Primeros pasos en R

Regresiones lineales

Josu Najera Zuloaga

jnajera@deusto.es

Universidad de Deusto

Programa Big Data y Business Intelligence

Contents

1. Instalación de R y Rstudio
2. Importar datos
3. Análisis preliminar
   - Introducción
   - Medidas analíticas
   - Visualización
4. Regresión Lineal

Table of Contents

1. Instalación de R y Rstudio
2. Importar datos
3. Análisis preliminar
   - Introducción
   - Medidas analíticas
   - Visualización
4. Regresión Lineal

Introducción: R

R es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

R nació como una reimplementación de software libre del lenguaje S, adicionado con soporte para ámbito estático. Se trata de uno de los lenguajes de programación más utilizados en investigación científica, siendo además muy popular en los campos de aprendizaje automático (machine learning), minería de datos, investigación biomédica, bioinformática y matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y graficación.

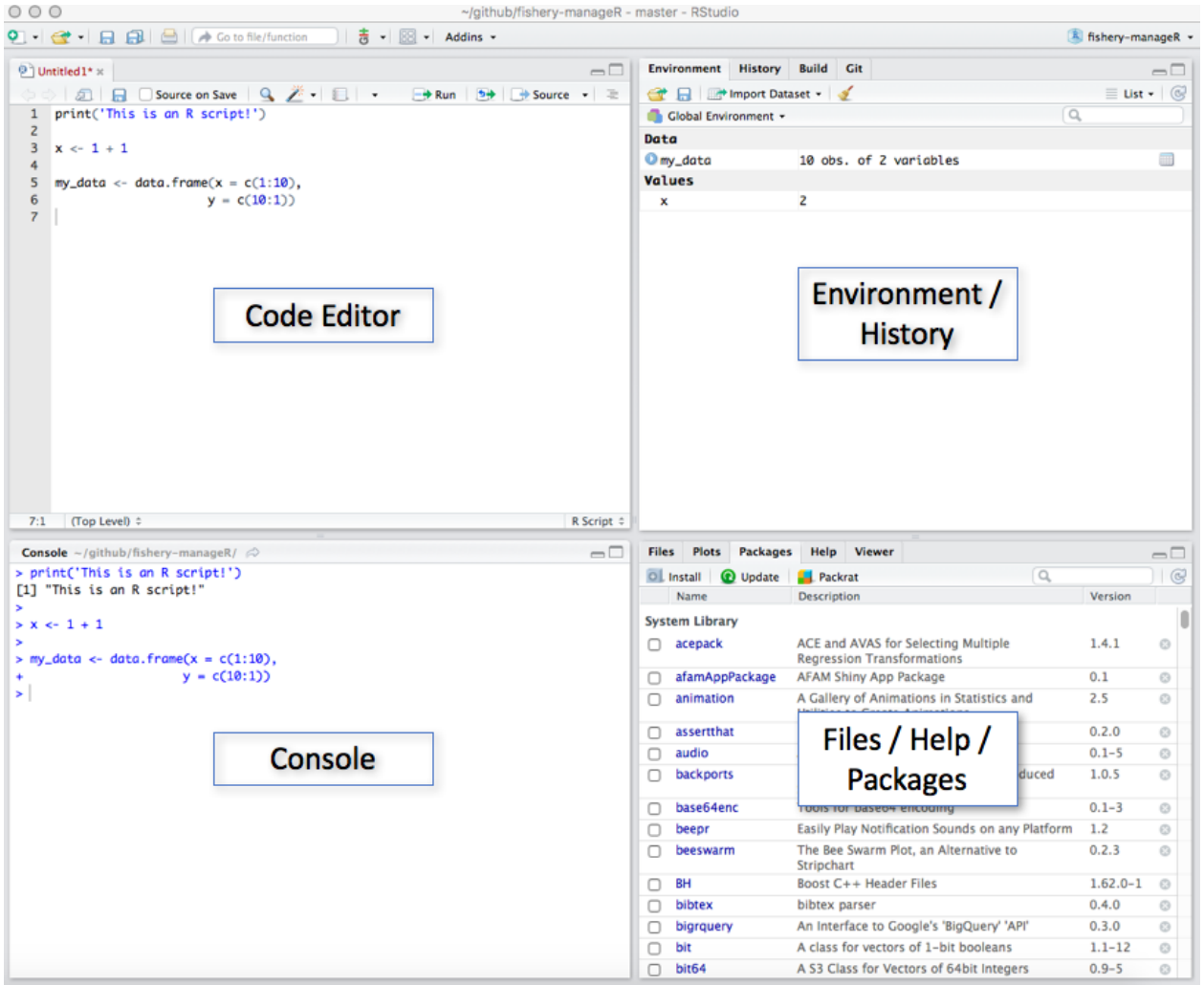
R es parte del sistema GNU y se distribuye bajo la licencia GNU GPL. Está disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux.

