Covid, viewer = "browser") # Covid: nombre del data.frame empleado
```

> Comprueba que el primer argumento (`Covid`) coincida con el nombre dado a los datos.

> Si tienes problemas con el navegador, usa el comando `esquisser(Covid)`

TEMA 5
Inferencia estadística básica

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 5: Inferencia estadística básica

Índice

1. Introducción
2. Distribuciones estadísticas: Normal y t de Student
3. Intervalos de confianza y contrastes de hipótesis
   A. Media de una población normal
   B. Poblaciones normales pareadas: Diferencia de medias

Tema 5: Inferencia estadística básica

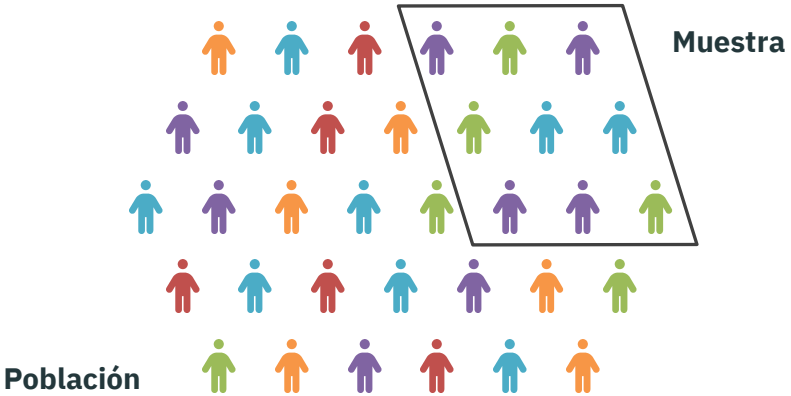
1 Introducción

En cualquier estudio, se cuenta con la información, limitada, provista por una **muestra**. La inferencia estadística busca inferir el comportamiento (modelo) de la **población** completa a partir de la información que nos proporciona la muestra.

Estadística inferencial

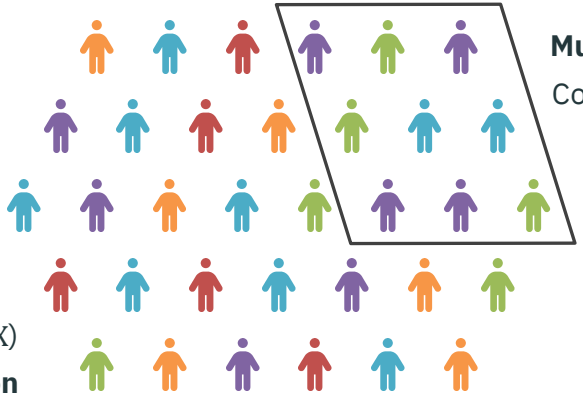
¿Para qué necesito la inferencia si es fácil estimar medidas (media, varianza,...) con una muestra?

La composición de las muestras es intrínsecamente aleatoria: su variabilidad puede representar con mayor o menor fidelidad la verdadera distribución de la población completa.



Estadística inferencial

Objetivo: Conocer la distribución de la población (P_X)



Muestra $(X_1, \ldots, X_n)$
Contienen información sobre P_X

Variable de interés (X)

Población

- ¿Qué distribución siguen los datos? ¿Siguen una distribución exponencial o una normal?
- ¿Cuáles son los valores de la media o la varianza a nivel poblacional?
- ¿En qué intervalo estará, con un 90% de probabilidad, el verdadero valor de la media?
- ¿Tenemos evidencias para afirmar que la media sea mayor que 0?

Estadística inferencial

- **Estadístico**: cualquier función calculada a partir de una muestra. Nos dan información sobre la distribución de la población. Ejemplos: $S_1 = \sum_{i=1}^{n} X_i$, $S_2 = \sum_{i=1}^{n} {X_i}^2$, …
- **Estimador**: un parámetro desconocido θ se estima por un estadístico $\hat{\theta} = f(X_1, \ldots, X_n)$.
  Se dice que $\hat{\theta}$ es estimador de θ.

Parámetro	Estimador
μ Media poblacional	$\hat{\mu} = \overline{X} = \frac{S}{n} \sum_{i=1}^{n} X_i$ Media muestral
σ^2 Varianza poblacional	$\hat{\sigma}^2 = S^2 = \frac{S}{\bullet} \sum_{i=1}^{n} (X_i - \overline{X})^2$ Varianza muestral

Estadística inferencial

Ejemplo 1: Elecciones USA 2016

Las encuestas daban la victoria a Hillary Clinton, cuando el ganador fue Donald Trump.

Con mi encuesta (muestra), ¿Basta con un "pronóstico" (gana Clinton), o son necesarios una estimación puntual (60% para Clinton) o una en intervalo ($60\%\pm30$ para Clinton)?

¿Era representativa la muestra? ¿Había votantes de diferentes estados, edades, culturas...?

Ejemplo 2: Alimentador de un ordenador

Se quiere estudiar si el alimentador funciona correctamente con una muestra:

$$X_1, \ldots, X_{15} \qquad \overline{X} = 7.5 \qquad S^2 = 1$$

El ordenador puede funcionar mal si la corriente que recibe es inferior a 7 voltios o si la variabilidad es superior a 1.1. ¿Hay indicios de que esté recibiendo un voltaje inadecuado?

Tema 5: Inferencia estadística básica

❷ Distribuciones estadísticas

En el ámbito de la inferencia, hay dos distribuciones que se usan principalmente para estudiar la media:

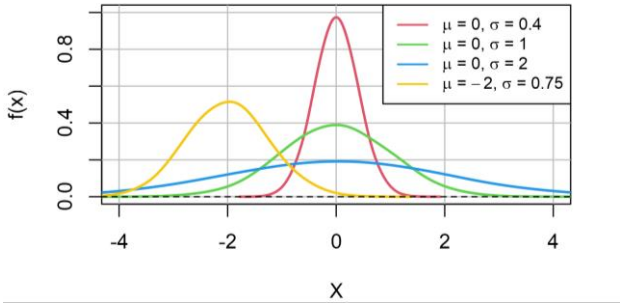
A. Distribución **normal**

B. Distribución t de Student

Distribución normal

Distribución (continua) asociada a variables que miden características que **toman valores alrededor de μ**, variando estas cantidades aproximadamente σ.

$$X \sim N(\mu, \sigma^2), \qquad \mu \in R, \sigma > 0$$



Función de probabilidad o densidad	$P(X = x) = f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	
Función de probabilidad acumulada o distribución	$P(X \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt.$	
Propiedades	Media $= \mu$ Mediana $=$ Moda $= \mu$	Varianza $= \sigma^2$ Desviación Estándar $= \sigma$

Distribución normal

Las funciones relativas a la normal son las siguientes en R:

- $X \sim N(2,3)$, ¿ $P(X \leq 2)$? Respuesta: 0.5

```
pnorm(2,mean=2,sd=3)  # Función de distribución
```

- $X \sim N(0,1)$, ¿ x tal que $P(X \leq x) = 0.975$? Respuesta: 1.959964

```
qnorm(0.975,0,1) # Función inversa a la distribución. Cuantil
```

- $X \sim N(0,1)$, ¿ $P(X = 0)$? Respuesta: 0.3989423

```
dnorm(0,0,1)     # Función de densidad
```

- Generar una muestra de 8 observaciones provenientes de una población $N(10,1)$.

```
muestra1 <- rnorm(8,mean=10,sd=1) # Genera muestras aleatorias normales
```

Distribución normal

- Representa una muestra normal. Cuanto mayor sea el tamaño muestral, primer argumento de la función `rnorm()`, más se asemejará el aspecto al de la distribución teórica de la normal.

```
pnorm(2,mean=2,sd=3)  # Función de distribución
```

- Para profundizar en el papel de μ y σ, podemos representar varias muestras de normales.

```
muestra1<-rnorm(10000,mean=0,sd=0.4)
muestra2<-rnorm(10000,mean=0,sd=1)
muestra3<-rnorm(10000,mean=0,sd=2)
muestra4<-rnorm(10000,mean=-2,sd=0.75)

plot(0,type="n",xlim=c(-4,4), ylim=c(0,1), xlab = NA, ylab = NA)
lines(density(muestra1), col=2, lwd=2) # Para poner líneas en el grafico
lines(density(muestra2), col=3, lwd=2) # col : color, lwd : grosor linea
lines(density(muestra3), col=4, lwd=2)
lines(density(muestra4), col=7, lwd=2)
```

`plot(type="n"…)`
Inicializa el área gráfica en blanco.

Distribución normal

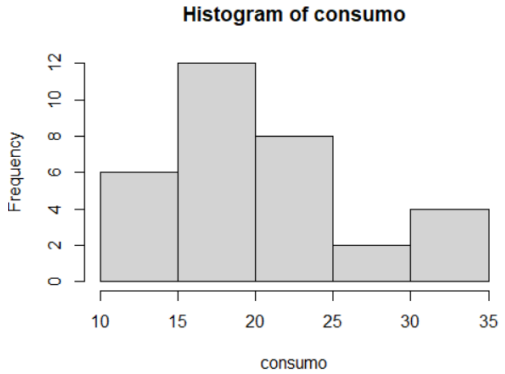
Diversos procedimientos y modelos tienen como hipótesis inicial la normalidad de las variables.

Como ejemplo, estudiaremos la normalidad de la variable consumo del conjunto de datos `mtcars`.

