



# Introducción a R

Yandira Cuvero

08 de marzo de 2022

# Contenido

# Contenido

- R, un lenguaje de programación.
- Conceptos básicos.
- Tipos de datos.
- Carga de objetos.

R un lenguaje de programación

# R core

- R es un lenguaje de programación y entorno computacional dedicado a la estadística.
- Es un lenguaje de programación porque nos permite dar instrucciones, usando código, a nuestros equipos de cómputo para que realicen tareas específicas.
- Cuando instalamos R en nuestra computadora en realidad lo que estamos instalando es el entorno computacional.

# Origen

- R tiene sus orígenes en S, un lenguaje de programación creado en los Laboratorios Bell de Estados Unidos.
- Ross Ihaka y Robert Gentleman, de la Universidad de Auckland en Nueva Zelanda, decidieron crear una implementación abierta y gratuita de S.
- La creación de R inició en 1992, teniendo una versión inicial del lenguaje en 1995 y en el 2000 una versión final estable.
- El mantenimiento y desarrollo de R es realizado por el R Development Core Team.
- Diariamente son publicados nuevos paquetes y sus respectivas actualizaciones.

# Descargar R

- R es un lenguaje de programación gratuito.
- CRAN es el The Comprehensive R Archive Network, una red en la que se archivan todas las versiones de R base, así como todos los paquetes para R que han pasado por un proceso de revisión riguroso, realizado por el CRAN Team, que se encarga de asegurar su correcto funcionamiento.
- CRAN es una red porque existen copias de su contenido en diferentes servidores alrededor del mundo, los cuales se actualizan diariamente.

# Descargar R

El sitio oficial de CRAN es <https://cran.r-project.org>

**The Comprehensive R Archive Network**

## **Download and Install R**

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#) ([Debian](#), [Fedora/Redhat](#), [Ubuntu](#))
- [Download R for macOS](#)
- [Download R for Windows](#)

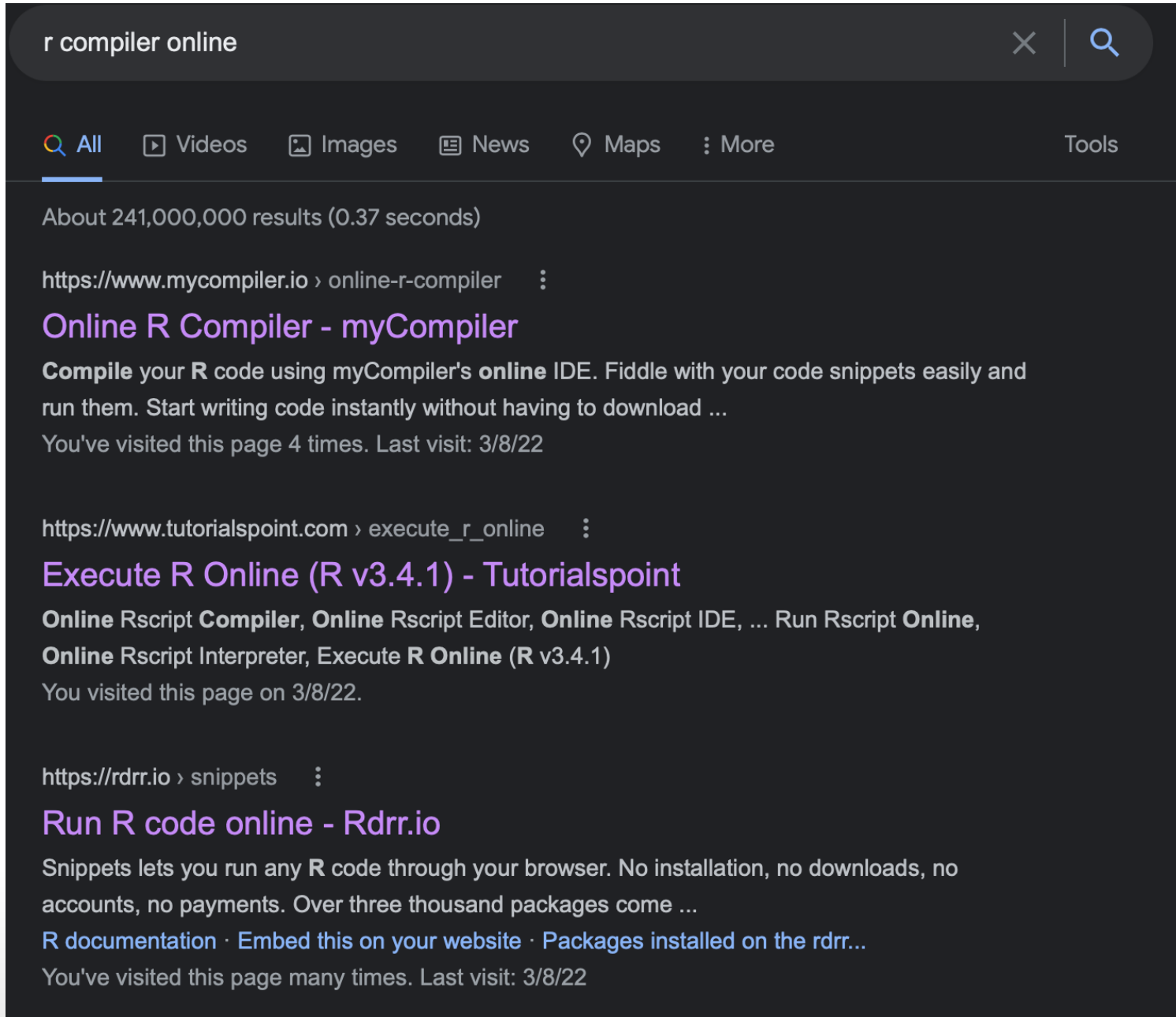
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.



# Un IDE para R

- Podemos utilizar R ejecutando nuestro código directamente desde documentos de texto plano.
- Un IDE nos proporciona herramientas para escribir y revisar nuestro código, administrar los archivos que estamos usando, gestionar nuestro entorno de trabajo y algunas otras herramientas de productividad.
- Hay varias opciones de IDE para R.

