

# Técnicas y Herramientas Modernas

Grupo: Trabajo Aprobado

2025-04-23

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Si quiero saber que significa la tabla de valores al colocar “cars” debo poner en la consola ? cars y me da información y comandos que puedo ejecutar a modo de ejemplos

¿Cómo hago para ver sólo la columna de las velocidades?

```
cars$speed
```

```
## [1] 4 4 7 7 8 9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15
## [26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 22 23 24 24 24 24 25
```

Si quiero ver estadísticas básicas de una variable o de una tabla, tengo que usar el comando summary

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

El comando summary muestra los valores estadísticos básicos (máximo, mínimo, mediana, etc.) a partir de la tabla de velocidad y distancia

```
# Toda línea que en el código del programa aparece después de un numeral, es un comentario
# Rutina para convertir de pies a metros
```

```
distancia = cars$dist*0.31
```

Cómo contar la cantidad de filas de una columna:

```
length(cars$dist)
```

```
## [1] 50
```

Cómo calcular el promedio de las velocidades

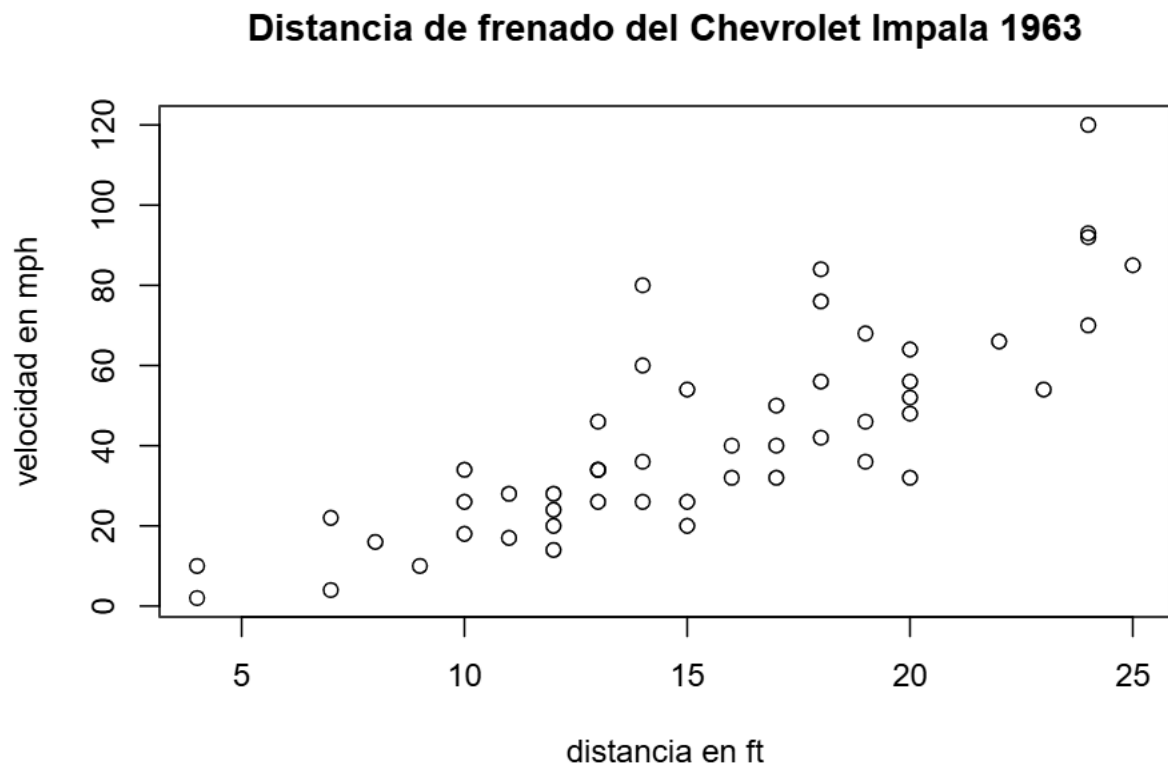
```
mean(cars$speed)
```

```
## [1] 15.4
```

```
#mode(cars$speed)
```

Cómo hacer un gráfico de velocidad vs distancia:

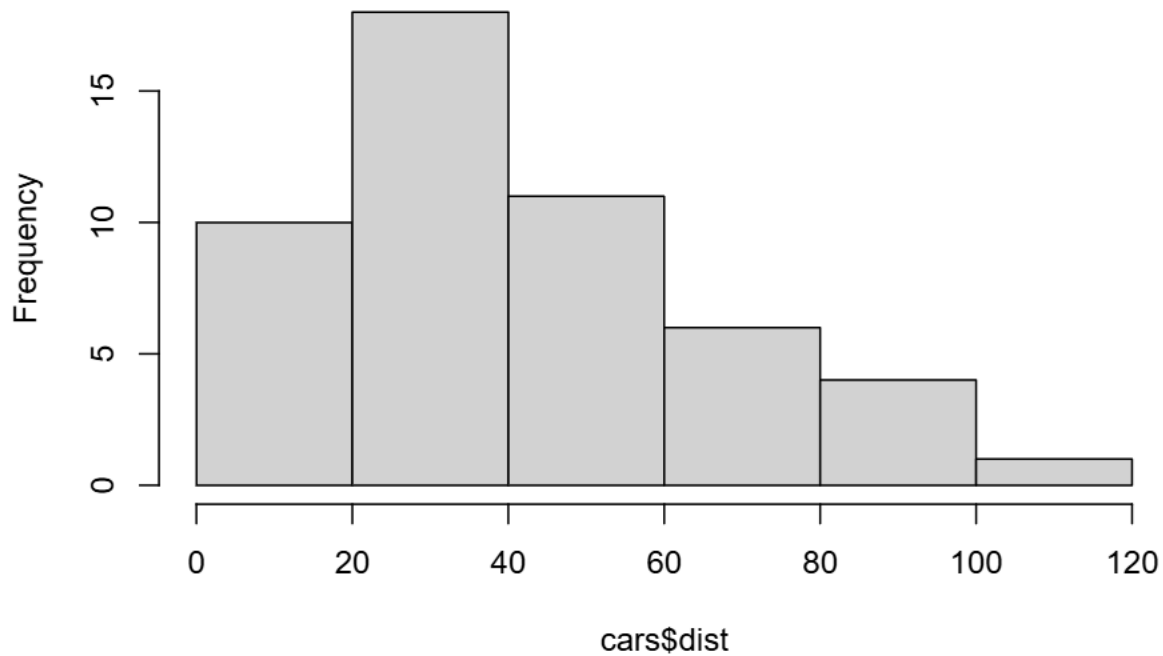
```
plot(cars,main="Distancia de frenado del Chevrolet Impala 1963",xlab="distancia en ft",ylab="velocidad en mph")
```