La instalación de R para Windows se puede realizar (https://cran.r-project.org/bin/windows/base/) y para Ubuntu (https://www.r-bloggers.com/download-and-install-r-in-ubuntu/).

Introducción: Rstudio

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

RStudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.



Introducción: Instalación de paquetes

Los paquetes son conjntos de funciones desarrollados por investigadores que aportan funciones adicional a la sesión de R.

En primer lugar hay que instalar el paquete (descargarlo en nuestro ordenador), y cada vez que se quiera usar alguna función de dicho paquete hay que cargarlo en la sesión abierta.
Para instalar un paquete:

- Desde la consola:

```
> install.packages("<Nombre del paquete>")
```

Ejemplo:

```
> install.packages("DAAG")
> # Más de un paquete by c("< Paquete 1 >","< Paquete 2 >")
> install.packages(c("DAAG","psych","gdata","foreign","Hmisc",
+                    "MASS","calibrate","corrplot","fields","xlsx",
+                    "RColorBrewer","ggplot2","lattice","visreg",
+                    "car","InformationValue","ROCR"))
```

- Desde el `Rstudio`: `Packages > Install`

Introducción: Cargar paquetes

Una vez hayamos instalado el paquete que contiene la función (funciones) que vamos a usar, debemos que cargar el paquete en la sesión de `R`.

Para ello usaremos la función `library` o `require`.

```
> library(DAAG) # o require(DAAG)
```

También podemos cargar el paquete clicando desde `Rsutdio` en el cuadrado que se encuentra delante de cada paquete previamente instalado.

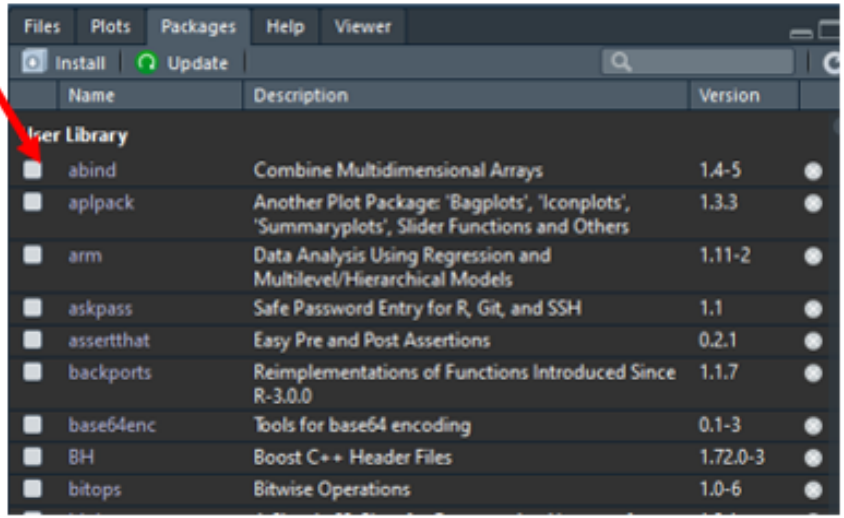


Table of Contents

1. Instalación de R y Rstudio
2. Importar datos
3. Análisis preliminar
   - Introducción
   - Medidas analíticas
   - Visualización
4. Regresión Lineal

Importar datos

Desde R podemos leer bases de datos guardados en diferentes formatos: `.txt`, `.csv`, `.xls`, `.xlsx`, SAS, Stata...
Cuando queramos importar una base de datos es muy importante establecer algunos parámetros que nos ayudarán a importar los datos de forma correcta:

- `sep=` Determina el símbolo de separación entre los datos: Espacios (por defecto), tabulación `"\tab"`, coma `","`, punto-coma `";"`.
- `header=TRUE` la primera fila de la base de datos hace referencia a los nombres de las columnas, `header=FALSE` (por defecto) de no ser así.
- `dec=` Símbolo decimal: Punto `"."` (por defecto), coma `","`.