# Consolas en línea



A screenshot of a Google search interface with a dark theme. The search bar at the top contains the text "r compiler online" and has a close button (X) and a search button (magnifying glass). Below the search bar, there are navigation tabs: "All" (selected), "Videos", "Images", "News", "Maps", and "More". A "Tools" link is also visible on the right. The search results show "About 241,000,000 results (0.37 seconds)". The first result is from "https://www.mycompiler.io" with the title "Online R Compiler - myCompiler" in purple. The description says "Compile your R code using myCompiler's online IDE. Fiddle with your code snippets easily and run them. Start writing code instantly without having to download ...". It also notes "You've visited this page 4 times. Last visit: 3/8/22". The second result is from "https://www.tutorialspoint.com" with the title "Execute R Online (R v3.4.1) - Tutorialspoint" in purple. The description lists "Online Rscript Compiler, Online Rscript Editor, Online Rscript IDE, ... Run Rscript Online, Online Rscript Interpreter, Execute R Online (R v3.4.1)". It notes "You visited this page on 3/8/22". The third result is from "https://rdr.io" with the title "Run R code online - Rdr.io" in purple. The description says "Snippets lets you run any R code through your browser. No installation, no downloads, no accounts, no payments. Over three thousand packages come ...". It includes links for "R documentation", "Embed this on your website", and "Packages installed on the rdr.io". It notes "You've visited this page many times. Last visit: 3/8/22".

r compiler online

All Videos Images News Maps More Tools

About 241,000,000 results (0.37 seconds)

https://www.mycompiler.io › online-r-compiler

**Online R Compiler - myCompiler**

**Compile** your **R** code using myCompiler's **online** IDE. Fiddle with your code snippets easily and run them. Start writing code instantly without having to download ...

You've visited this page 4 times. Last visit: 3/8/22

https://www.tutorialspoint.com › execute\_r\_online

**Execute R Online (R v3.4.1) - Tutorialspoint**

**Online** Rscript **Compiler**, **Online** Rscript Editor, **Online** Rscript IDE, ... Run Rscript **Online**, **Online** Rscript Interpreter, Execute **R** **Online** (R v3.4.1)

You visited this page on 3/8/22.

https://rdr.io › snippets

**Run R code online - Rdr.io**

Snippets lets you run any **R** code through your browser. No installation, no downloads, no accounts, no payments. Over three thousand packages come ...

[R documentation](#) · [Embed this on your website](#) · [Packages installed on the rdr...](#)

You've visited this page many times. Last visit: 3/8/22

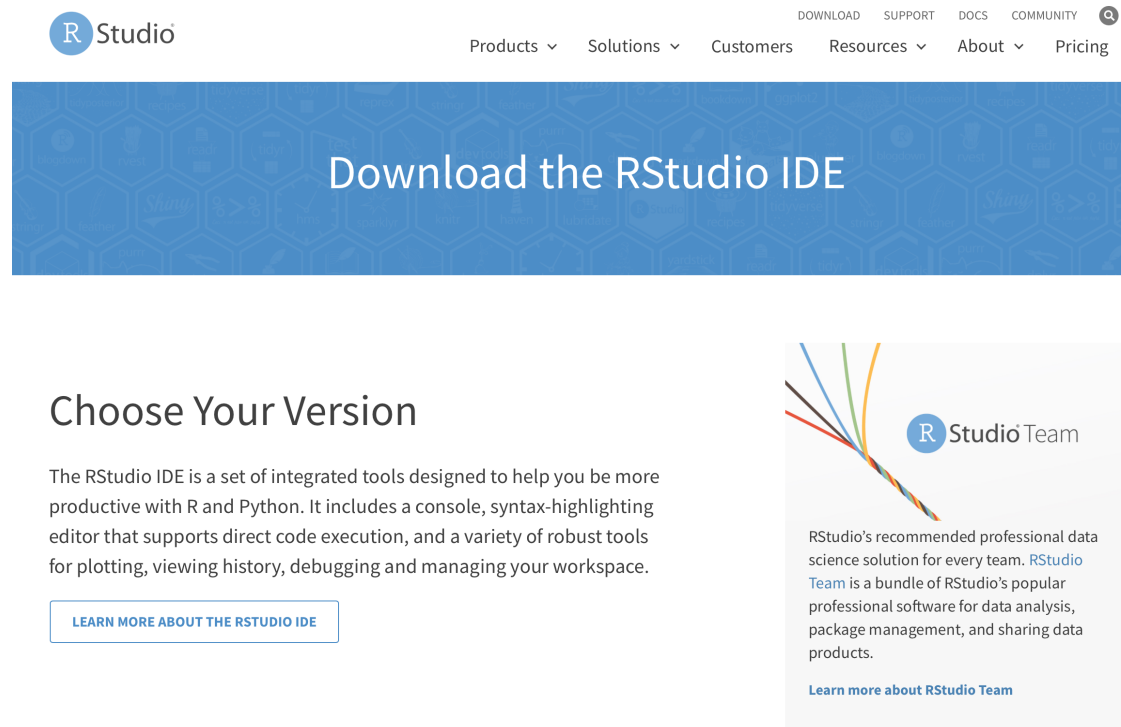
# R Studio

- RStudio es desarrollado por un equipo que ha contribuido de manera significativa para lograr que R sea lenguaje de programación más accesible, con un énfasis en la colaboración y la reproducción de los análisis.
- Permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R.