Histogramas

```
hist(cars$dist)
```

**Histogram of cars\$dist**



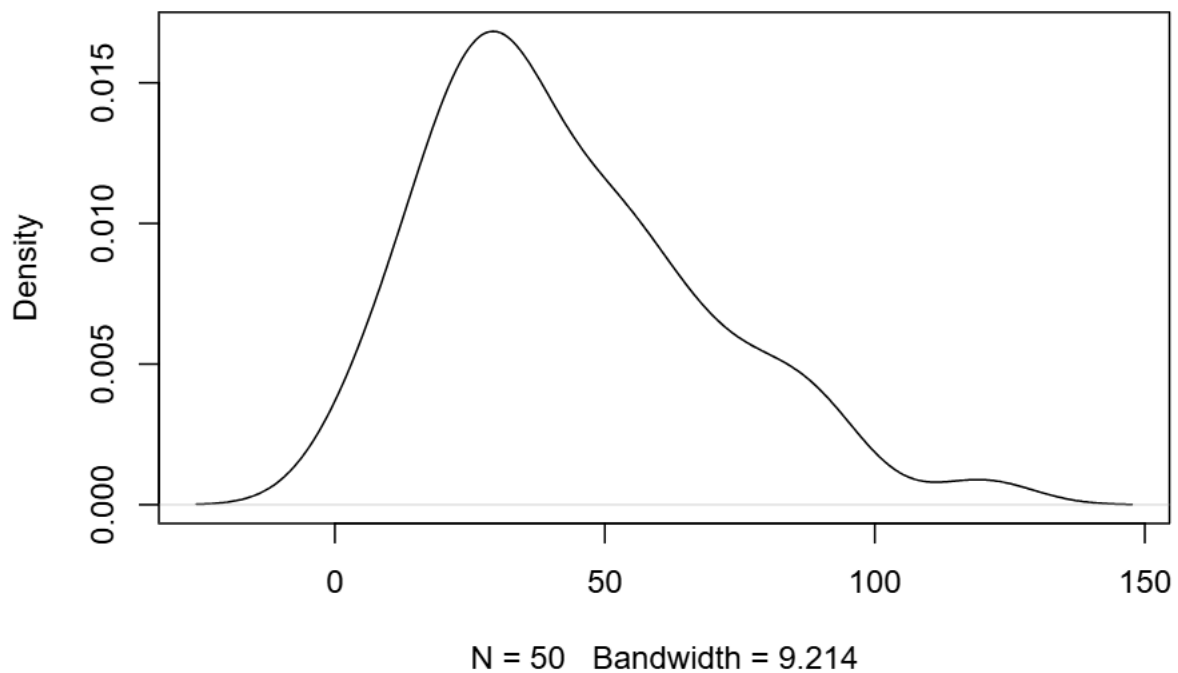
Aquí

también se pueden colocar los títulos y nombres a los ejes

Gráfico de densidad

```
plot(density(cars$dist))
```

**density(x = cars\$dist)**



Cómo poner una flecha de asignación de valores a una variable:

```
a <- 23
a
```

```
## [1] 23
```

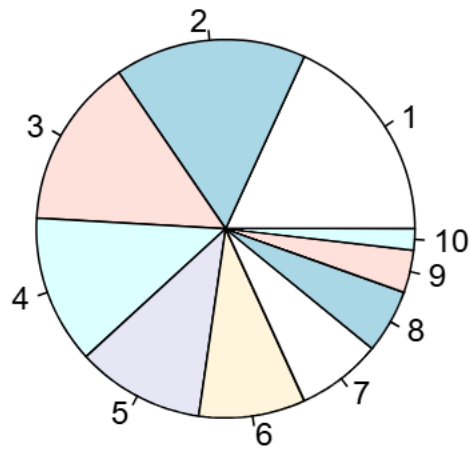
Cómo convertir una variable en un vector:

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

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

Para hacer un gráfico de tortas:

```
pie(b)
```



```
library(readr)
Puertos_Chile <- read_csv("https://themys.sid.uncu.edu.ar/rpalma/R-cran/Puertos_Chile.csv")
```

```
## Rows: 150 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (1): Puerto
## dbl (5): F, Tecnologia, Normas, Seguridad, Equipo
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Puertos_Chile
```

```
## # A tibble: 150 x 6
##       F Tecnologia Normas Seguridad Equipo Puerto
##   <dbl>      <dbl> <dbl>      <dbl>  <dbl> <chr>
## 1     1         5.1   3.5         1.4    0.2 Iqui
## 2     2         4.9    3         1.4    0.2 Iqui
## 3     3         4.7   3.2         1.3    0.2 Iqui
## 4     4         4.6   3.1         1.5    0.2 Iqui
## 5     5         5       3.6         1.4    0.2 Iqui
## 6     6         5.4   3.9         1.7    0.4 Iqui
## 7     7         4.6   3.4         1.4    0.3 Iqui
## 8     8         5       3.4         1.5    0.2 Iqui
## 9     9         4.4   2.9         1.4    0.2 Iqui
## 10    10         4.9   3.1         1.5    0.1 Iqui
```

```
## # i 140 more rows
```

```
library(readxl)
Tecnicas_y_herramientas_excel <- read_excel("Tecnicas y herramientas excel.xlsx",
  col_types = c("date", "numeric", "numeric",
    "numeric", "numeric"))
```

```
## Warning: Expecting numeric in B27 / R27C2: got '339 2.0'
```