```
data(mtcars)
consumo<-mtcars$mpg
```

De forma preliminar, podemos estudiar visualmente su distribución:

```
hist(consumo)
```



A favor: la moda (valor con más frecuencia observada), media y mediana toman valores cercanos.

En contra: la distribución no es totalmente simétrica (cola derecha más pesada).

Distribución normal

Para saber si una variable sigue una distribución normal, se usan **tests de normalidad**.

```
shapiro.test(consumo)
```

```
        Shapiro-Wilk normality
testdata:  consumo = 0.94756,
           p-value = 0.1229
```

$p - valor > 0.10$: asumimos normalidad.
$0.1 > p - valor > 0.01$: dudas.
$p - valor < 0.01$: rechazamos normalidad.

R Commander provee varios tests de normalidad:

```
library(Rcmdr)
```

R Commander		
Estadísticos ▶	Gráficas	(...)
Resúmenes ▶		Conjunto de datos activo
Tablas de contingencia		(...)
Medias		**Test de normalidad...**
(...)		Transformar para normalizar...

Distribución normal

Los más empleados son Shapiro-Wilk, Cramer-von Mises y Shapiro-Francia.

```
Shapiro-Wilk test:  mpg
W = 0.947, p-value = 0.1229
```

```
Cramer - von Mises test:  mpg
W = 0.088, p    - value = 0.1558
```

```
Shapiro-Francia test:  mpg
W = 0.952, p-value = 0.1495
```

Los estimadores de los parámetros del modelo son `mean(consumo)` y `sd(consumo)`.

Se suelen realizar varios tests en situación de duda (como $0.1 > \text{p} - \text{valor} > 0.01$).

En función de las características de las muestras, algunos tests son más adecuados.

Distribución normal

Muchos procesos naturales, biológicos, productivos, ... siguen una distribución normal.

Alturas de individuos	Presión arterial	Errores de medición
Tiempo para ir de un punto geográfico a otro	Calificaciones de una clase	**Distribución de la media muestral**

Aunque X no sea normal, si n es grande, entonces $\bar{X} \rightarrow N(\mu, \frac{\sigma}{\sqrt{n}})$ por el TCL.

→ **Teorema Central del Límite**: con n grande, X muy probablemente tomará un valor próximo a μ.

Distribución t de Student

La aproximación por el TCL requiere conocer la desviación estándar poblacional: $\bar{X} \to N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$. Normalmente solo conoceremos su estimador muestral S, lo que nos lleva a la distribución t:

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \to t_{n-1}$$

Se trata de una distribución continua y simétrica que depende de un parámetro: **los grados de libertad**. Se usa en muestras pequeñas, ya que a partir de $n > 30$, $t_n \rightsquigarrow N(0,1)$.

Propiedades	Media $= \mu$	Varianza $= \sigma^2$
	Mediana $=$ Moda $= \mu$	Desviación Estándar $= \sigma$
Funciones en R	`pt` (distribución), `qt` (cuantil)	
	`dt` (densidad), `rt` (muestra aleatoria).	

Tema 5: Inferencia estadística básica

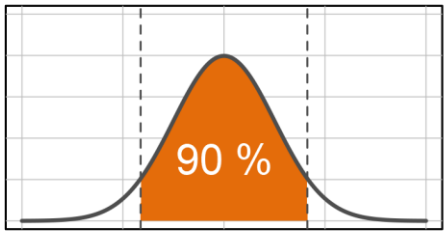
❸ Int. de confianza y contrastes de hipótesis

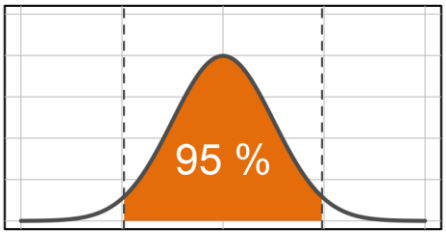
Las estimaciones puntuales resultan, a menudo, insuficientes: la **probabilidad** de que el **estimador** (muestral) tome el valor (poblacional), exactamente, es muy **baja**. De esto, surge la necesidad de las estimaciones por intervalo y los contrastes.

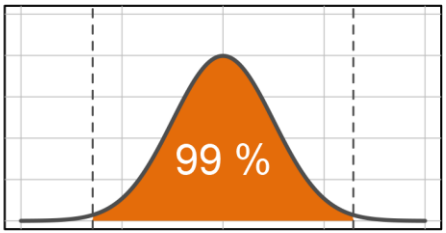
Estimación por intervalos

Un **intervalo de confianza** γ % contiene al verdadero valor del parámetro con una confianza ("probabilidad") del γ %. A menudo, se formula γ para que dependa del error: $\gamma = 100(1-\alpha)\%$.

Cuánto más confianza deseemos en la estimación, más amplio será el rango de valor. Por otro lado, altos tamaños muestrales suelen reducir la amplitud de los ICs.







Contrastes de hipótesis

Permiten estimar si un parámetro es igual, superior o inferior a un determinado valor con un error asumido α en base a unas hipótesis nula y alternativa. Pueden ser uni- o bilaterales.

Hipótesis nula H_0

Asumimos inicialmente como valida. Suele ser la más sencilla, además de simplificar el modelo.

Hipótesis alternativa H_1

Solo la asumiremos si las evidencias de los datos están con suficientemente en contra de H_0.

Bilateral

$$\begin{cases} H_0: \theta = \theta_0 \\ H_1: \theta \neq \theta_0 \end{cases}$$

Unilateral

$$\begin{cases} H_0: \theta \leq \theta_0 \\ H_1: \theta > \theta_0 \end{cases}$$

Dado un estimador de un parámetro (que asumimos que sigue una cierta distribución), el **p-valor** asociado a un contraste de hipótesis es la probabilidad de que la hipótesis nula sea cierta (es decir, que el verdadero valor del parámetro sea el que constata H_0).

Media de una población normal

Partimos de $T = \frac{\bar{X}-\mu}{S/\sqrt{n}} \rightarrow t_{n-1}$. El **intervalo de confianza de la media de una población normal** que contiene al verdadero valor con una confianza del $100(1-\alpha)\%$ se define como:

$$P\left(t_{n-1,1-\alpha/2} \leq \frac{\bar{X}-\mu}{S/\sqrt{n}} \leq t_{n-1,\alpha/2}\right) = P\left(-t_{n-1,\alpha/2} \leq \frac{\bar{X}-\mu}{S/\sqrt{n}} \leq t_{n-1,\alpha/2}\right) = 1-\alpha$$

Si despejamos el parámetro μ: $\mu \in \bar{X} \pm t_{n-1,\alpha/2}\frac{S}{\sqrt{n}}$

$t_{n-1,\alpha/2} \Rightarrow$ Cuantil,
$X \sim t_{n-1}$, ¿ x tal que $P(X \leq x) = \alpha/2$?

Los siguientes comandos permiten calcular un I.C. para la media del consumo:

```
nObs<-32; alfa<-0.05
cuantil<-qt(1-(alfa/2), df=nObs-1)
mean(consumo)-cuantil*(sd(consumo)/sqrt(nObs))
mean(consumo)+cuantil*(sd(consumo)/sqrt(nObs))
# Al ser n suficiente, se podría aproximar a la normal -usar qnorm-)
```

Media de una población normal

Un **contraste de hipótesis de la media de una población normal** se calcula a partir del estadístico T bajo la hipótesis nula. Para hacer el contraste, debemos establecer el valor crítico μ_0, el error α y el tipo de contraste que, a su vez, varía el cálculo del p-valor.

$$T_0 = \frac{\bar{X}-\mu_0}{S/\sqrt{n}} \rightarrow t_{n-1}$$

Unilateral $\begin{cases} H_0: \mu \geq \mu_0 \\ H_1: \mu < \mu_0 \end{cases}$ $p - valor = P(T < T_0)$

Bilateral $\begin{cases} H_0: \mu = \mu_0 \\ H_1: \mu \neq \mu_0 \end{cases}$ $p - valor = 2P(T < T_0)$

Si queremos contrastar $H_0: \mu = 21$ vs. $H_1: \mu \neq 21$ para la variable consumo, usamos estas órdenes:

```
mu0=21
t0<-(mean(consumo)-mu0)/(sd(consumo)/sqrt(nObs))
pvalor<-2*pt(t0, df=nObs-1) # Bilateral
# pvalor<-1-pt(t0, df=nObs-1) # Para el unilateral
# Al ser n suficiente, se podría aproximar a la normal -usar qnorm-)
```

```
p-value = 0.3999
La media poblacional podría ser 21.
```

Media de una población normal

R Commander ofrece un interfaz amigable y sencillo para calcular intervalos de confianza y contrastes de hipótesis sobre la media de una población normal:

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia					
	Medias ▶		**Test t para una muestra….**			
	Proporciones		Test t para muestras independientes…			
	(…)		(…)			

En la ventana emergente, seleccionamos la variable `mpg`.

Normales pareadas: Diferencia de medias

Otro caso interesante derivado del anterior es el estudio de la **diferencia de medias en muestras pareadas**. Suponiendo que el tamaño de las muestras es el mismo y que su desviación estándar es la misma y común, el resultado principal es el siguiente:

$$\frac{\overline{X}_1 - \overline{X}_2 - (\mu_1 - \mu_2)}{\frac{S_1^2 + S_2^2}{2}\sqrt{\frac{2}{n}}} \to t_{2n-2}$$

$$\begin{cases} H_0: \mu_1 = \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases} \qquad \begin{cases} H_0: \mu_1 - \mu_2 = 0 \\ H_1: \mu_1 - \mu_2 \neq 0 \end{cases}$$

(Sin las suposiciones hechas, varía la expresión del estadístico y su distribución)

Se usa, por ejemplo, para estudiar el efecto de tratamiento, tomando medidas antes y después de aplicarlo. Veamos un ejemplo en el que calcularemos intervalos de confianza y contrastes.

Normales pareadas: Diferencia de medias

El conjunto de datos `sleep` contiene el incremento en horas de sueño de 10 pacientes usando un placebo (control, grupo 1) y un tratamiento somnífero (grupo 2). Queremos conocer si hay diferencias entre los dos grupos de pacientes.

Para realizar fácilmente los cálculos, crearemos un `data.frame` con dos columnas:

```
data("sleep")
sleepData<-data.frame("placebo"=sleep$extra[sleep$group == 1],
                      "medicamento"=sleep$extra[sleep$group == 2])
```

Por último, el intervalo de confianza y el contraste se pueden calcular mediante `t.test()`:

```
t.test(sleepData$medicamento, sleepData$placebo, paired = TRUE)
```

Normales pareadas: Diferencia de medias

```
Paired t-test. data:  sleepData$medicamento and sleepData$placebo
t = 4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval: 0.7001142 2.4598858
sample estimates (mean of the differences): 1.58
```

R Commander también permite acceder a este procedimiento mediante su interfaz.

R Commander						
Datos	**Estadísticos ▶**	Gráficas	Modelos	Distribuciones	Herramientas	Ayuda
	Resúmenes					
	Tablas de contingencia					
	Medias ▶		(...)			
	Proporciones		**Test t para datos relacionados...**			
	(...)		(...)			

Conclusión

Hemos visto a lo largo de este tema diversos métodos básicos de inferencia. Se trata de un campo de la estadística muy amplio. Existen otros muchos casos de estudio de interés:

- Contrastes sobre la varianza de una población normal.
- Estudio de 2 poblaciones normales independientes.
- Estudio de proporciones y diferencias entre dos proporciones.
- Ajuste de datos a otras distribuciones de probabilidad.

En R, muchos paquetes facilitan la aplicación de las técnicas de inferencia estadística.

Extensiones de `ggplot2`:

`ggstatsplot`

Paquetes sobre inferencia:

`statsr`

Extensiones de R Commander:

`RcmdrPlugin.UCA`

TEMA 6
Fundamentos de Inteligencia Artificial

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Tema 6: Fundamentos de IA

Índice

1. IA: Machine Learning.
2. Aprendizaje supervisado: Clasificación y Regresión.
3. Otras técnicas de ML: PCA y Clustering.
4. Redes Neuronales.

Tema 6: Fundamentos de IA

1 IA: Machine Learning

A ¿Qué es la **inteligencia artificial**?

B ¿Qué es el aprendizaje automático o **Machine Learning**?

¿Qué problemas reales se han resuelto con estas técnicas?

Inteligencia Artificial (IA)

Disciplina perteneciente a las **ciencias de la computación** que persigue que las máquinas sean capaces de imitar funciones cognitivas humanas, como reconocer patrones en imágenes, audio y vídeo, o comprender y responder al lenguaje natural.

Es un **campo interdisciplinar** que aúna conocimientos de muchas disciplinas, como informática, estadística, matemáticas, telecomunicaciones, biología, neurología,...

Los primeros desarrollos se dieron en la década de 1950 por científicos como A. **Turing**.
Se consolido como disciplina entre 1970-1990, mientras que su reciente auge se debe a la mejora de los **recursos computacionales** y la acumulación de **grandes volúmenes de datos**.

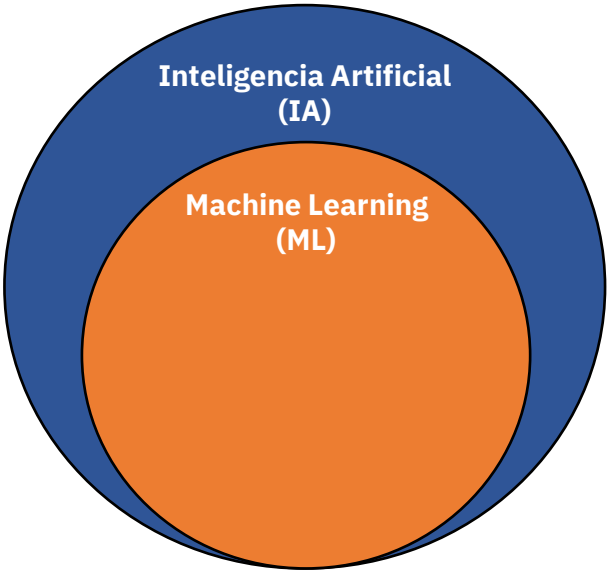
Machine Learning (ML)

Aprendizaje Automático (*Machine Learning*)

Disciplina que busca patrones recurrentes en los datos, haciendo uso de modelos.

- Los **modelos** son formulaciones matemáticas que se ajustan a los datos a través de **algoritmos**.
- También recibe el nombre de *Statistical Learning*.

El **propósito** de las técnicas permite definir tipos de ML.

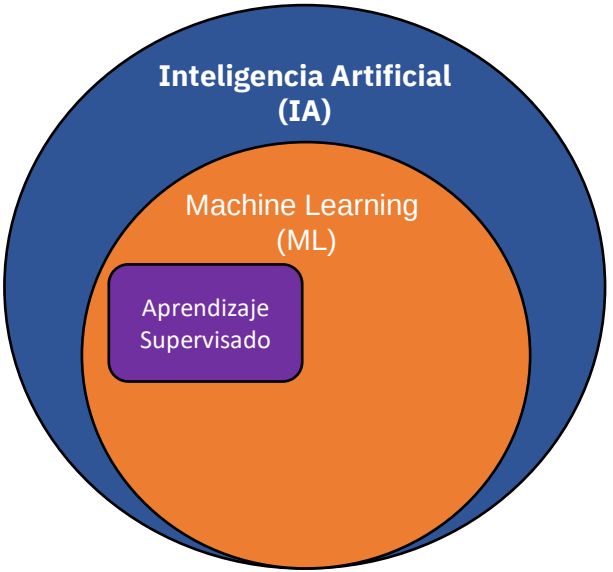


Aprendizaje supervisado

Aprendizaje Supervisado (*Supervised Learning*)

Propósito: predecir una variable observada a partir de los valores de otras variables relacionadas.

- Busco un modelo (expresión matemática), que relacione las variables independientes con la dependiente.
- Con un modelo adecuado, podré estimar el valor de la variable predicha con tan solo medir las otras.



Aprendizaje supervisado

Clasificación (Variable predicha es **categórica**)

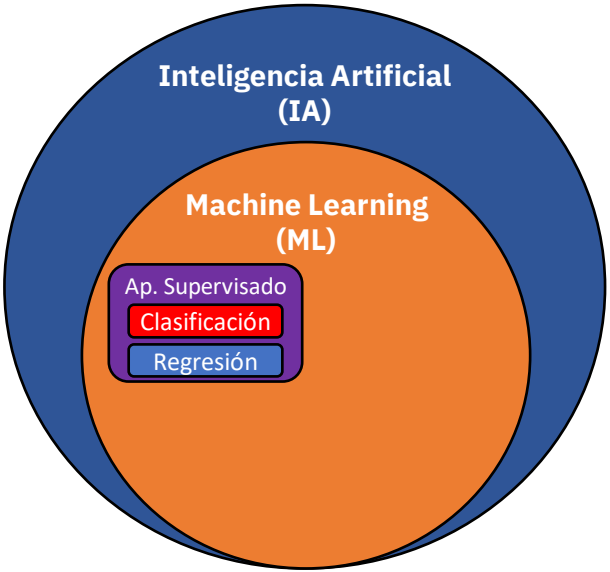
Buscar patrones para distinguir diferentes tipos de observaciones, marcados por la variable respuesta.

Emails	**Chequeo médico**	**Imágenes**
¿Mensaje o spam?	¿Sano o enfermo?	¿Perro, gato, ratón?

Regresión (Variable predicha es **numérica**)

Predecir una variable respuesta relacionándola con otras variables independientes (como en regresión lineal).

Marketing	**Inmobiliarias**	**Epidemiología**
¿Beneficios?	¿Precio vivienda?	¿Contagios/semana?



Tema 6: Fundamentos de IA

❷ Apren. Supervisado: Clasificación y Regresión

Esta sección detalla los modelos más usados de aprendizaje supervisado y su implementación en R. Además, se resolverá un problema de **clasificación** y otro de **regresión**.

Clasificación

Clasificación (Variable predicha es **categórica**) → Binaria / Multiclase

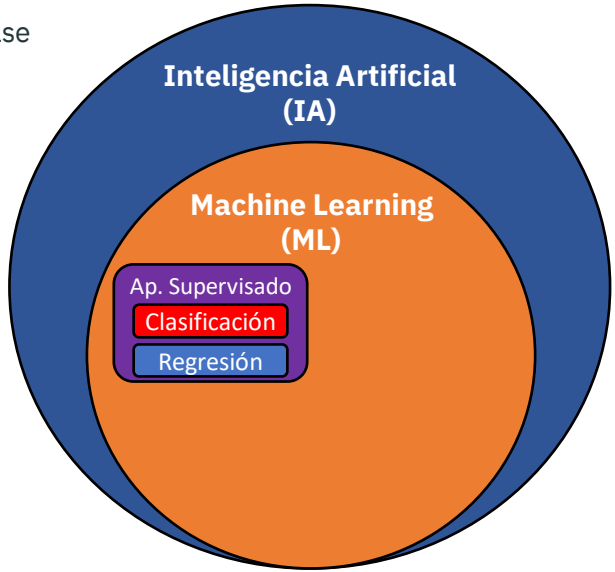
Buscar patrones para distinguir diferentes tipos de observaciones, marcados por la variable respuesta.

Emails	**Chequeo médico**	**Imágenes**
¿Mensaje o spam?	¿Sano o enfermo?	¿Perro, gato, ratón?

Regresión (Variable predicha es **numérica**)

Predecir una variable respuesta relacionándola con otras variables independientes (como en regresión lineal).

Marketing	**Inmobiliarias**	**Epidemiología**
¿Beneficios?	¿Precio vivienda?	¿Contagios/semana?



Conjunto de datos: Pima Indians Diabetes

Con los siguientes comandos cargaremos el conjunto de datos:

```
install.packages("mlbench")
# Necesario solo la primera vez
```

```
library(mlbench)
data(PimaIndiansDiabetes)
```

PIMA INDIANS DIABETES

¿Se puede saber si un individuo indio Pima sufre diabetes con una serie de mediciones sencillas?

Variable dependiente (respuesta o a predecir)	▪ Diabetes: ¿Tiene diabetes?	
Variables independientes (predictores, características, *features*)	1. Pregnant: n.º de embarazos. 2. Glucose: glucosa en sangre. 3. Pressure: presión sanguínea. 4. Triceps: grosor del triceps.	5. Insulin: medición de insulina. 6. Mass: índice de masa corporal. 7. Pedigree: relativo al nº ancestros diabéticos. 8. Age: edad.

Conjunto de datos: Pima Indians Diabetes

1. **Escalado** (o normalización) de los **predictores**.

$$X \qquad E(X) = \mu \quad Var(X) = \sigma^2$$

$$X' = \frac{X - \mu}{\sigma} \qquad E(X') = 0 \quad Var(X') = 1$$

Fundamental cuando tienen unidades de medición (y/o distribuciones) muy diferentes. De no realizarse, se dará más importancia a las que tengan más variabilidad.

```
pimaMedias<-apply(PimaIndiansDiabetes[,-9], 2, mean)
pimaDesvEst<-apply(PimaIndiansDiabetes[,-9], 2, sd)

PimaIndiansDiabetes[,-9]<-scale(PimaIndiansDiabetes[,-9])
```

2. Re-etiquetado de la variable respuesta (`diabetes`). Transformamos {`neg`, `pos`} → {`0`,`1`}.

```
PimaIndiansDiabetes$diabetes<-factor(ifelse(PimaIndiansDiabetes$diabetes=="pos",1,0))
```

Paquete de ML: RWeka

Interfaz en R de Weka, plataforma de Machine Learning y minería de datos con infinidad de modelos desarrollada en Java por la Universidad de Waikato.

Se trata de un paquete sencillo de usar para los usuarios (“amigable”) en el que la interfaz de los modelos es común. Su manual os puede servir para expandir vuestros conocimientos.

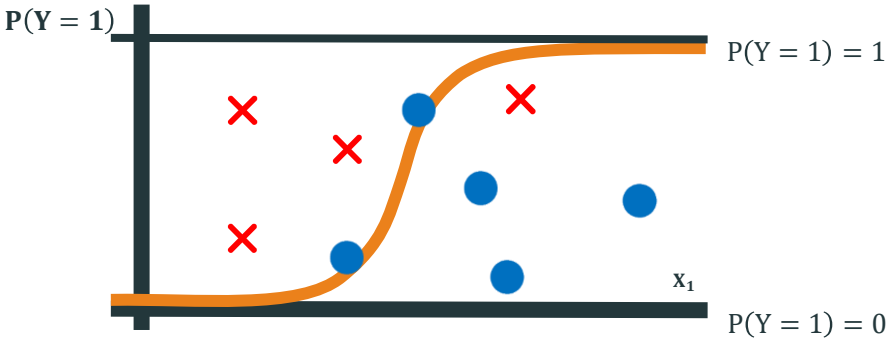
```
install.packages("RWeka") # Necesario solo la primera vez
library(RWeka)
```

Si salta un **error** en la instalación...

1) Instala el paquete `rJava`: `install.packages("rJava")`
2) ¿Al volver a ejecutar `library(RWeka)` salta un error relativo a `JAVA_HOME`?
   Instala Java, disponible en https://www.java.com/es/download/

Clasificación: Regresión logística

Modelo que busca predecir la probabilidad de que una observación sea de una clase mediante una función lineal de los valores de las variables predictoras.



¿Un modelo de clasificación llamado "regresión" logística? Su nombre deriva de que se predice un valor numérico, en concreto, $p = P(Y = 1|X_1 = x_1, \ldots, X_p = x_p)$. El modelo tiene la siguiente formulación: $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

Clasificación: Regresión logística

Los siguientes comandos ajustan una regresión logística de una variable (`insulin`):

```
regLogistica<-Logistic(diabetes~insulin, data = PimaIndiansDiabetes)
summary(regLogistica)
regLogistica # Parámetros del modelo
```

=== Summary ===

Correctly Classified Instances	505	65.7552 %
Incorrectly Classified Instances	263	34.2448 %

(...)

=== Confusion Matrix ===

a	b	<-- classified as
490	10	a = neg
253	15	b = pos

Se aciertan el **65.7 %** de los diagnósticos.

Matriz de confusión

		Predicho: neg	Predicho: pos
Clase real	neg	490	10
	pos	253	15

Le cuesta identificar a los diabéticos.

Clasificación: Regresión logística

¿Podríamos mejorar la predicción empleando en el modelo más variables que la insulina?

```
regLogistica<-Logistic(diabetes~., data = PimaIndiansDiabetes)
summary(regLogistica)
```

=== Summary ===

Correctly Classified Instances	601	78.2552 %
Incorrectly Classified Instances	167	21.7448 %

(...)

=== Confusion Matrix ===

a	b	<-- classified as
445	55	a = neg
112	156	b = pos

Especificación de fórmulas en modelos

`.` → Resto de variables en el dataset.

`insulin + glucose` → Dos variables.

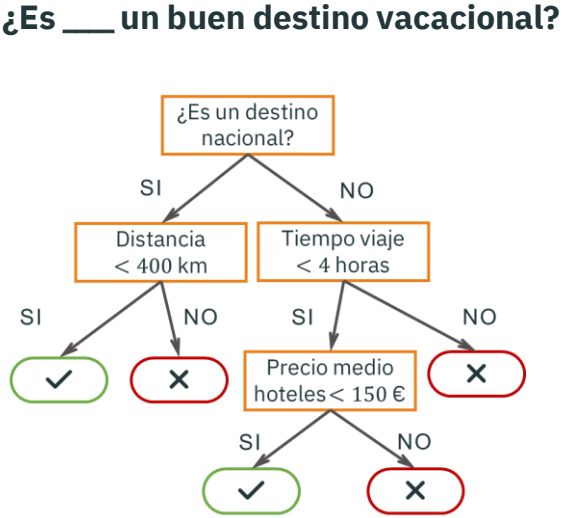
Ha mejorado el acierto más de un 10 %. La mejora en diabéticos es notable (incorrectos menos del 50%).

Clasificación: Árbol de decisión

Modelos de clasificación en los que se establecen una serie de reglas basadas en los predictores para determinar la clase de una observación.

Nodos de división (o internos) **y hoja** (o terminales) se disponen en diferentes niveles de **profundidad**:

- En los primeros, se selecciona una variable, discreta o discretizada, y un umbral de decisión.
- En los nodos hojas, se da una predicción en forma de clase o probabilidad de ser de una clase.



Clasificación: Árbol de decisión

Si creamos un árbol de decisión, ¿se predecirá mejor que con la regresión logística?

