

Clase 1.2

Datos y dónde encontrarlos

Marcos Rosetti y Luis Pacheco-Cobos

Estadística y Manejo de Datos con R (EMDR) — Virtual

Bases de datos

Introducción

- “Era de los Datos”: procesos automatizados generan datos en gran cantidad.
- Hoy, obtener datos es más fácil que saber qué hacer con ellos.
- R ofrece varias bases de datos (pequeñas y grandes) “curadas” para practicar
- También hay excelentes métodos de generación de *mock data*.

Dónde encontrar datos

En la red global

- Google data sets (<https://toolbox.google.com/datasetsearch>)
- 1000 Genomes Project (<https://www.internationalgenome.org/>) contiene genomas humanos diversos.
- Food and Agriculture Organization (<https://www.fao.org/statistics/databases/en/>) producción de alimentos y agricultura mundial.
- Department of Transport, UK (<https://www.dft.gov.uk/>) contiene georeferencias de eventos del tráfico del Reino Unido.
- SIMAT (<https://www.aire.cdmx.gob.mx/>) monitoreo en tiempo real de los datos de contaminación de la CDMX.
- The Digital Archaeological Data <https://core.tdar.org/>
- World Bank Open Data <https://data.worldbank.org/>
- <https://www.reddit.com/r/datasets> es una comunidad donde se comparten data sets variados.

En R: data()

Con la función `data()` podemos conocer los datos curados que R tiene por defecto.

```
data()
```

```
data(Animals, package = "MASS")  
head(Animals)
```

```
##           body brain  
## Mountain beaver  1.35  8.1  
## Cow              465.00 423.0  
## Grey wolf       36.33 119.5  
## Goat            27.66 115.0  
## Guinea pig      1.04  5.5  
## Dipliodocus    11700.00 50.0
```

```
help(Animals, package = "MASS")
```

En R: data ()

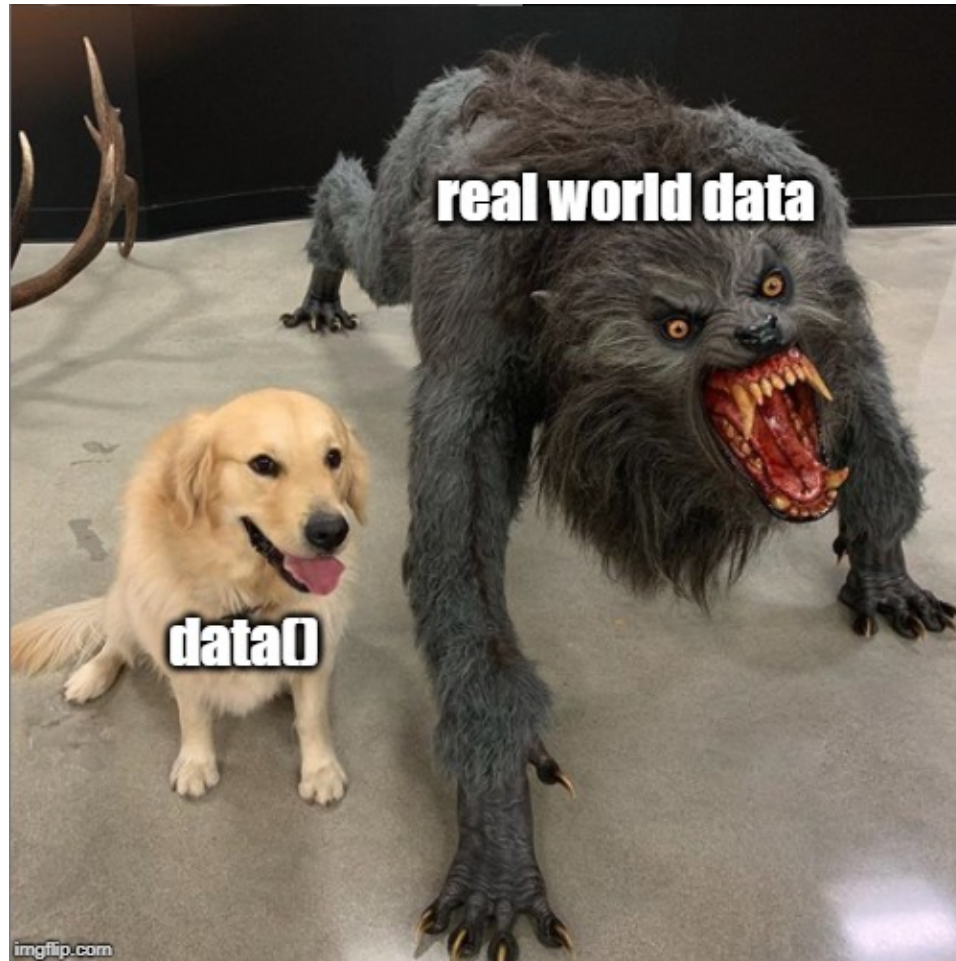
```
head(LifeCycleSavings)
```

```
##           sr pop15 pop75      dpi ddpi
## Australia 11.43 29.35  2.87 2329.68 2.87
## Austria   12.07 23.32  4.41 1507.99 3.93
## Belgium   13.17 23.80  4.43 2108.47 3.82
## Bolivia    5.75 41.89  1.67  189.13 0.22
## Brazil    12.88 42.19  0.83  728.47 4.56
## Canada     8.79 31.72  2.85 2982.88 2.43
```

```
help(LifeCycleSavings)
```