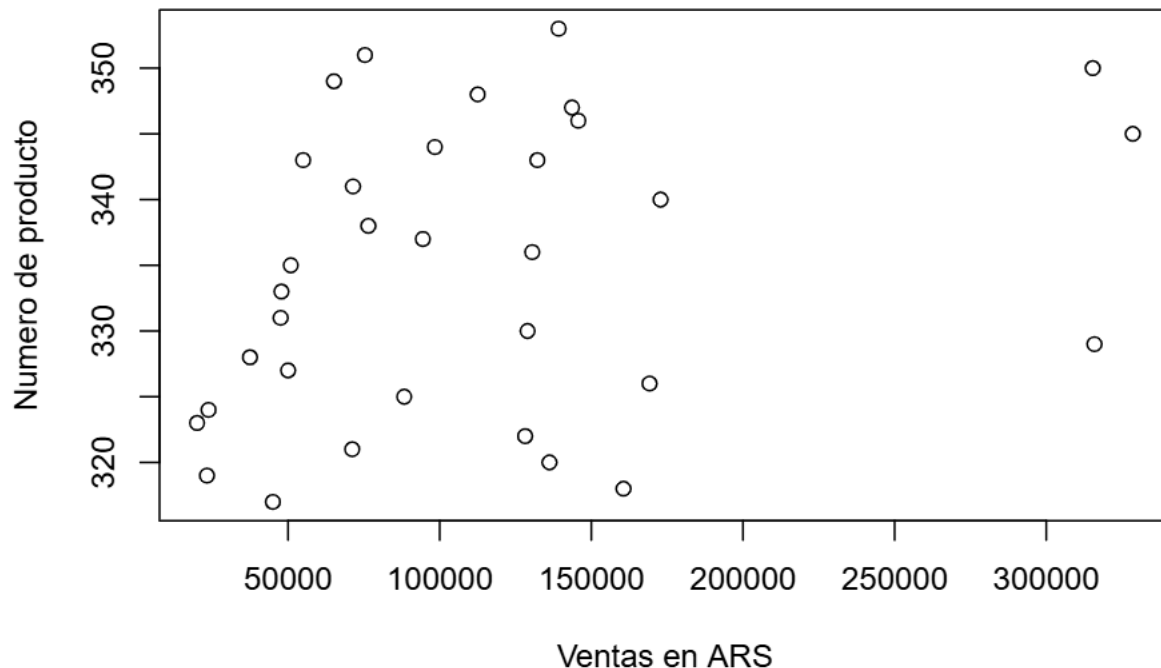
```
Tecnicas_y_herramientas_excel
```

```
## # A tibble: 34 x 5
```

```
##   Fecha                `Venta de:` `Precio costo` `Precio venta` Ganancia
##   <dtm>                <dbl>        <dbl>        <dbl>        <dbl>
## 1 2024-01-09 00:00:00      319          7000        23269.    16269.
## 2 2024-01-09 00:00:00      343         40000       132200    16269.
## 3 2024-01-11 00:00:00      317         25000        45000    92200
## 4 2024-01-11 00:00:00      321         48000        71160    20000
## 5 2024-01-12 00:00:00      323         15000        20000    23160
## 6 2024-01-16 00:00:00      320         70000       136186     5000
## 7 2024-01-16 00:00:00      318        106900       160600     5900
## 8 2024-01-16 00:00:00      322         75000       128170    66186
## 9 2024-01-16 00:00:00      326         40000       169200    53700
## 10 2024-01-16 00:00:00     324         10000        23780    53170
## # i 24 more rows
```

```
plot(Tecnicas_y_herramientas_excel$`Precio venta`,Tecnicas_y_herramientas_excel$`Venta de:`,main="Tecni
```

## Tecnicas y herramientas modernas



## EJERCICIO DE MICROBENCHMARK

Para la clase que viene

```

library(microbenchmark)
set.seed(2017)
n <- 10000
p <- 100
X <- matrix(rnorm(n*p), n, p)
y <- X %*% rnorm(p) + rnorm(100)
check_for_equal_coefs <- function(values) {
  tol <- 1e-12
  max_error <- max(c(abs(values[[1]] - values[[2]]),
    abs(values[[2]] - values[[3]]),
    abs(values[[1]] - values[[3]])))
  max_error < tol
}
mbm <- microbenchmark("lm" = { b <- lm(y ~ X + 0)$coef },
  "pseudoinverse" = {
    b <- solve(t(X) %*% X) %*% t(X) %*% y
  },
  "linear system" = {
    b <- solve(t(X) %*% X, t(X) %*% y)
  },
  check = check_for_equal_coefs)
mbm

```

```

## Unit: milliseconds
##      expr      min       lq      mean    median      uq      max neval
##      lm  35.63215  99.84405 147.1790 140.8953 209.8997 302.3286   100
## pseudoinverse 153.43436 300.67107 452.0862 401.3330 520.3132 1302.3193   100
## linear system  92.90494 201.15541 326.4535 301.1359 400.0800 1598.3024   100

```

## Clase 30/04/2025

Problema: Penitencia de Newton Hallar un algoritmo para sumar todos los números comprendidos entre 1 y 100

```

n <- 100
suma <- (n * (n + 1)) / 2
print(paste("La suma es:", suma))

```

```
## [1] "La suma es: 5050"
```