```
arbolDecision<-J48(diabetes~., data = PimaIndiansDiabetes)
summary(arbolDecision)
```

=== Summary ===

Correctly Classified Instances	646	84.1146 %
Incorrectly Classified Instances	122	15.8854 %

(...)

=== Confusion Matrix ===

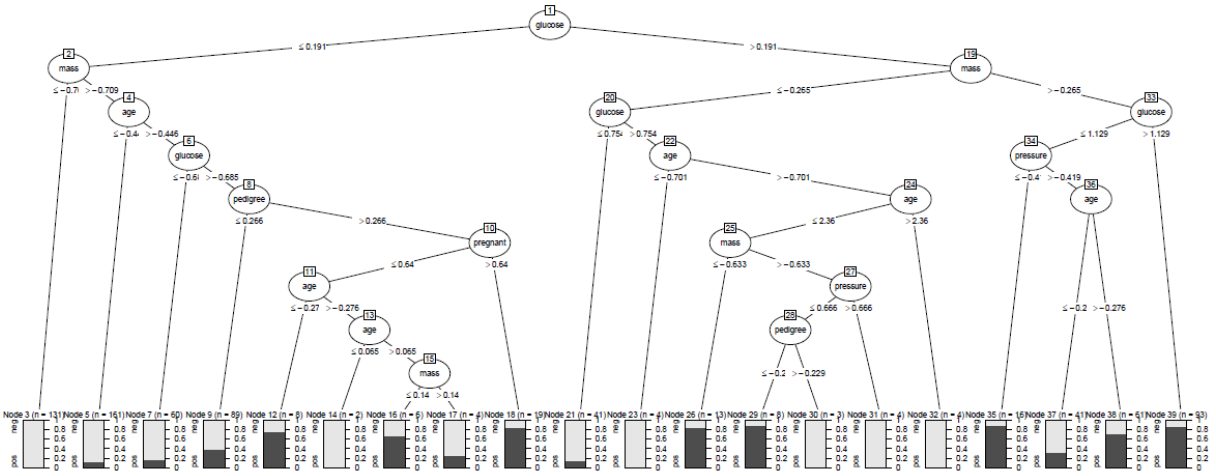
a	b	<-- classified as
468	32	a = neg
90	178	b = pos

Acierta más que la regresión logística (78 %). Erra en menos del 35 % de las personas diabéticas.

Puede ser necesario: `install.packages("partykit")`

Para visualizar el árbol de decisión, debemos ejecutar `plot(arbolDecision)`

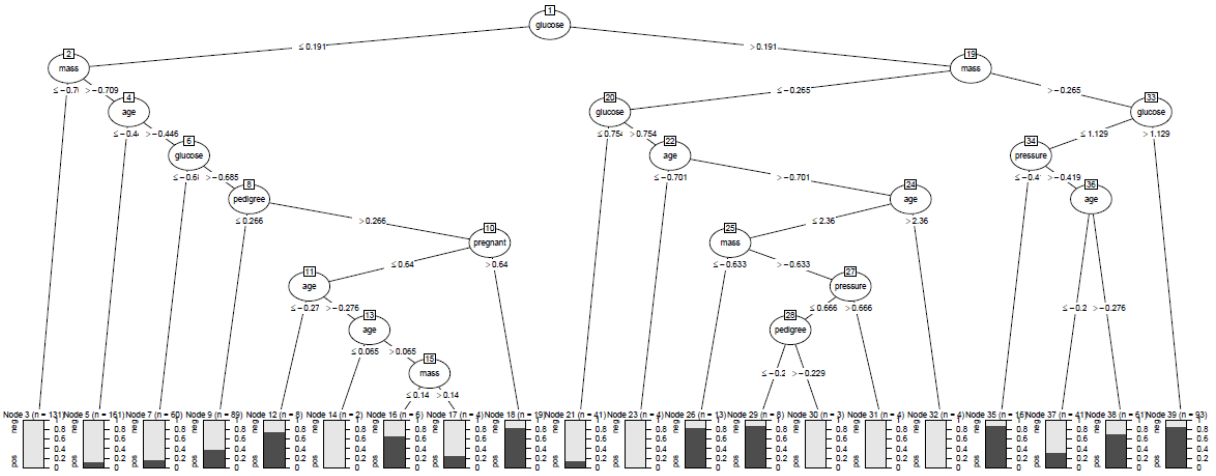
Clasificación: Árbol de decisión



Nos han enviado los datos de otro individuo de Pima que no estaba en los datos, ¿Podría ser diabético?

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
Valores	0	100	100	20	120	60	1	40	¿?

Clasificación: Árbol de decisión



Nos han enviado los datos de otro individuo de Pima que no estaba en los datos, ¿Podría ser diabético?

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
Valores	0	100	100	20	120	60	1	40	¿?
V. escalados	-1.14	-0.65	1.59	-0.033	0.34	3.55	1.59	0.57	¿?

Clasificación: Árbol de decisión

Nos han enviado los datos de otro individuo de Pima que no estaba en los datos, ¿Podría ser diabético?

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
Valores	0	100	100	20	120	60	1	40	¿?
V. escalados	-1.14	-0.65	1.59	-0.033	0.34	3.55	1.59	0.57	¿?

Para calcular la probabilidad de ser positivo con R, podemos usar los siguientes comandos:

```
obsN<-data.frame(pregnant=0, glucose=100, pressure=100, triceps=20,
                 insulin=120, mass=60, pedigree=1, age=40)
obsN <-(obsN-pimaMedias)/pimaDesvEst
predict(arbolDecision, obsN, type="prob")
```

¿Este modelo podría predecir el riesgo de diabetes de un individuo de la tribu Pima de 90 años?

¿Sería este modelo capaz de predecir el riesgo de padecer diabetes de cualquiera de nosotros?

Regresión

Clasificación (Variable predicha es **categórica**)

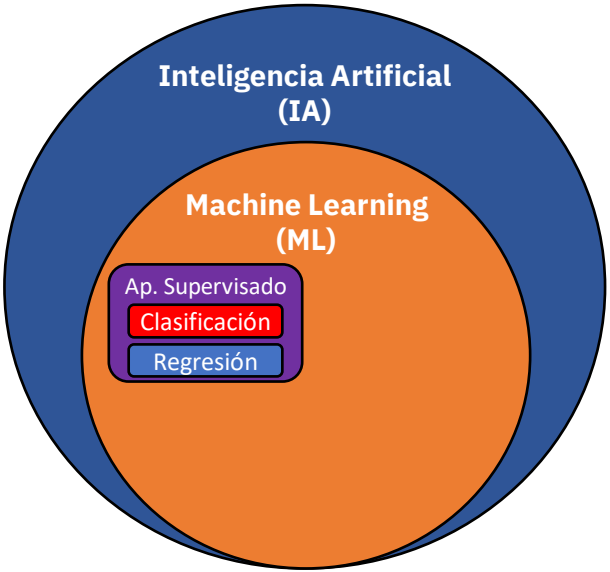
Buscar patrones para distinguir diferentes tipos de observaciones marcados por la variable respuesta.

Emails	Chequeo médico	Imágenes
¿Mensaje o spam?	¿Sano o enfermo?	¿Perro, gato, ratón?

Regresión (Variable predicha es **numérica**)

Predecir una variable respuesta relacionándola con otras variables independientes (como en regresión lineal).

Marketing	Inmobiliarias	Epidemiología
¿Beneficios?	¿Precio vivienda?	¿Contagios/semana?



Conjunto de datos: Marketing

Ya hemos afrontado un problema de regresión: la predicción de beneficios de una empresa según sus inversiones. Para cargar los datos, que volveremos a usar, podéis usar:

- Comandos de R: con el fichero en el directorio de trabajo, ejecutamos `load("marketing.rda")`
- R Commander: mediante el comando *Datos / Cargar conjunto de datos*.

MARKETING

¿Se pueden predecir los beneficios de una empresa en función de sus inversiones?

Variable dependiente (respuesta o a predecir)	▪ sales: Millones de $ de beneficio.
Variables independientes (regresores o predictores, características, *features*)	1. youtube: Millones de $ invertidos en Youtube. 2. facebook: Millones de $ invertidos en Facebook. 3. newspaper: Millones de $ invertidos en periódicos.

NOTA: No escalaremos los predictores por estar en la misma unidad y tener una distribución similar. Ante la duda, escalad.

Regresión: Regresión lineal

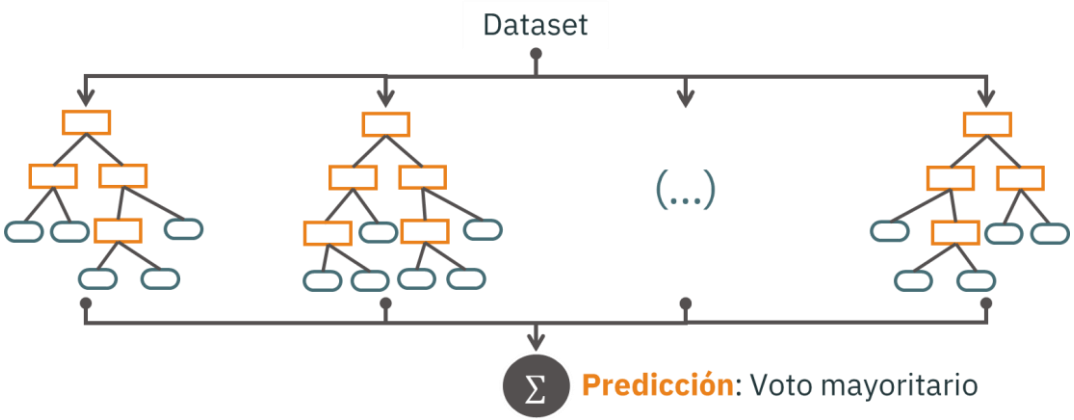
En el tema 2, empleamos un modelo de regresión lineal múltiple para predecir la variable respuesta, en concreto: $\text{sales} = f(X) + \varepsilon = \beta_0 + \beta_1\,\text{youtube} + \beta_2\,\text{facebook} + \beta_3\,\text{newspapers} + \varepsilon$.
Los comandos usados por R Commander fueron los siguientes:

```
RegModel.2 <- lm(sales~facebook+newspaper+youtube, data=marketing)
summary(RegModel.2)
RegModel.3 <- lm(sales~facebook+youtube, data=marketing)
summary(RegModel.3)    # R2 = 89.7 %
```

Como la regresión lineal es el modelo de regresión más simple, nos podemos preguntar: ¿Hay otro modelo más complejo (que no asuma que $f(X)$ es lineal) que explique mejor las ventas?

Regresión: Random Forest

Modelo en el que se inducen varios árboles de regresión (generalización de arb. decisión) con sub-conjuntos aleatorios de observaciones y variables. La predicción final es por voto mayoritario.



Regresión: Random Forest

Es un modelo muy flexible que se usa tanto como **método de regresión como de clasificación**. Su popularidad radica en que sus resultados son excelentes, robustos e interpretables, sin requerir que el usuario tenga conocimiento avanzados del procedimiento o el problema.

- Son un poco ***Black Box*** (extremo: redes neuronales o Support Vector Machine -SVM-).

El proceso de predicción del modelo Random Forest es análogo a pedir un pronóstico a varias personas, cada una con unos conocimientos del problema, y tomar la media como predicción.

Con las siguientes ordenes, instalaremos y cargaremos la librería necesaria:

```
install.packages("randomForest") # Necesario solo la primera vez
library(randomForest)
```

Regresión: Random Forest

Si creamos un modelo RF, ¿se predecirá mejor la variable `sales` que con la regresión lineal?