# Instalar R Studio

Para instalar RStudio, es necesario con descargar y ejecutar alguno de los instaladores disponibles en su sitio oficial

<https://www.rstudio.com/products/rstudio/download/>



The screenshot shows the RStudio website's download page. At the top, the RStudio logo is on the left, and navigation links for DOWNLOAD, SUPPORT, DOCS, and COMMUNITY are on the right. Below these, a horizontal menu contains Products, Solutions, Customers, Resources, About, and Pricing. A large blue banner with a hexagonal pattern in the background reads "Download the RStudio IDE". Below the banner, the section "Choose Your Version" is displayed. It contains a paragraph describing the RStudio IDE as a set of integrated tools for R and Python, and a button labeled "LEARN MORE ABOUT THE RSTUDIO IDE". To the right of this text is a card for "RStudio Team". The card features the RStudio logo and the text "RStudio Team" at the top. Below this, it describes RStudio Team as a professional data science solution for every team, a bundle of RStudio's popular professional software for data analysis, package management, and sharing data products. At the bottom of the card is a link that says "Learn more about RStudio Team".

RStudio

DOWNLOAD SUPPORT DOCS COMMUNITY

Products Solutions Customers Resources About Pricing

## Download the RStudio IDE

### Choose Your Version

The RStudio IDE is a set of integrated tools designed to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.

[LEARN MORE ABOUT THE RSTUDIO IDE](#)

**RStudio Team**

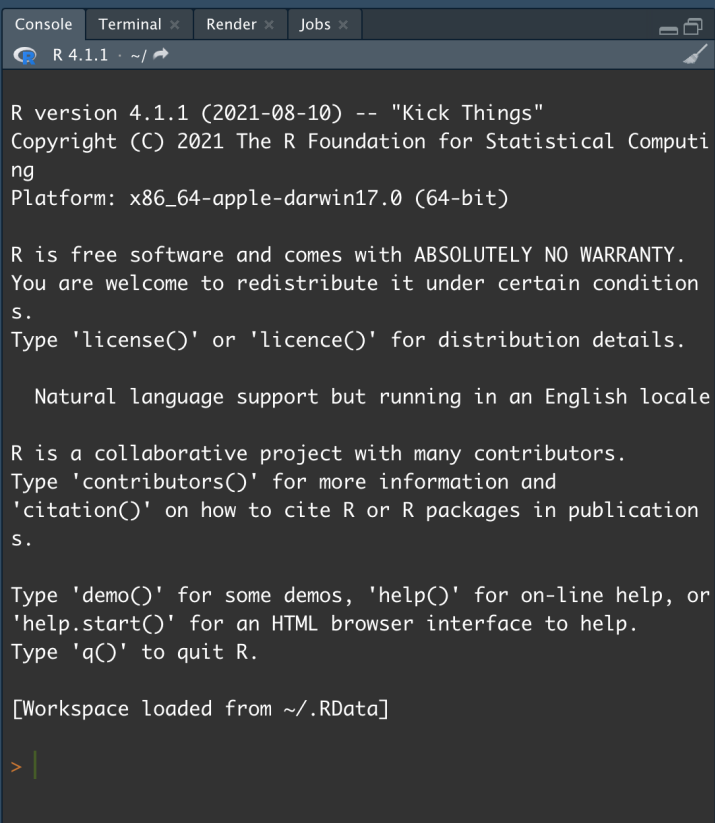
RStudio's recommended professional data science solution for every team. **RStudio Team** is a bundle of RStudio's popular professional software for data analysis, package management, and sharing data products.

[Learn more about RStudio Team](#)

Conceptos básicos

# Consola de R

- Corresponde al entorno computacional de este lenguaje.
- Es aquí donde nuestro código es interpretado.



```
Console Terminal x Render x Jobs x
R 4.1.1 · ~/

R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

Natural language support but running in an English locale

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

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

[Workspace loaded from ~/.RData]

> |
```

# R como calculadora

Podemos escribir código directamente en la consola

```
3*5
```

```
## [1] 15
```

```
12.3/2.6
```

```
## [1] 4.730769
```

```
cos(0)
```

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

# Variables

- En R usamos `<-` o `=` para hacer asignaciones o definir *variables*.
- Utilizamos `#` para escribir comentarios.

```
a <- 3*5  
a
```

```
## [1] 15
```

```
b = 2+6; b
```

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

```
# Comentario
```



# Funciones

- Una función es una serie de operaciones a la que les hemos asignados un nombre.
- En R reconocemos a una función usando la notación:  
*nombre\_de\_la\_función()*

```
cos(0)
```

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

```
sqrt(4)
```

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

```
sin( pi )
```

```
## [1] 1.224647e-16
```

# Documentación

- Para saber como trabajar con una función podemos ver su *documentación*.
- La documentación describe qué hace la función, sus argumentos, detalles sobre las operaciones que realiza, los resultados que devuelve y ejemplos de uso.