Para leer datos desde `.txt` o `.csv`:

```
> mydata1 <- read.table("localizacion/datos.txt",sep=",",dec=",",header=FALSE)
> mydata2 <- read.csv("localizacion/datos.csv",sep="\tab",dec=".",header=TRUE)
```

Importar datos

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

Para poder importar otro tipo de formato necesitamos funciones que la sesión de R no contiene por defecto. Por lo tanto, debemos de cargar los paquetes que contienen dichas funciones:

Para leer ficheros con formato `.xls` o `.xlsx`, necesitamos cargar la librería `gdata`.

```
> library(gdata)
> mydata3 <- read.xls ("localizacion/datos.xls", sheet=1, header = TRUE)
```

Como es conocido, los ficheros de Excel pueden contener más de una hoja, por lo que para este tipo de importaciones debemos de añadir el parámetro `sheet` que indica la hoja de Excel que queremos importar en R.
Para importar datos en formatos Minitab, SPSS, SAS o Stata necesitamos otra librería, `foreing`, que contiene las siguientes funciones:

```
> library(foreign)
> mydata = read.mtp("localizacion/datos.mtp") # Minitab
> mydata = read.spss("localizacion/datos.spss", to.data.frame=TRUE) # SPSS
> mydata = read.dta("localizacion/datos.dta") # Stata
```

Table of Contents

1 Instalación de R y Rstudio

2 Importar datos

3 Análisis preliminar
- Introducción
- Medidas analíticas
- Visualización

4 Regresión Lineal

Análisis preliminar: Introducción

Universidad de Deusto
Deustuko Unibertsitatea

Base de datos Auto del paquete ISLR:

```
> library(ISLR)
> data(Auto) # Cargar los datos
```

Una vez tengamos cargada la base de datos, lo primero será saber que dimensión tiene (número de filas y columnas)

```
> dim(Auto)
```

```
[1] 392   9
```

y obtener una pequeña representación de las primeras filas

```
> head(Auto,n = 5) # tail() para los ultimos n datos
```

```
  mpg cylinders displacement horsepower weight acceleration
1  18         8          307        130   3504         12.0
2  15         8          350        165   3693         11.5
3  18         8          318        150   3436         11.0
4  16         8          304        150   3433         12.0
5  17         8          302        140   3449         10.5
  year origin                       name
1   70      1 chevrolet chevelle malibu
2   70      1         buick skylark 320
3   70      1        plymouth satellite
4   70      1             amc rebel sst
5   70      1               ford torino
```

Para seleccionar datos usaremos:

- `Auto[i,j]` para seleccionar el elemento de la fila i y columna j de la base de datos `Auto`.

```
> Auto[1,1]

[1] 18
```

- `Auto[i,]` para seleccionar la fila i.

```
> Auto[1,]

  mpg cylinders displacement horsepower weight acceleration
1  18         8          307        130   3504           12
  year origin                       name
1   70      1 chevrolet chevelle malibu
```

- `Auto[,j]` para seleccionar la columna j.

```
> head(Auto[,1],n=25)

 [1] 18 15 18 16 17 15 14 14 14 15 15 14 15 14 24 22 18 21
[19] 27 26 25 24 25 26 21
```

Análisis preliminar: Introducción

Universidad de Deusto
Deustuko Unibertsitatea

Existe una manera más cómoda de seleccionar columnas en bases de datos, que consiste en usar el símbolo $ de la siguiente manera: bbdd$<nombre columna>.