```
salesRandomForest<-randomForest(sales~., data = marketing)
salesRandomForest
```

Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 1

Por debajo, hay 500 árboles en los que se usa sólo una variable.

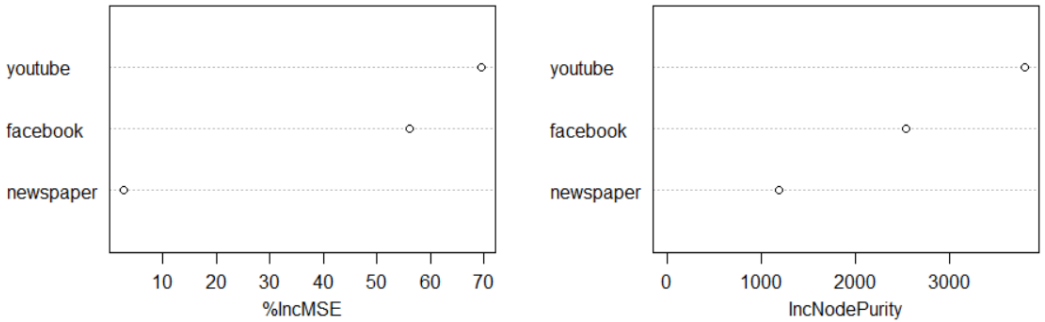
Mean of squared residuals: 3.050775
% Var explained: **92.18**

$$R^2_{\text{Reg. Lineal}} = 89.7\ \% \rightarrow R^2_{\text{Random Forest}} = 92.2\ \%$$

Random Forest destaca dentro de los modelos *Black Box* por su interpretabilidad, ya que ofrece una medida de la importancia de las variables implicadas en el modelo.

Regresión: Random Forest

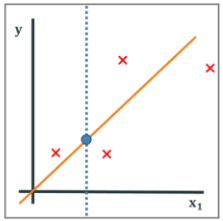
```
salesRandomForest<-randomForest(sales~., data = marketing, importance=TRUE)
varImpPlot(salesRandomForest)
```



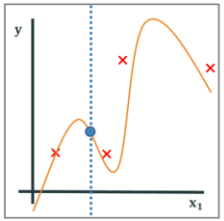
En ambos gráficos, cuánto más alto sea el valor de la variable, más relevante es en la predicción.
El eje X mide cuánto aumenta el error en aquellos árboles en los que la variable no está involucrada.

Aprendizaje supervisado: Consideraciones

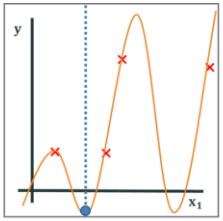
- **Limitaciones de los modelos:** para usarse en observaciones nuevas (no empleadas en el ajuste del modelo), éstas deben tener características similares (misma distribución y rango en las variables).
- **Sobreajuste (*overfit*):** Los modelos muy complejos (redes neuronales o, hasta cierto punto, RF) tienden a ajustar con demasiada exactitud las observaciones usadas en su ajuste. Causa que la predicción de una nueva observación sea predicha mucho peor que un modelo más sencillo.



UNDERFIT



GOOD FIT



OVERFIT

La estrategia más habitual para evitar el *overfit* es tener dos (o más) conjuntos de datos independientes.

- Train o entrenamiento: Sirven para ajustar el modelo.
- Test o prueba: Para comprobar la bondad de ajuste.

Tema 6: Fundamentos de IA

3 Otras técnicas de ML: PCA y Clustering

A ¿Qué otras técnicas de Machine Learning existen?

B Resolución un problema de reducción de dimensionalidad con la técnica PCA en R.

Otras técnicas de Machine Learning

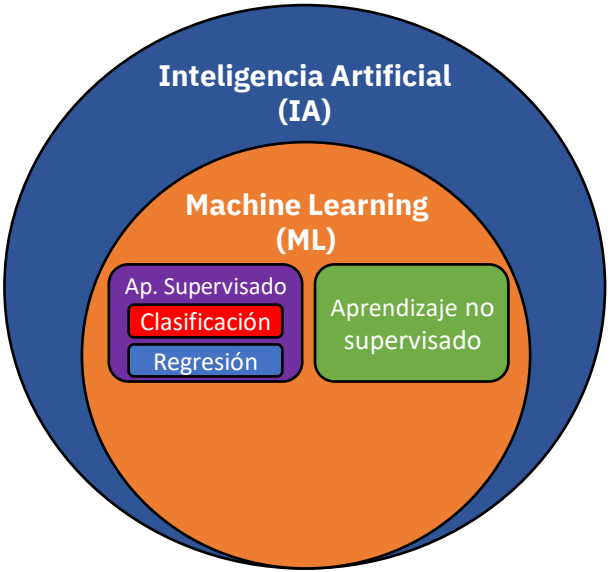
Aprendizaje no supervisado (*Unsupervised Learning*)

Detectar patrones recurrentes para agrupar a las observaciones en clusters (**clases o grupos *no observadas***). Destacan, como modelos, las k-medias, clustering jerárquico y los basados en densidad.

Comercio online
Tipos de clientes

Medicina
Subtipos enfermedad

Genética
Grupos genes por función



Otras técnicas de Machine Learning

Aprendizaje no supervisado (*Unsupervised Learning*)

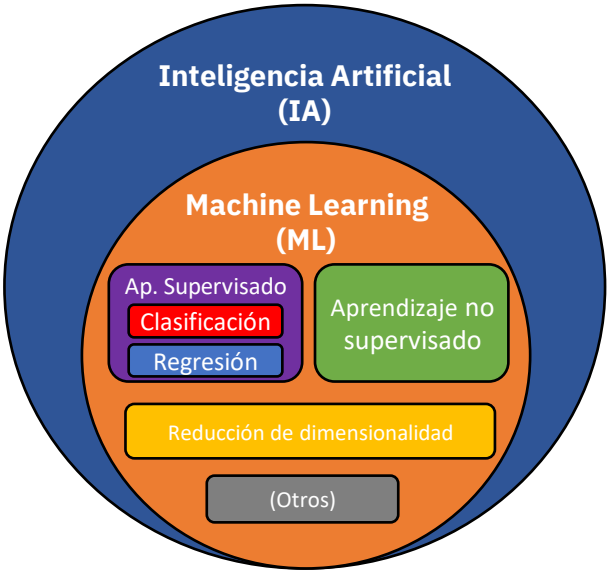
Detectar patrones recurrentes para agrupar a las observaciones en clusters (**clases o grupos *no observadas***). Destacan, como modelos, las k-medias, clustering jerárquico y los basados en densidad.

Comercio online	Medicina	Genética
Tipos de clientes	Subtipos enfermedad	Grupos genes por función

Reducción de dimensionalidad (N.º variables ⌄⌄)

Transformar un conjunto (grande) de variables en otro más reducido con proyecciones o seleccionando variables. Destacan, entre estos métodos, las componentes principales (PCA) o t-SNE.

Imágenes	Gráficos	Machine Learning
Reducir resolución	Representar en 2D/3D	Reducir nº. variables



Análisis en Componentes Principales (PCA)

Método de reducción de dimensionalidad que busca m^* **componentes principales (CP)**, combinaciones lineales del conjunto inicial de variables que expliquen la máxima variabilidad. Se supone que la utilidad (información) de una variable equivale a su varianza.

La transformación $X_{n\times m} \xrightarrow{f_1, \dots, f_n} X^*_{n\times m^*}$ tienen la siguiente expresión:

$$CP_j = X^*_j = f_j(X|b_{j1}, \dots, b_{jm}) = b_{j1}X_1 + b_{j2}X_2 + \cdots + b_{jm}X_m \qquad \forall j = 1,2, \dots, m^*$$

En R, el PCA se calcula con la función `prcomp`. El análisis de esta subsección se ha realizado con las funciones `fviz_eig`, `fviz_pca_ind` y `fviz_pca_var` del paquete `factoextra`.

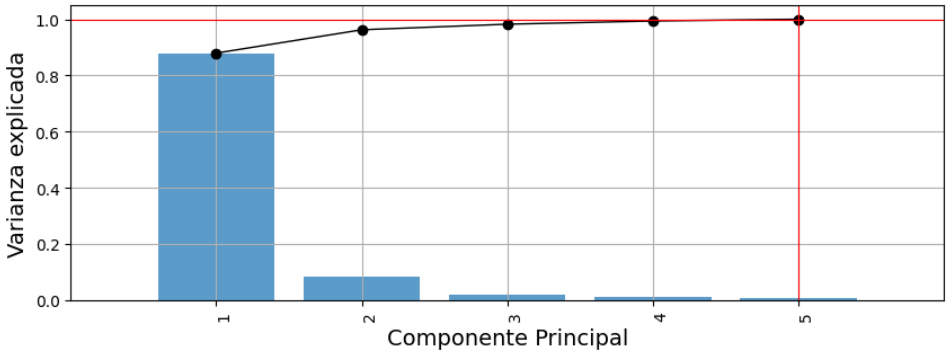
Los gráficos de esta subsección corresponden al conjunto de datos “CO2 Emission by Vehicles”.

Análisis en Componentes Principales (PCA)

Los resultados de PCA se estudian mediante gráficos, al menos en una fase preliminar.

Gráficos de varianza explicada o *Screeplot*

Representación de la varianza explicada por cada componente respecto al total. Ayuda a determinar cuántas componentes extraer (extraer 2 o hasta que sumen el 95% de varianza).

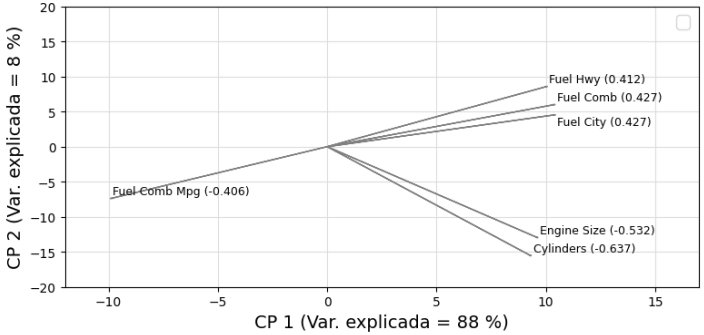


	Var. explicada
CP 1	87.94 %
CP 2	8.31 %
CP 3	1.99 %
CP 4	1.15 %
CP 5	0.61 %

Análisis en Componentes Principales (PCA)

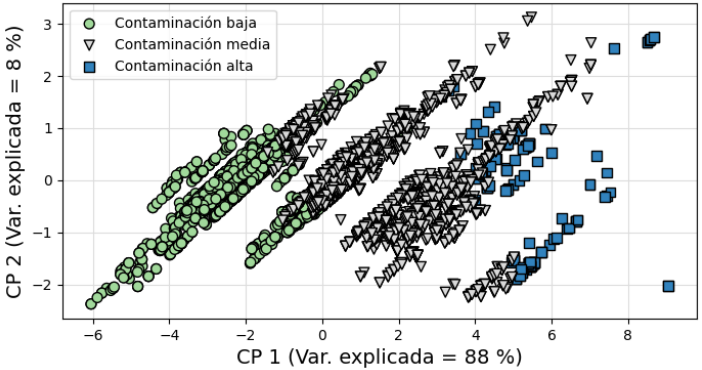
Gráfico de cargas o *Biplot*

Representación gráfica de las cargas $b_{j,1}, b_{j,2}, \ldots, b_{j,m}$ de las CP. Sirve para realizar comparaciones entre ellas.



Diagramas de dispersión de las CP

