Clase 1.4
Cargando y explorando datos

Marcos Rosetti y Luis Pacheco-Cobos

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

Carga de datos

- Las funciones `read.table()` y `read.csv()` nos permiten leer los datos que tengamos almacenados con formato tabular.
- Necesitamos especificar si los datos están separados por comas, por algún caracter especial o por tabulador.
- Para esto utilizamos los argumentos de la función. Si no los conocemos pedimos ayuda a R `?read.table()` o `?read.csv()`

La ayuda en R

- funcion {paquete}
- Descripción
- Uso: funcion(argumento 1, argumento 2, etc.)
- Argumentos: descripción operativa
- Detalles
- Nota
- Referencias: libros, artículos o enlaces
- Ver también: funciones relacionadas
- Ejemplos: ejecutables

Comandos genéricos para cargar datos

```
# x <- read.table("archivo.csv", sep=",")
# x <- read.csv("archivo.csv")
?read.csv()
```

- read.table {utils} R Documentation
- Data Input
    - Description
        - Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.
    - Usage
        - read.csv(file, header = TRUE, sep = ",", quote = "\"", dec = ".", fill = TRUE, comment.char = "", …)

Exploración general de datos

- 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.

Conjunto de datos: Cabeza y cola

- `data()` nos muestra una lista de los conjuntos de datos que R tiene pre-cargados como ejemplos.

```
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
```

Conjunto de datos: Dimensiones

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

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

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

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

Conjunto de datos: Estructura

- ¿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"
```

Ejercicios

- Carga un conjunto de datos desde un archivo *.csv
- Pistas

```
# read.table()
# read.csv()
```

- Asígnalos a un objeto (nombre corto) en R con el operador flecha <-

Bases de datos: **googlesheets**

```
install.packages("googlesheets")
library(googlesheets)
# Requerirás una cuenta en Google
gs_ls() # autenticación
gs_title("Britain Elects / Public Opinion")
```

Bases de datos: **googlesheets**

```
install.packages("RCurl")
library(RCurl)
#bycatch <- getURL('https://sakai.unc.edu/access/content/group/3d1eb92e-7848-4f55-90c3-7c72a5
# No carga como marco de datos
```

Licencia CC BY