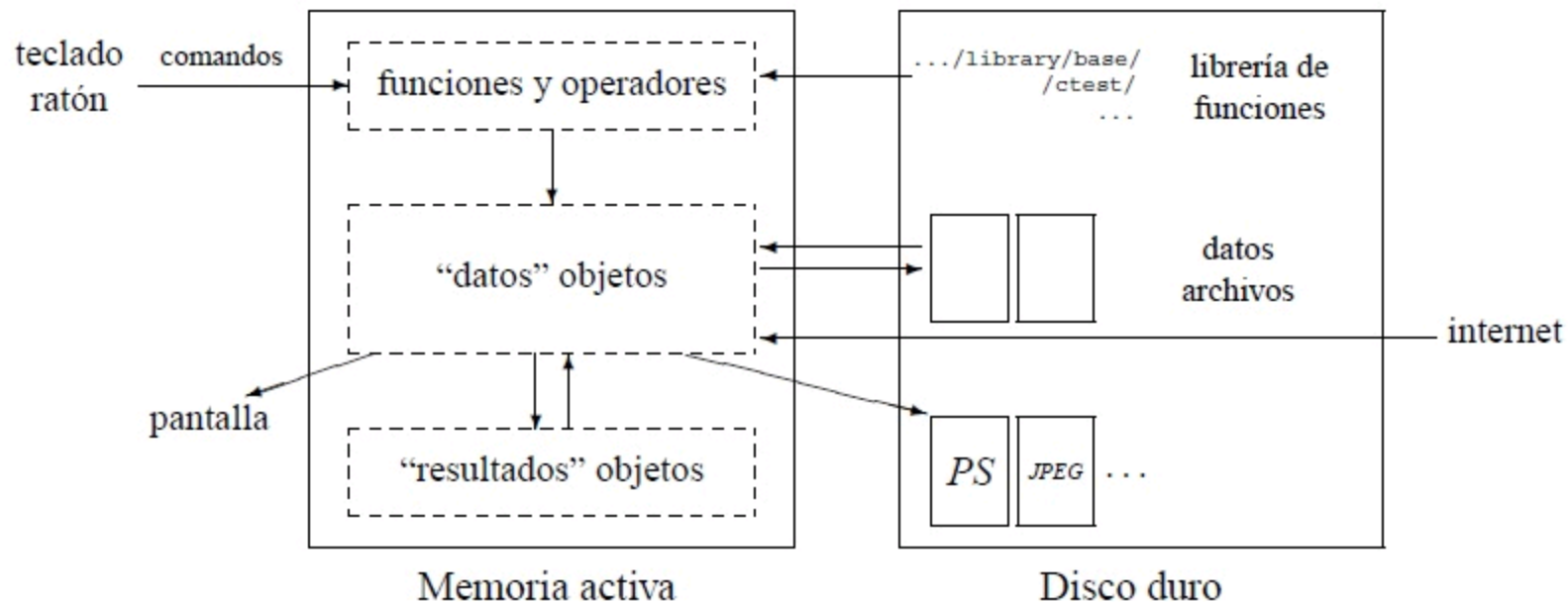
```
?pnorm(1.75)  
help("pnorm")
```

- En R, todo es un objeto.
- Todos los datos y estructuras de datos son objetos.
- Todos los objetos tienen un nombre para identificarlos.

# R orientado a objetos

# ¿Cómo funciona R?

Estructura de datos en R.

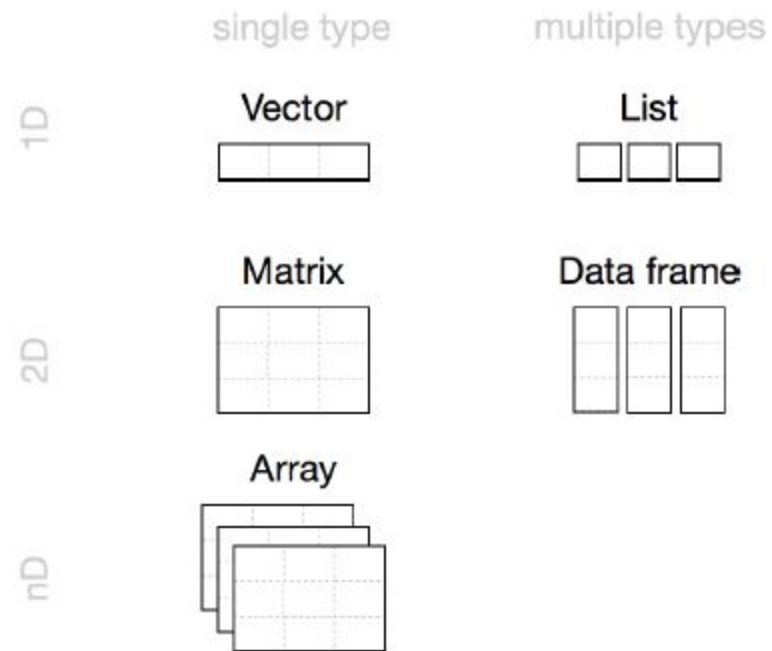


Tipos de datos

# Estructura de datos

Las estructuras de datos en R se organizan por:

- Dimensionalidad y
- Homogeneidad o heterogeneidad.



# Vectores

- Los elementos de un vector son del mismo tipo.
- Un vector es creado mediante la función combinar `c()`.

```
a = 3  
vec = c(1,a)  
vec
```

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

# Tipos de Vectores

Vector double:

```
dbl_vec <- c(3.5, 2, -1)
```

Vector entero: El sufijo L para crear un vector entero;

```
int_vec <- c(3L, 7L, 1L)
```



# Tipos de Vectores

Vector character: Se usa " para crear un vector texto ;

```
chr_vec <- c("R", "Users", "Group")
```

Vector lógico: Se utiliza TRUE y FALSE o T y F para crear un vector lógico.

```
log_vec <- c(FALSE, TRUE, F, T)
```

# Elementos de un Vector

La componente  $k$  del objeto *vec* se obtiene mediante *vec[k]*.

```
vec <- c(6, 1, 3, 6, 10, 5)  
vec[5]
```

```
## [1] 10
```

Para seleccionar varios elementos utilizamos *vec[c(elementos)]*.

```
vec[c(2, 4)]
```

```
## [1] 1 6
```

# Elementos de un vector

Para omitir el elemento  $k$  de  $\text{vec}$  se utiliza  $\text{vec}[-k]$ .

```
vec[-5]
```

```
## [1] 6 1 3 6 5
```

# Generación de vectores como sucesiones

El operador  $a:b$  genera el vector  $a, a + 1, a + 2, \dots, b$ .

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
5:-5
```

```
## [1] 5 4 3 2 1 0 -1 -2 -3 -4 -5
```

# Generación de vectores como sucesiones

La función `seq( )` genera secuencias controlando: inicio, fin y salto.

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

```
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

# Matrices

Una matriz es un vector con el atributo dimensión `dim`. El atributo `dim` es un vector de longitud `2:c(nrow, ncol)`. Se construye por columnas por defecto (`byrow=FALSE`).

```
mtx <- matrix (1:12,nrow=3, ncol=4, byrow=FALSE)
mtx
```

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

# Listas

Una lista es un vector que puede contener elementos de cualquier tipo y de distinta longitud.