Representación bidimensional de las proyecciones de las observaciones. A veces, se representan en función de una variable categórica.



Análisis en Componentes Principales (PCA)

Aplicaciones

Dentro de la ciencia y el análisis de datos, se usa con multitud de propósitos:

- Como paso previo en un procedimiento de aprendizaje, sobre todo si $\mathrm{m} \geq \mathrm{n}$.
- Método de detección de atípicos (*outliers*) multivariante.
- Visualización de datos de alta dimensión: estructura y similitudes entre observaciones.

Desde un punto de vista más aplicado, se emplean en multitud de campos para, por ejemplo, comprimir imágenes y sonido (menos resolución, el tamaño de los ficheros disminuye).

El PCA es un método sumamente versátil, que se puede aplicar a todo tipo de datos. Sin embargo, para aplicaciones concretas, suele haber métodos con más ventajas.

Tema 6: Fundamentos de IA

❹ Redes neuronales

Después de repasar de distintas técnicas de Machine Learning, no hemos hecho mención alguna del **modelo de IA** más celebre: ¿A qué campo de la IA pertenecen?

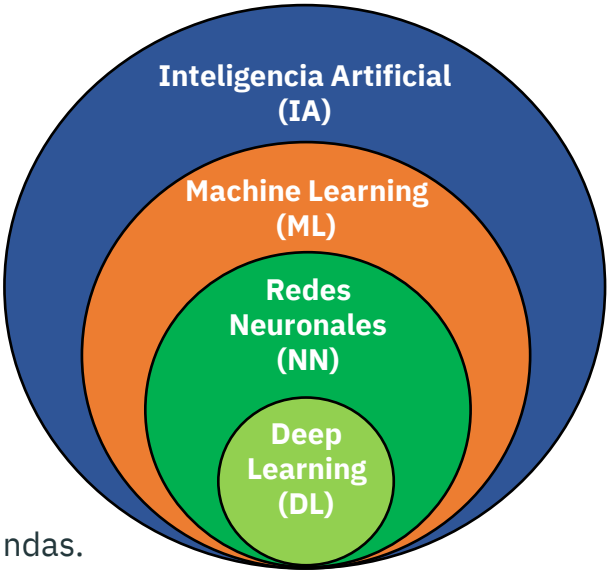
Redes neuronales

Amplio conjunto de modelos de ML capaces de resolver complejas tareas, **siempre y cuando se cuenten con suficientes datos**.

Son modelos muy diversos que tienen en común el componerse de capas de neuronas, unidades de cómputo que generan una salida con el input.

- **Aprendizaje profundo** (*Deep Learning*)

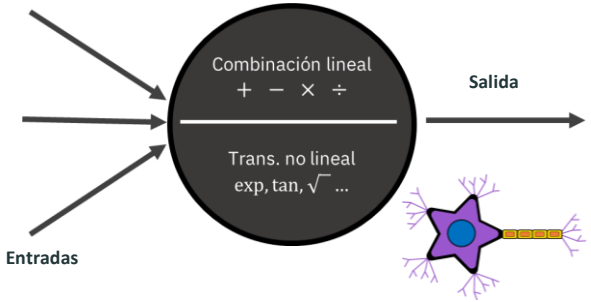
¿Problema complejo con muchos datos? → NN muy profundas.



Redes neuronales

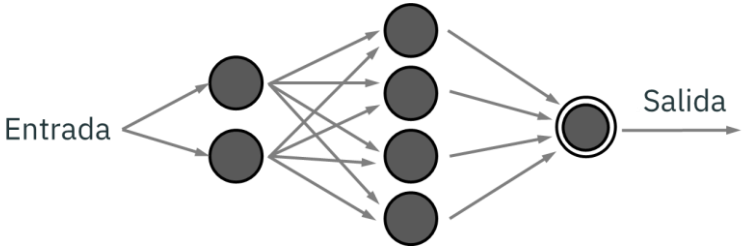
Neurona

Unidad de cómputo básico en las NN. Cada neurona, en base a una entrada (dendritas), calcula (en el soma) una salida (axón).



Capas

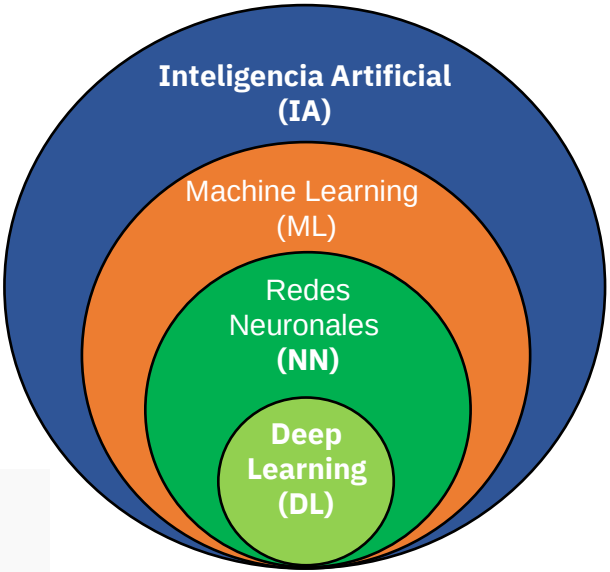
Asociación de neuronas encargadas de extraer un tipo concreto de características. Cuanto más profunda es la capa, más complejas son las características extraídas.



Redes neuronales

Relación de las NN con las técnicas de ML

- Su principal uso es en Aprendizaje Supervisado
  - Para resolver problemas de Clasificación y Regresión.
- Su uso en Aprendizaje no supervisado es menos común.
- Intrínsecamente, las neuronas realizan proyecciones y tareas de Reducción de dimensionalidad.



Entonces, ¿Son los modelos superiores de ML?
No tiene por qué. Son modelos muy complejos que necesitan de un mayor volumen datos. Son poco interpretables (*Black Box*).

EJERCICIOS

Estadística e IA con R

Alejandro Rodríguez-Collado

alejandrorodriguezcollado@gmail.com

Ejercicio 1

Carga el conjunto de datos **`mtcars`**, incluido en el paquete `datasets`. Éste se instala por defecto con R. Es recomendable que leas el manual de ayuda del conjunto de datos (usando R Commander, o con el comando `?mtcars` en la consola de R Studio).

Contesta a las siguientes preguntas haciendo uso de **R Commander**:

- Realiza un resumen numérico de las variables. ¿Cuál es la media de la potencia de los coches (variable `hp`)? ¿Y la mediana del número de marchas (variable `gear`)?
- La variable `mpg` contiene el consumo de cada coche en millas por galón americano. Crea una nueva variable con el consumo de los coches en litros por cada 100 kilómetros con la siguiente transformación: `consumoEU = 235.21 / mpg`.

Ejercicio 1

- ¿Qué comentarios puedes hacer sobre la distribución de la variable `consumoEU`, realizando cálculos de los estadísticos más relevantes y/o creando gráficos?
- Crea un diagrama de cajas múltiple para la variable `consumoEU` para cada valor de `cyl` ¿Qué relación parece existir entre estas dos variables?
  Previamente, deberás transformar `cyl` para que R trate esta variable como categórica. Para ello, haz click en *Datos / Modificar variables del conjunto de datos activo / Convertir variable numérica en factor* y selecciona la variable `cyl`.
- Crea con el paquete `esquisse` una representación gráfica que destaque algún aspecto de interés del conjunto de datos `mtcars`.

Ejercicio 2

El conjunto de datos **`bostonHousing`** contiene información sobre las viviendas de los barrios de Bostón en 1978. Las variables más relevantes del conjunto de datos son las siguientes:

Variable	Descripción
`MEDV`	Precio mediano en miles de $ de las casas del barrio.
`AGE`	Años de antigüedad media de las casas del barrio.
`CRIM`	Índice de criminalidad del barrio.
`RM`	Número medio de habitaciones por casa del barrio.
`LSTAT`	Porcentaje de vecinos con ingresos bajos en el barrio.
`PTRATIO`	Ratio de alumnos por profesor en los colegios del barrio.

Ejercicio 2

Contesta a las siguientes preguntas usando comandos de **R** y **R Commander**:

- Carga el conjunto de datos en R. ¿Cuántas observaciones y variables tiene? Suponiendo que has llamado a los datos `bostonHousing` (cuidado con minúsculas y mayúsculas), ejecuta las siguientes órdenes para eliminar filas con faltantes. ¿Cuántas observaciones quedan?