- Intercountry Life-Cycle Savings Data.
- Data on the savings ratio 1960–1970.
 - A data frame with 50 observations on 5 variables.
 - [,1] sr numeric aggregate personal savings.
 - [,4] dpi numeric real per-capita disposable income.

En R: `data()`



Bases de datos: *Mock data*

- También es posible generar mock data (datos falsos) para explorar algún aspecto.
- R tiene varias funciones que generan datos aleatorios con distintas distribuciones.
- Únicamente, tenemos que especificar los parámetros estadísticos y las dimensiones de los marcos de datos.

Bases de datos: *Mock data*

Distribution	Function to generate numbers
Beta	<code>rbeta()</code>
Binomial	<code>rbinom()</code>
Chi-square	<code>rchisq()</code>
Exponential	<code>rexp()</code>
Gamma	<code>rgamma()</code>
Geometric	<code>rgeom()</code>
Logistic	<code>rlogis()</code>
Log Normal	<code>rlnorm()</code>
Negative Binomial	<code>rnbinom()</code>
Normal	<code>rnorm()</code>
Poisson	<code>rpois()</code>
Uniform	<code>runif()</code>
Weibull	<code>rweibull()</code>

Bases de datos: *Mock data*

```
x <- runif(10, min = 0, max = 10)
x
```

```
## [1] 5.7443774 3.9676058 2.0696649 8.0424371 9.9792133 5.4457259 2.0050145
## [8] 0.2389672 6.9369555 6.7867221
```

```
x <- rnorm(10, mean = 10, sd = 0.5)
x
```

```
## [1] 9.724596 9.432240 10.013471 10.450694 10.882622 9.957986 10.080627
## [8] 9.711423 9.581355 9.088243
```

```
x <- rpois(10, lambda = 5)
x
```

```
## [1] 4 5 5 6 2 1 10 9 6 7
```

Bases de datos: *Mock data*

- La función `sample()` equivale a elegir un número de una serie.

```
x <- sample(1:10, 1)  
x
```

```
## [1] 3
```

- O en el caso que permitamos las repeticiones

```
x <- sample(1:10, 5, replace=TRUE)  
x
```

```
## [1] 8 2 7 1 3
```

Leer y guardar datos

Bases de datos: Leer y guardar

- Podemos cargar archivos desde una localización local o desde un URL.
- La funcionalidad básica de R permite lectura del formato `.csv`.
- Para otras extensiones (`.xls`, `.xlsx`, `.sav`, `.gpx`, etc) existen paquetes adicionales.
- Es importante explorar los parámetros de estas funciones para leer y guardar datos tal cual los queremos.