Para crear una lista se utiliza la función *list()* en lugar de *c()*.

```
lst <- list(c(1, 2), c(TRUE), c("a", "b", "c"))  
lst
```

```
## [[1]]  
## [1] 1 2  
##  
## [[2]]  
## [1] TRUE  
##  
## [[3]]  
## [1] "a" "b" "c"
```

# Data Frame

- Es una lista en la cual todos los elementos tienen la misma longitud.
- A diferencia de las matrices, pueden almacenar vectores de cualquier tipo.
- Presenta varios atributos adicionales como: class, rownames y names.
- Es la estructura de datos más utilizada para almacenar data tabulada.

data frame	1	"R"	TRUE
	2	"S"	FALSE
	3	"T"	TRUE
	numeric	character	logical



# Dataframe

Para crear un data frame se utiliza la función *data.frame()*. Con los siguientes vectores:

```
ingreso <- c(2020, 2015, 2018)
nombre <- c("Rubi", "Saul", "Tomas")
instrumento <- c("Guitarra", "Piano", "Violin")
df <- data.frame(ingreso, nombre, instrumento)
df
```

```
## ingreso nombre instrumento
## 1    2020   Rubi    Guitarra
## 2    2015   Saul      Piano
## 3    2018  Tomas    Violin
```

# Elementos de un data frame:

Mediante `df[i, j]` se obtiene la componente *i, j* del data frame.

```
df[2, 2] # componente 2, 2
```

```
## [1] "Saul"
```

```
df[3, 1] # componente 3, 1
```

```
## [1] 2018
```

# Elementos de un data frame:

```
df[3, ] # fila 3
```

```
## ingreso nombre instrumento  
## 3      2018  Tomas      Violin
```

```
df[c(1, 4), ] # filas 1, 4
```

```
## ingreso nombre instrumento  
## 1      2020  Rubi      Guitarra  
## NA      NA  <NA>      <NA>
```

# Elementos de un data frame:

```
df[ , "ingreso"] # equivalente a df[ , 2]
```

```
## [1] 2020 2015 2018
```

```
df[ , c("ingreso", "nombre")] # equivalente a df[ , c(1, 2)]
```

```
## ingreso nombre  
## 1      2020   Rubi  
## 2      2015   Saul  
## 3      2018  Tomas
```

Paquetes

# Instalación de paquetes

- Utilizamos la función `install.packages`
- Se indica el nombre del paquete
- Se indica si se deben instalar o no los paquetes requeridos para que este funcione.

```
install.packages("nombre_del_paquete", dependencies = TRUE)
```

- Los paquetes se instalan una sola vez y deben ser cargados en cada inicio de sesión.
- Utilizamos la función *library* o *require*

```
library(nombre_del_paquete)  
require(nombre_del_paquete)
```

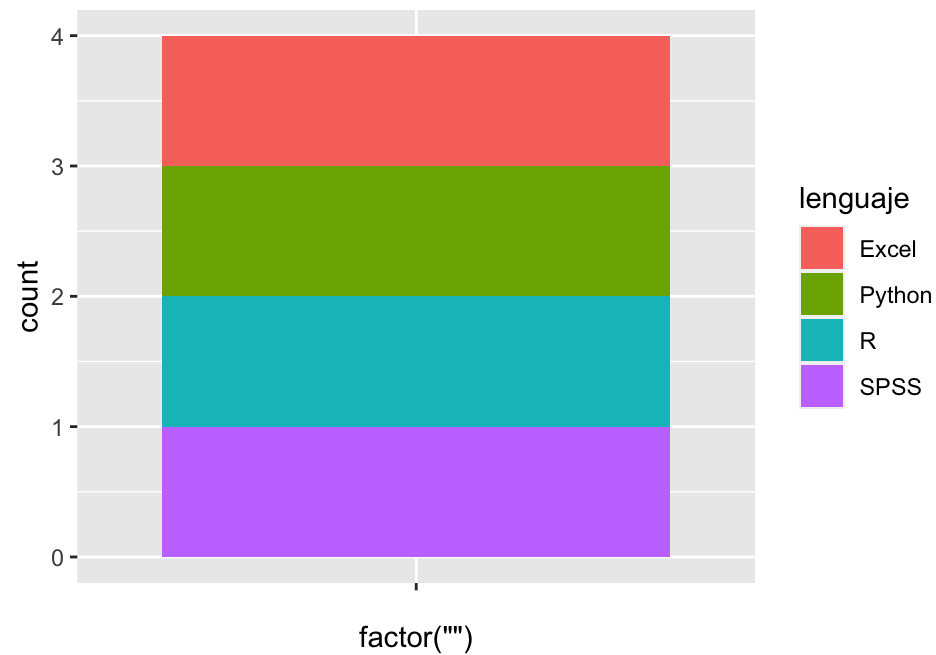
# Ejemplo

```
#install.packages("ggplot2",dependencies = TRUE)  
library(ggplot2)  
#??ggplot2
```

Ahora podemos realizar un gráfico.

# Ejemplo

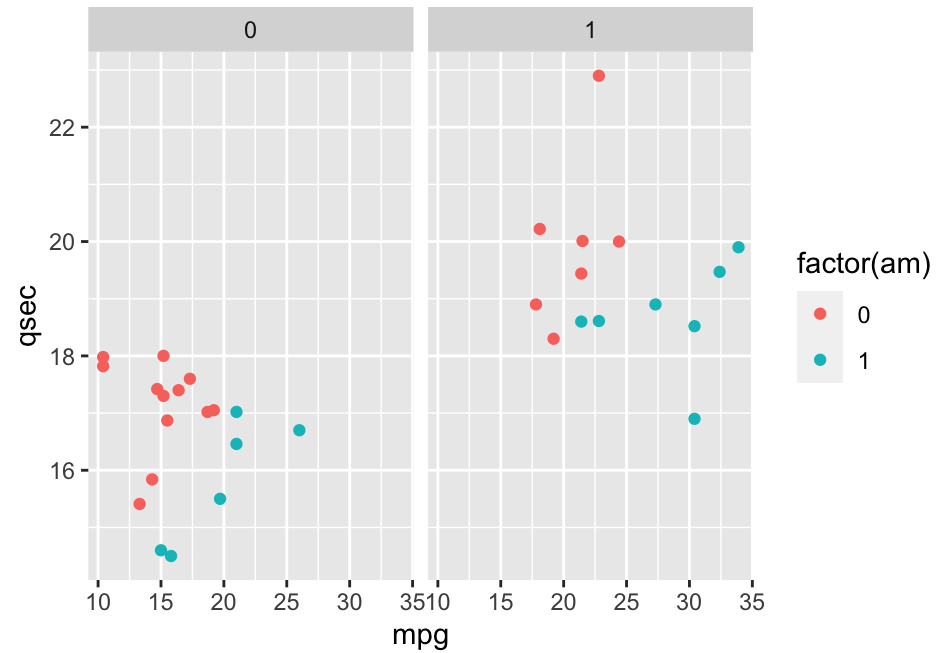
```
df <- data.frame( a = c(60,30,8, 2),  
  lenguaje=c('R','Python','Excel', 'SPSS'))  
ggplot(df, aes(x = factor(""),  
  fill = lenguaje) ) +  
  geom_bar()
```