```
filaTieneNAs <- apply(is.na(bostonHousing),1,any)
bostonHousing <- bostonHousing[!filaTieneNAs,]
```

- ¿Cuántos años de media tienen las casas del barrio situado en la fila 24? ¿Cuál es el precio mediano de las casas del barrio cuyas casas son las más nuevas?
- ¿Sigue la variable índice de criminalidad una distribución normal?

Ejercicio 2

- Una ciudad tiene alta criminalidad si su índice es significativamente superior a 2.5, ¿Era Bostón, en 1978, una ciudad de alta criminalidad? Contesta con un contraste de hipótesis.
- En este apartado, crearemos varios modelos para predecir el valor mediano de las casas de los barrios (`MEDV`) a partir del resto de variables. Realiza las siguientes tareas:
  1. Escala todas las variables del conjunto de datos, menos la variable respuesta (`MEDV`).
  2. Predice `MEDV` con un modelo de regresión lineal múltiple que use las variables `AGE`, `CRIM`, `RM`, `LSTAT`, `PTRATIO`. ¿Cuál es el R^2? ¿Hay alguna variable poco significativa?
  3. Predice `MEDV` con un modelo Random Forest que use todo el resto de las variables del conjunto de datos, ¿Cuál es el R^2? ¿Qué variables son más relevantes en la predicción?

Bibliografía

- J. Assaker (2021). *Dataset: Covid-19 Global Data*. Kaggle.
  URL: https://www.kaggle.com/datasets/josephassaker/covid19-global-dataset
- R. Cushny y A. R. Peebles (1905). Dataset: Sleep - The action of optical isomers: II hyoscines. *The Journal of Physiology* 32, páginas 501–510.
  URL: https://doi.org/10.1113/jphysiol.1905.sp001097
- P. Debajyoti y Canada Government (2020). *Dataset: CO2 Emission by Vehicles*. Kaggle.
  URL: https://www.kaggle.com/datasets/debajyotipodder/co2-emission-by-vehicles
- R. Dua, M. Singh Ghotra y N. Pentreath (2017). *Machine Learning with Spark*. Packt Publishing, segunda edición. ISBN: 978-1785889936.

Bibliografía

- J. Fox (2005). The R Commander: A Basic Statistics Graphical User Interface to R. *Journal of Statistical Software*, 14(9), páginas 1–42. URL: https://doi.org/10.18637/jss.v014.i09
- J. Fox (2017). *Using the R Commander: A Point-and-Click Interface for R*. Chapman and Hall/CRC Press. ISBN: 978-1498741903.
- J. Fox, M. M.Marquez y M. Bouchet-Valat (2024). *Rcmdr: R Commander*. R package, versión 2.9-2. URL: https://socialsciences.mcmaster.ca/jfox/Misc/Rcmdr/
- D. Gandy (2024). *Font Awesome: the Internet's icon library and toolkit*, versión 6.5.1. URL : https://fontawesome.com/

Bibliografía

- T. Hastie, R. Tibshirani y F. Jerome (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Series in Statistics, segunda edición. ISBN: 9780387848570.
- D. Harrison y D. L. Rubinfeld (1978). Dataset: Boston House Prices - Hedonic prices and the demand for clean air. *J. Environ. Economics & Management*, 5, páginas 81-102. URL: https://doi.org/10.1016/0095 -0696(78)90006 -2
- H. V. Henderson y P. F. Velleman (1981). Dataset: Mtcars - Building Multiple Regression Models Interactively. *Biometrics*, 37(2), páginas 391-411. URL: https://doi.org/10.2307/2530428

Bibliografía

- H. V. Henderson y P. F. Velleman (1981). Dataset: Sleep - Building Multiple Regression Models Interactively. *Biometrics*, 37(2), páginas 391-411.
  URL: https://doi.org/10.2307/2530428
- K. Hornik, C. Buchta y A. Zeileis (2009). Open-Source Machine Learning: R Meets Weka. *Computational Statistics*, 24(2), páginas 225–232.
  URL: https://doi.org/10.1007/s00180 -008- 0119 -7
- T. Hothorn, F. Bretz y P. Westfall (2008). Simultaneous Inference in General Parametric Models. *Biometrical Journal*, 50(3), páginas 346-363.
  URL: https://doi.org/10.1002/bimj.200810425

Bibliografía

- IBM (2023). *IBM Plex Sans Font: It's global, it's versatile and it's distinctly IBM*, versión 6.4.0. URL : https://www.ibm.com/plex/
- J. Izenman (2008). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Texts in Statistics, primera edición. ISBN: 9780387781884.
- A. Kassambara (2017). *Dataset: Marketing Data*. Github. URL: https://github.com/kassambara/datarium/blob/master/data/marketing.rda
- A. Kassambara y F. Mundt (2020). *Factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R Package, versión 1.0.7. URL: https://CRAN.R-project.org/package=factoextra

Bibliografía

- J. D. Kelleher, B. M. Namee y A. D'Arcy (2020). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press, segunda edición. ISBN: 9780262361101.
- F. Lasso, World Factbook y US government (2018). *Dataset: Countries of the World*. Kaggle. URL: https://www.kaggle.com/datasets/fernandol/countries -of-the-world
- A. Liaw y M. Wiener (2002). Classification and Regression by randomForest. *R News*, 2(3), páginas 18-22. URL: https://journal.r -project.org/articles/RN -2002- 022/RN-2002-022.pdf
- L. van der Maaten y G. Hinton (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (86), páginas 2579-2605. URL: https://jmlr.org/papers/v9/vandermaaten08a.html

Bibliografía

- F. Meyer, V. Perrier (2024). *esquisse: Explore and Visualize Your Data Interactively*. R package, versión 1.2.0. URL: https://dreamrs.github.io/esquisse/
- F. Meyer, V. Perrier (2024). *esquisse: Explore and Visualize Your Data Interactively*. Github. URL: https://github.com/dreamRs/esquisse
- D. J. Newman, S. Hettich, C. L. Blake y C. J. Merz (1998). *Dataset: Indian Pima Diabetes - UCI: Repository of machine learning databases*. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html
- A. Ng, Y. B. Mourri y K. Katanforoosh (2021). *Deep Learning Specialization*. DeepLearning.AI a través de Coursera. URL: https://www.coursera.org/specializations/deep -learning

Bibliografía

- I. Patil (2021). Visualizations with statistical details: The 'ggstatsplot' approach. *Journal of Open Source Software*, 6(61), página 3167. URL: https://doi.org/10.21105/joss.03167
- F. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, páginas 2825-2830.
  URL: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
- D. Ripley (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, primera edición. ISBN: 9780511812651.
- A. Rodríguez-Collado (2022). *Data Science with R*. Cargraf, primera edición.
  ISBN: 978-84-125460-2-6.

Bibliografía

- RStudio Team (2020). *RStudio: Integrated Development for R*. RStudio Development Team. URL: http://www.rstudio.com/

- R Core Team (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. URL: https://www.R -project.org

- Student (1908). Dataset: Sleep - The probable error of the mean. *Biometrika* 6(1), 1–25. URL: https://doi.org/10.2307/2331554

Bibliografía

- R. Wirth y J. Hipp (2000). Crisp-DM: towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, páginas 29-40.
  URL: https://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

- H. Witten y E. Frank (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, cuarta edición. ISBN: 0128042915.

Alejandro Rodríguez-Collado

Estadística e Inteligencia Artificial con R

9 788412 857719

La era actual requiere de conocimientos básicos de inteligencia artificial y estadística. El análisis semiautomático de nuestros datos ha cambiado para siempre la vida diaria: se usa para calcular el riesgo de impago de un préstamo, elegir el mejor tratamiento para una enfermedad e, incluso, recomendarnos qué serie ver, entre otros. Es tal la intromisión que ya no somos capaces de distinguir imágenes sintéticas de las reales.

Este libro permitirá al lector obtener estas competencias elementales mediante la resolución de problemas con datos reales usando el lenguaje de programación R. La compresión del texto no requiere de conocimientos previos, tan solo de motivación para aprender.