

# Introducción a R

Andrés Vallone

Escuela de Ciencias Empresariales

2019



# Contenidos

Repaso de la sesión anterior

Importando y exportando datos

Filtrado de datos



Repaso de la sesión anterior

# Reapso de la sesión anterior

- ▶ “todo en R es un objeto”

- ▶ Objetos atómicos:

- ▶ numeric
    - ▶ interger
    - ▶ character
    - ▶ LOGICAL

- ▶ Estrcuturas básicas e indexación

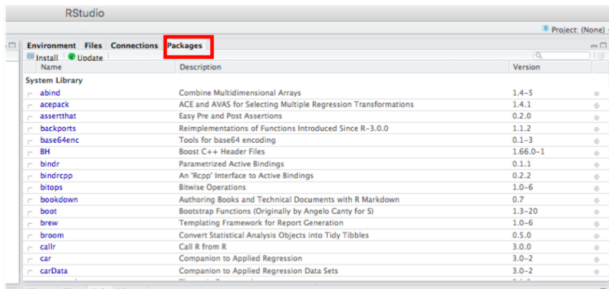
Scalar	-
Vector	$x[i]$
Matrix	$x[i,j]$
Dataframe	$x[i,j]$
List	$x[[k]][i]$ or $x[[k]][i,j]$

- ▶ Operadores Básicos

- ▶ Usando funciones nombre(argumentos)

# Usando los paquetes en R

- ▶ Los paquetes son la unidad fundamental del código compartido en R (Wickham, 2015)
- ▶ Los paquetes contienen funciones que permiten ampliar la capacidad de realizar operaciones con R
- ▶ Hay 14307 paquetes disponibles en el Comprehensive R Archive Network o CRAN.

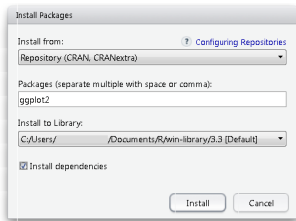


- ▶ Los paquetes son la unidad fundamental del código compartido en R (Wickham, 2015)
- ▶ Los paquetes contienen funciones que permiten ampliar la capacidad de realizar operaciones con R
- ▶ Hay 14307 paquetes disponibles en el Comprehensive R Archive Network o CRAN.
- ▶ Un paquete contiene las funciones, su documentación (archivos de ayuda) y las *vignettes* (guía de uso)
- ▶ Es necesario instalar los paquetes para usarlos.

# Usando los paquetes en R

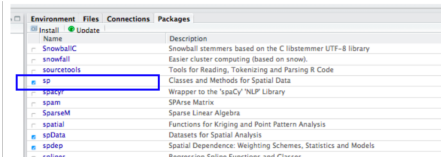
- Para instalar un paquete se usa el siguiente código

```
install.packages("ggplot2") #install the ggplot package
```



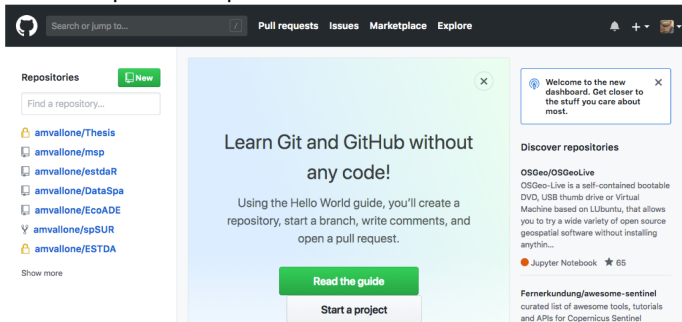
- Una vez instalado se debe cargar el paquete para utilizarlo.

```
library("sp")
```



# Usando los paquetes en R

- Existen paquetes que están en estado de desarrollo y se encuentran disponibles en repositorios públicos como GitHub



- Es posible descargar e instalar estos paquetes para su uso utilizando el paquete devtools

```
library("devtools")  
install_github("amvallone/estdaR")
```





Importando y exportando datos

# Importando y exportando información

- ▶ El proceso de importar y exportar información depende del formato en que la información
- ▶ Para importa y exportar es necesario entregar la información del *path* o configurar el directorio de trabajo

```
setwd("path")  
getwd()
```

-Los archivos separados por comas \*.csv

```
read.csv(file, header = TRUE, sep = ",", dec = ".", ...)  
write.csv(x, file, , sep = ",", dec = ".", ...)
```

- ▶ Para otro tipos de archivos de texto plano pueden usarse las funciones `read.table()` o `read.delim()`
- ▶ Para \*.xlsx es recomendable usar

```
library("openxlsx")  
read.xlsx(file, sheet = 1, startRow = 1,...)  
write.xlsx(x,file,...)
```

Importando el archivo mtcars.csv

```
data_cars <- read.csv("mtcars.csv", header=TRUE, sep=",")
```

Importando el archivo crime.txt

```
crime <- read.table("crime.txt", sep="\t", dec=".")
```

Importando el archivo muni17.xlsx

```
muni <- read.xlsx("muni17.xlsx")
```

# Otra forma de incorporar datos

- ▶ Otra forma es descargando los datos desde la red, incluso si los datos no están estructurados es posibles importarlos a R
- ▶ Nos interesa trabajar con la segunda tabla de datos de la esta pagina de Wikipedia.