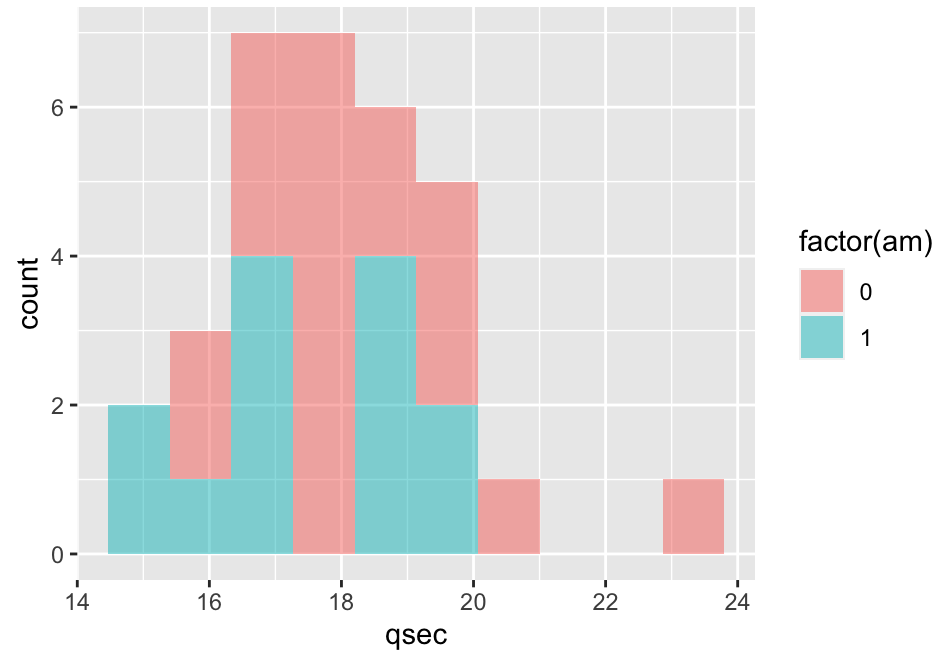
# Ejemplo

```
ggplot(data = mtcars) +  
  geom_point(aes(mpg,  
    qsec, colour = factor(am))) +  
  facet_grid(~vs)
```



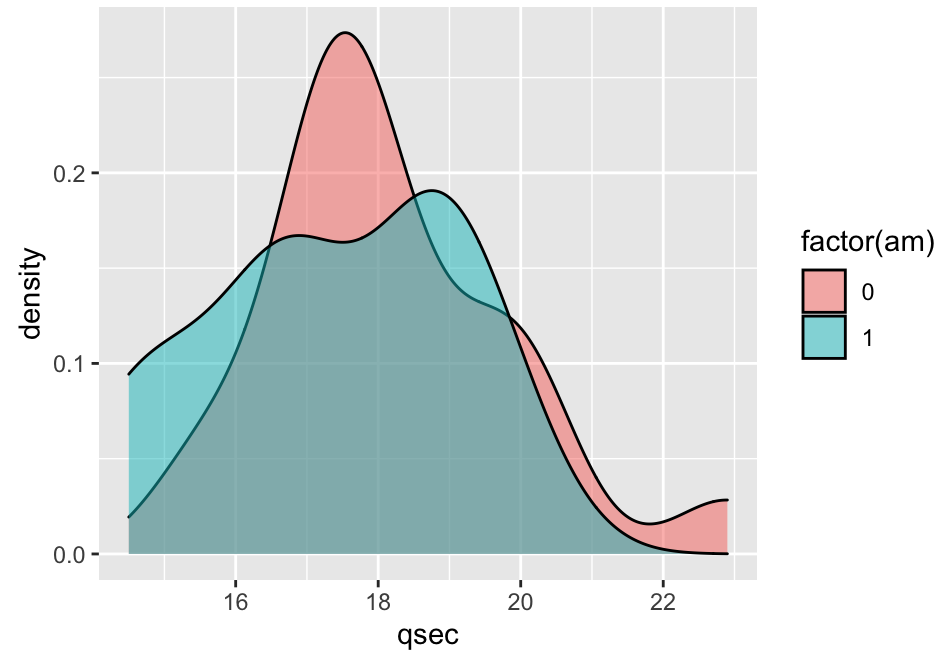
# Ejemplo

```
ggplot(data = mtcars) +  
  geom_histogram(aes(x=qsec,  
    fill=factor(am)),bins=10,  
    position = "stack",alpha = 0.5)
```



# Ejemplo

```
ggplot(data = mtcars) +  
  geom_density(aes(x=qsec,  
    fill=factor(am)),bins=10,  
  position = "identity",alpha = 0.5)
```



Cargar archivos

# Lectura de datos

R permite una amplia variedad de acceso de archivos:

- Archivos de excel .xls, .xlsx, .csv;
- Archivos de texto plano .txt;
- Archivos de spss .sav;
- Archivos de la web;
- Archivos de bases de datos, etc.