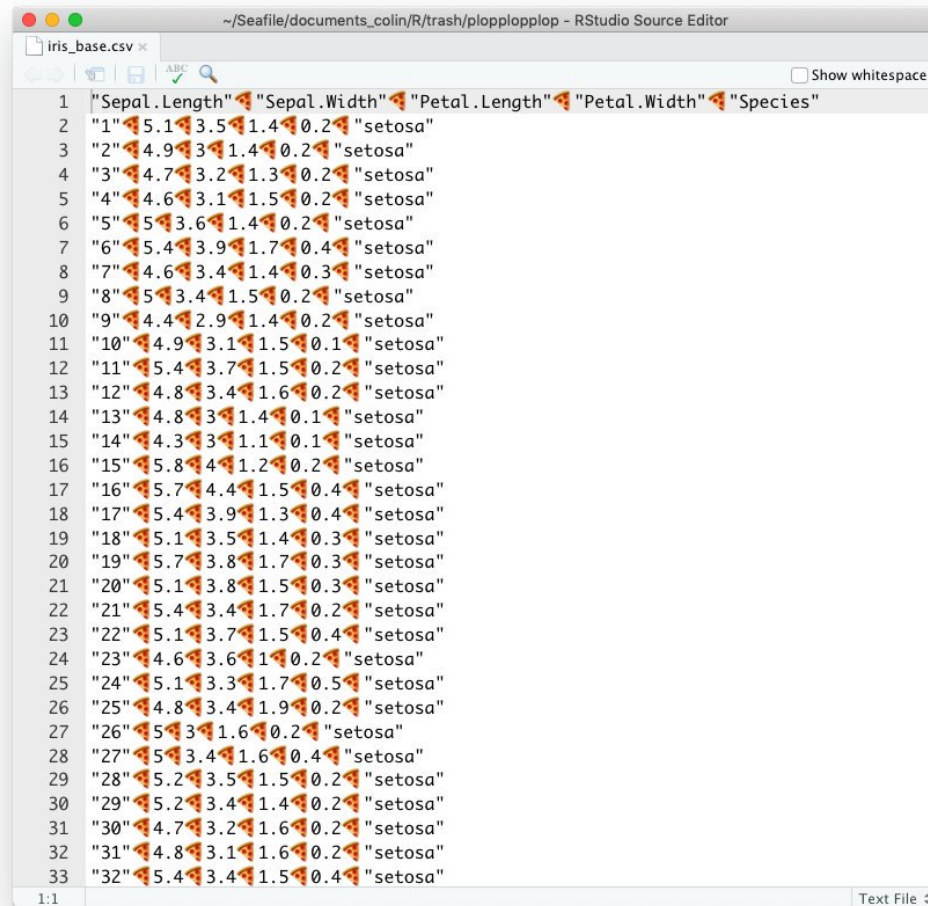
Bases de datos: Leer y guardar

- La función básica para leer es `read.table()`

```
read.table(path_to_filename, header = FALSE, sep = ",", row.names, col.names,  
           skip = 0, fill = !blank.lines.skip,  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#")
```

- Podemos elegir el caracter que separa las columnas: ",", "\t", " "

Bases de datos: Leer y guardar



```
iris_base.csv
1 "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
2 "1" 5.1 3.5 1.4 0.2 "setosa"
3 "2" 4.9 3.0 1.4 0.2 "setosa"
4 "3" 4.7 3.2 1.3 0.2 "setosa"
5 "4" 4.6 3.1 1.5 0.2 "setosa"
6 "5" 5.0 3.6 1.4 0.2 "setosa"
7 "6" 5.4 3.9 1.7 0.4 "setosa"
8 "7" 4.6 3.4 1.4 0.3 "setosa"
9 "8" 5.0 3.4 1.5 0.2 "setosa"
10 "9" 4.4 2.9 1.4 0.2 "setosa"
11 "10" 4.9 3.1 1.5 0.1 "setosa"
12 "11" 5.4 3.7 1.5 0.2 "setosa"
13 "12" 4.8 3.4 1.6 0.2 "setosa"
14 "13" 4.8 3.0 1.4 0.1 "setosa"
15 "14" 4.3 3.1 1.1 0.1 "setosa"
16 "15" 5.8 4.1 1.2 0.2 "setosa"
17 "16" 5.7 4.4 1.5 0.4 "setosa"
18 "17" 5.4 3.9 1.3 0.4 "setosa"
19 "18" 5.1 3.5 1.4 0.3 "setosa"
20 "19" 5.7 3.8 1.7 0.3 "setosa"
21 "20" 5.1 3.8 1.5 0.3 "setosa"
22 "21" 5.4 3.4 1.7 0.2 "setosa"
23 "22" 5.1 3.7 1.5 0.4 "setosa"
24 "23" 4.6 3.6 1.0 0.2 "setosa"
25 "24" 5.1 3.3 1.7 0.5 "setosa"
26 "25" 4.8 3.4 1.9 0.2 "setosa"
27 "26" 5.0 3.0 1.6 0.2 "setosa"
28 "27" 5.0 3.4 1.6 0.4 "setosa"
29 "28" 5.2 3.5 1.5 0.2 "setosa"
30 "29" 5.2 3.4 1.4 0.2 "setosa"
31 "30" 4.7 3.2 1.6 0.2 "setosa"
32 "31" 4.8 3.1 1.6 0.2 "setosa"
33 "32" 5.4 3.4 1.5 0.4 "setosa"
```


Bases de datos: Leer y guardar

- La función básica para guardar es `write.table()`

```
write.table(x, file = "path_to_filename", append = FALSE, sep = ",",  
           row.names = TRUE, col.names = TRUE)
```

Otros paquetes para leer datos

Bases de datos: **readxl**



Bases de datos: **readxl**

- Añade funcionalidad para leer archivos `.xls` y `.xlsx` directamente.
- Funciones `read_excel()`, `read_xls()` y `read_xlsx()`

```
install.packages("readxl")
library(readxl)
read_excel(path_to_filename, sheet = NULL, range = NULL, col_names = TRUE,
           na = "", trim_ws = TRUE, skip = 0)
```