## Ejemplo de web scraping

```
url <- "https://en.wikipedia.org/wiki/List_of_states_  
and_territories_of_the_United_States_by_population"  
require(rvest)  
raw_data <- html_table(read_html(url))  
data <- raw_data[[3]]
```



Filtrado de datos

- La forma más sencilla de filtrar información es utilizando indexación.

```
data(mtcars)
carb4 <- mtcars[mtcars$carb==4, ] # cars with 4 carb.
```

- Es posible realizar subconjuntos de datos que cumplan un conjunto de condiciones mediante la función subset

```
carb4 <- subset(mtcars, mtcars$cyl==4)
str(mtcars)
names(mtcars)
cyl <- mtcars$cyl #Extract the variable number of cylinders
```

¿Cuántos autos tiene 4 cilindros?

```
cyl4 <- mtcars[mtcars$cyl==4, ]  
cyl4_1 <- subset(mtcars, cyl==4)
```

¿Cuántos autos tienen 4 cilindros y mas de 90 caballos de fuerza?

¿Cuántos autos tiene 4 cilindros?

```
cyl4 <- mtcars[mtcars$cyl==4, ]  
cyl4_1 <- subset(mtcars, cyl==4)
```

¿Cuántos autos tienen 4 cilindros y mas de 90 caballos de fuerza?

```
cyl4_hp90 <- mtcars[mtcars$cyl==4 & mtcars$hp>90, ]  
cyl4_hp90_1 <- subset(mtcars, cyl==4 & hp>90)
```

¿Cuántos autos tienen más de 200 y menos de 60 caballos de fuerza?



¿Cuántos autos tiene 4 cilindros?

```
cyl4 <- mtcars[mtcars$cyl==4, ]  
cyl4_1 <- subset(mtcars, cyl==4)
```

¿Cuántos autos tienen 4 cilindros y mas de 90 caballos de fuerza?

```
cyl4_hp90 <- mtcars[mtcars$cyl==4 & mtcars$hp>90, ]  
cyl4_hp90_1 <- subset(mtcars, cyl==4 & hp>90)
```

¿Cuantos autos tienen más de 200 y menos de 60 caballos de fuerza?

```
hp60_200 <- mtcars[mtcars$hp>200 | mtcars$hp <60, ]  
hp60_200_1 <- subset(mtcars, hp>200 | hp <60)
```

¿Cuáles celdas contienen datos con autos con 6 cilindros?

```
which(mtcars$cyl==6)
```

```
## [1] 1 2 4 6 10 11 30
```

Realice una tabla cruzada de los autos teniendo en cuenta la cantidad de cilindros y el número de carburadores.

```
table(mtcars$cyl,mtcars$carb)
```

```
##
```

```
##      1 2 3 4 6 8
```

```
## 4 5 6 0 0 0 0
```

```
## 6 2 0 0 4 1 0
```

```
## 8 0 4 3 6 0 1
```

¿Cuál es la cilindrada media de los autos datdo el munero de caruradores?

```
aggregate(cyl ~carb, mtcars, mean)
```

```
##      carb      cyl
## 1      1 4.571429
## 2      2 5.600000
## 3      3 8.000000
## 4      4 7.200000
## 5      6 6.000000
## 6      8 8.000000
```

## Desafio 1

1. instale el paquete 'gapminder'
2. Use la ayuda para ver como usar y que contiene la base gapminder
3. conijmpruebe el tipo de dato de cada una de las variables de la base de datos
4. escoja una de las variables y tranformela de su estado original a caracteres (consejo: utilice alguna variable no numerica)
5. calcule la media del GDP per capita anual para cada continente

## Desafío 1

1. instale el paquete 'gapminder'
2. Use la ayuda para ver como usar y que contiene la base gapminder
3. conjmpruebe el tipo de dato de cada una de las variables de la base de datos
4. escoja una de las variables y tranformela de su estado original a caracteres (consejo: utilice alguna variable no numerica)
5. calcule la media del GDP per capita anual para cada continente

## Una posible solución

```
install.packages("gapminder")  
help(packages="gapmainder")  
str(gapminder) #también pude mirar la ventana del Global.  
cont <- as.character(gapminder$continent)  
aggregate(gdpPercap ~ year + continent, gapminder, mean )
```