# Directorio de trabajo

- Working directory (wd). Es la dirección donde se almacenan, leen y escriben los archivos utilizados y generados mediante R.
- `getwd()` permite obtener el wd actual:

```
getwd()
```

```
## [1] "/Users/yandiracuvero/Desktop/IntroR2022"
```

# Directorio de trabajo

setwd() permite setear un nuevo wd, por ejemplo:

```
setwd("/Users/yandiracuvero/Desktop/IntroR2022/")
```

# Directorio de trabajo

`dir()` enlista los nombres de los archivos en el wd actual.

```
dir()
```

```
## [1] "coercionR.png"      "combinarR.png"      "comoFunciona.png"
## [4] "dataFrameR.png"    "elementosDF.png"    "EPN.png"
## [7] "factorR.png"       "fig01R.png"         "fig02.png"
## [10] "fig03.png"         "fig04.png"          "form11.jpg"
## [13] "hip.jpg"           "IntroduccionR_files" "IntroduccionR.html"
## [16] "IntroduccionR.Rmd" "leeArchivosR.png"   "nyc.csv"
## [19] "objetosR.png"      "ScriptManana.R"     "TIIE_RUGE_u1"
## [22] "TIIE_RUGE_u2"      "TIIE_RUGE_u3"
```



# ¿Cómo leer archivos en R?

The diagram shows a data file structure with three columns: variable.1, variable.2, and variable.n. The data is organized into rows, with the first row being a header. The separator character is a vertical bar (|), and the decimal character is a comma (,). The value 4,1 in the third column is highlighted in yellow.

variable.1	variable.2	variable.n
25	"Sierra"	1,5
24	"Costa"	0
24	"Costa"	7
27	"Sierra"	4,1
.	.	.
.	.	.
.	.	.
20	"Amazonia"	-1,9
28	"Amazonia"	2,3
20	"Insular"	6,2

# Archivos .txt

Se requiere leer un archivo en formato txt: archivo.txt.

Se utiliza la función `read.table()`

Parámetros: - file: nombre del archivo (incluida extensión); - sep: caracter utilizado para separar columnas (variables); - dec: caracter utilizado para decimales; - header: TRUE, si la primera fila contiene los nombres de las columnas. `str()` describe la estructura de datos.

# Archivo .txt

```
data_txt <- read.table(file = "archivo.txt", sep = "nt",  
dec = ",", header = TRUE)  
str(data_txt)
```

# Lectura de archivos .csv

- Se requiere leer un archivo en formato csv: **archivo.csv**
- Se utiliza la función `read.csv` que presenta por defecto los argumentos:
- `sep = ","`,
- `dec = "."`,
- `header = TRUE`

```
data_csv <- read.csv(file = "archivo.csv")
```

# Ejemplo

```
nyc <- read.csv(file = "nyc.csv")  
head(nyc)
```

##	Case	Restaurant	Price	Food	Decor	Service	East
## 1	1	Daniella Ristorante	43	22	18	20	0
## 2	2	Tello's Ristorante	32	20	19	19	0
## 3	3	Biricchino	34	21	13	18	0
## 4	4	Bottino	41	20	20	17	0
## 5	5	Da Umberto	54	24	19	21	0
## 6	6	Le Madri	52	22	22	21	0

# Ejemplo

```
mean(nyc$Price)
```

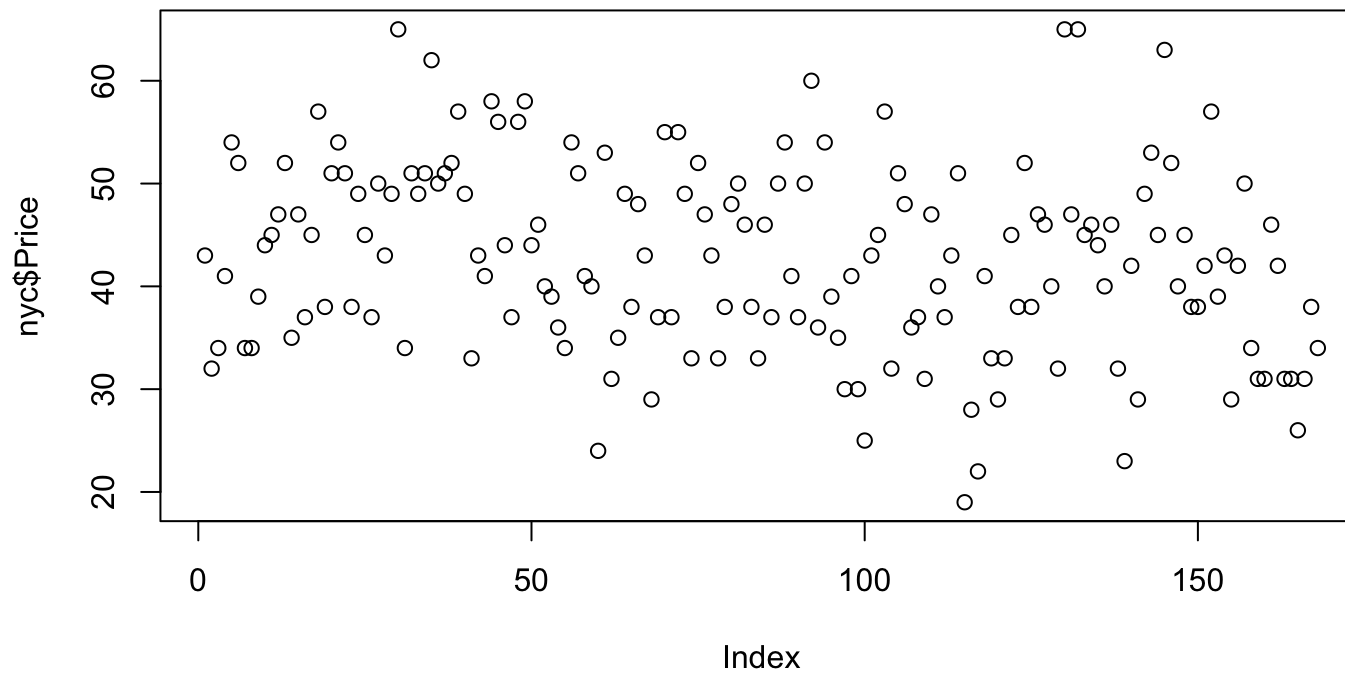
```
## [1] 42.69643
```

```
sd(nyc$Price)
```

```
## [1] 9.292814
```