Bases de datos: **readr**



Bases de datos: readr

```
install.packages("readxl")  
library(readr)  
read.csv(filename)
```

La ayuda en R

Solicitar la ayuda en R

- ¿Cómo?
 - Tecleando en la consola `?función()`, por ejemplo: `?plot()`
- Lo que nos mostrará su descripción y contenidos
 - `función {paquete}`
 - Uso: `función(argumento 1, argumento 2, etc.)`
 - Argumentos: descripción operativa
 - Detalles
 - Notas
 - Referencias: libros, artículos o enlaces
 - Ver también: funciones relacionadas
 - Ejemplos: ejecutables

Bases de datos: Exploración

- Después de leer y asignar a un objeto en R los datos, podemos iniciar su exploración.
- Con `head()` y `tail()` podemos conocer la parte superior e inferior del df (marco de datos, por su acrónimo en inglés).
- Con `str()` podemos conocer su estructura y el tipo de datos que contiene.
- Con `summary()` podemos conocer un resumen descriptivo del marco de datos.
- Otro aspecto importante es manipular el df para obtener un subconjunto, separar o juntar columnas, deshacerse de casos o celdas vacías, etc.

Bases de datos: Cabeza y cola

```
head(USArrests) # cabeza
```

```
##           Murder Assault UrbanPop Rape
## Alabama      13.2      236       58 21.2
## Alaska       10.0      263       48 44.5
## Arizona       8.1      294       80 31.0
## Arkansas      8.8      190       50 19.5
## California    9.0      276       91 40.6
## Colorado     7.9      204       78 38.7
```

```
tail(USArrests) # cola
```

```
##           Murder Assault UrbanPop Rape
## Vermont       2.2       48       32 11.2
## Virginia      8.5      156       63 20.7
## Washington    4.0      145       73 26.2
## West Virginia 5.7       81       39  9.3
## Wisconsin     2.6       53       66 10.8
## Wyoming      6.8      161       60 15.6
```

Bases de datos: Dimensiones

```
dim(Seatbelts) # filas y columnas
```

```
## [1] 192  8
```

```
dim(Titanic) # ¿filas y columnas? ¿y/o qué más?
```

```
## [1] 4 2 2 2
```

Bases de datos: Descripción estructural

- ¿Qué sucede con la estructura de algunos conjuntos de datos?

```
str(Seatbelts)
```

```
## Time-Series [1:192, 1:8] from 1969 to 1985: 107 97 102 87 119 106 110 106 107 134 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:8] "DriversKilled" "drivers" "front" "rear" ...
```

```
str(Titanic)
```

```
## 'table' num [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...  
## - attr(*, "dimnames")=List of 4  
## ..$ Class : chr [1:4] "1st" "2nd" "3rd" "Crew"  
## ..$ Sex : chr [1:2] "Male" "Female"  
## ..$ Age : chr [1:2] "Child" "Adult"  
## ..$ Survived: chr [1:2] "No" "Yes"
```

Estadística descriptiva en un paso

```
summary(Titanic)
```

```
## Number of cases in table: 2201
## Number of factors: 4
## Test for independence of all factors:
##  Chisq = 1637.4, df = 25, p-value = 0
##  Chi-squared approximation may be incorrect
```

```
summary(iris)
```

```
##   Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
##   Min.      :4.300   Min.      :2.000   Min.      :1.000   Min.      :0.100
##   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##   Median :5.800   Median :3.000   Median :4.350   Median :1.300
##   Mean    :5.843   Mean    :3.057   Mean    :3.758   Mean    :1.199
##   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##   Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2.500
##           Species
##   setosa      :50
##   versicolor :50
##   virginica   :50
##
##
##
```

Nombres de las variables (columnas)

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

```
# ¿Qué sucede con 'WorldPhones'?  
names(WorldPhones)
```

```
## NULL
```

```
str(WorldPhones)
```

```
## num [1:7, 1:7] 45939 60423 64721 68484 71799 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : chr [1:7] "1951" "1956" "1957" "1958" ...  
## ..$ : chr [1:7] "N.Amer" "Europe" "Asia" "S.Amer" ...
```

```
colnames(WorldPhones)
```

```
## [1] "N.Amer" "Europe" "Asia" "S.Amer" "Oceania" "Africa" "Mid.Amer"
```

```
help(iris)  
help(WorldPhones)
```

Licencia CC BY



Estadística y Manejo de Datos con R (EMDR) por Marcos F. Rosetti S. y Luis Pacheco-Cobos se distribuye bajo una [Licencia Creative Commons Atribución 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).