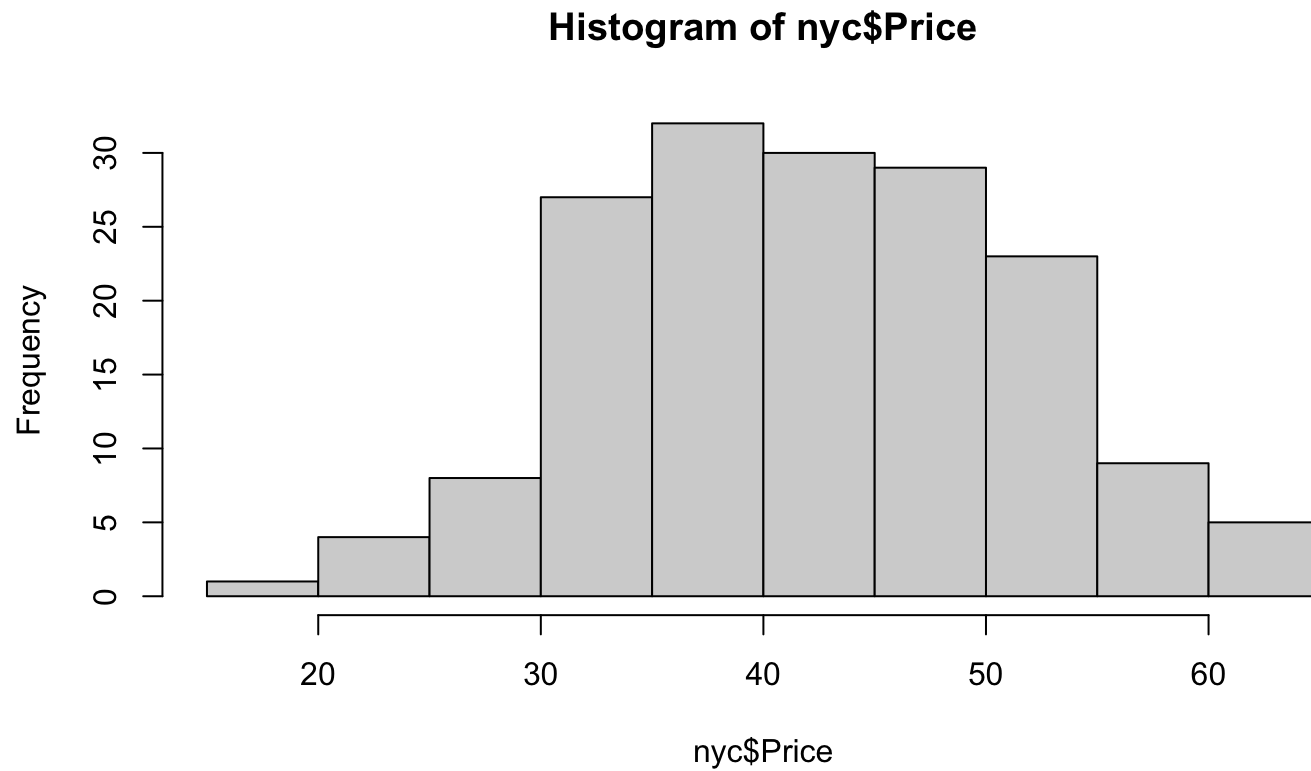
# Ejemplo

```
plot(nyc$Price)
```



# Ejemplo

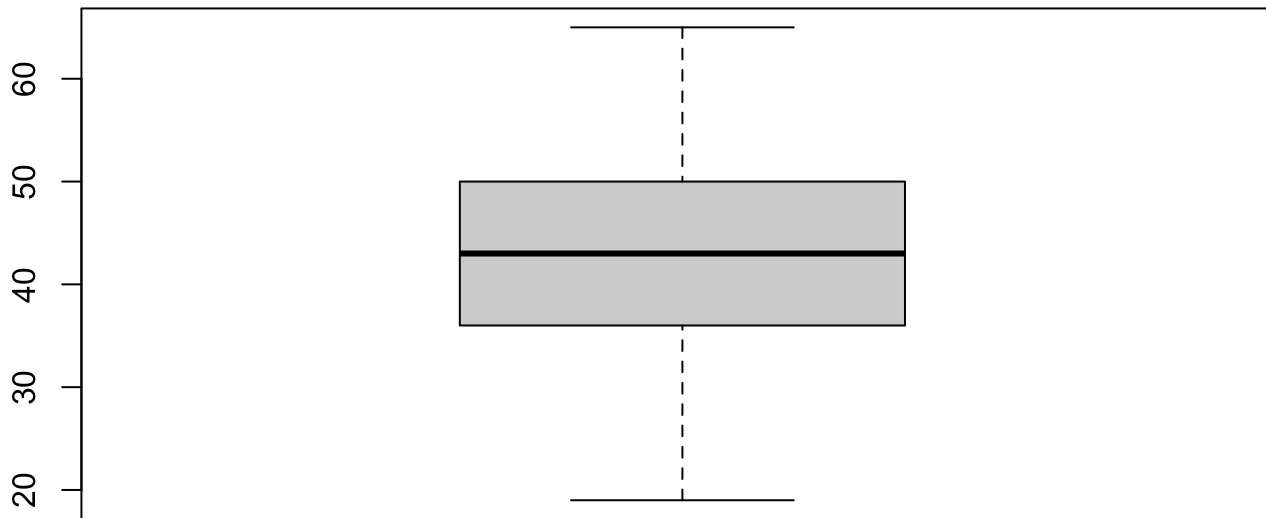
```
hist(nyc$Price)
```





# Ejemplo

```
boxplot(nyc$Price)
```



Gracias