Para saber los nombres de las variables (columnas) usaremos:

```
> names(Auto)

[1] "mpg"          "cylinders"    "displacement"
[4] "horsepower"   "weight"       "acceleration"
[7] "year"         "origin"       "name"
```

y para seleccionar la columna llamada `horsepower`:

```
> head(Auto$horsepower,n=25)

 [1] 130 165 150 150 140 198 220 215 225 190 170 160 150 225
[15]  95  95  97  85  88  46  87  90  95 113  90
```

Para seleccionar más de una fila o columna usaremos la función `c()` que sirve para crear vectores, es decir,

```
> c(1,4,9,16,25)
```

```
[1]  1  4  9 16 25
```

Por lo tanto, para seleccionar las filas 1, 4 y 9:

```
> Auto[c(1,4,9),]
```

```
  mpg cylinders displacement horsepower weight acceleration
1  18         8          307        130   3504           12
4  16         8          304        150   3433           12
9  14         8          455        225   4425           10
  year origin                       name
1   70      1 chevrolet chevelle malibu
4   70      1            amc rebel sst
9   70      1        pontiac catalina
```

También podemos seleccionar un número de columnas del mismo modo.

Pero, además, también podemos eliminar filas o columnos de forma muy similar usando la función `-`. Si queremos borrar las filas 1 y 2, y guardar la nueva base de datos en el objeto `Auto.v2`,

```
> Auto.v2 <- Auto[-c(1,2),]
```

Análisis preliminar: Introducción

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

Procedemos a analizar si existe algún dato faltante. Para ello usaremos dos funciones:

1. `which()`, devuelve la posisición de los valores que satisfacen una regla:

   ```
   > which(Auto$cylinders==3)
   ```

   ```
   [1]  71 111 242 332
   ```

   y si quisieramos seleccionar dichos valores,

   ```
   > Auto$horsepower[which(Auto$horsepower>200)]
   ```

   ```
    [1] 220 215 225 225 215 210 208 215 225 230
   ```

2. `is.na()`, devuelve `TRUE` si el valor es faltante y `FALSE` en caso contrario.

   ```
   > head(is.na(Auto$origin),n=18)
   ```

   ```
    [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
   [10] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
   ```

3. Por lo tanto, combinando las dos funciones podemos saber la posición de los datos faltantes:

   ```
   > which(is.na(Auto$name)==TRUE)
   ```

   ```
   integer(0)
   ```

   En este caso, vemos que no existen datos faltantes en la columna `name`. De haber habido, podríamos quitar las filas que lo contienen con la función – mendionada con anterioridad.

Medidas analíticas: Introducción

Antes de calcular las medidas analíticas de cada variable (columna) debemos de saber con qué tipo de variable traramos.

En `R` hay tres grandes tipos de clases de objetos:

- `numeric`: objeto que contiene datos numéricos (enteros, reales...)
- `factor`: objeto que contiene datos categóricos estructurados.
- `character`: objeto que contiene datos categóricos desestructurados.

  Es importante saber la clase de cada variable, porque muchas de las funciones solo tienen sentido con ciertas clases de variables.

  Para saber la clase de una variable usaremos `class()`:

```
> class(Auto$mpg)

[1] "numeric"

> class(Auto$name)

[1] "factor"
```

Medidas analíticas

Podemos calcular medidas analíticas de centralización, dispersión, forma etc siempre que tengamos variables numéricas:

```
> max(Auto$horsepower) # máximo

[1] 230

> min(Auto$horsepower) # mínimo

[1] 46

> mean(Auto$horsepower) # media

[1] 104.4694

> median(Auto$horsepower) # mediana

[1] 93.5

> quantile(Auto$horsepower,probs = 0.25)

25%
 75
```

```
> var(Auto$horsepower) # varianza

[1] 1481.569

> sd(Auto$horsepower) # desv. est.

[1] 38.49116

> range(Auto$horsepower) # rango

[1]  46 230

> sd(Auto$horsepower)/mean(Auto$horsepower)

[1] 0.3684444
```

Medidas analíticas: Correlación

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

Antes de realizar cualquier modelo estadístico es muy importante conocer la dependencia que existe en los datos, es decir, la correlación que guardan las diferentes variables de la base de datos.

Para calcular la correlación entre dos variables usaremos la función `cor()`

```
> cor(Auto$cylinders,Auto$horsepower)
```

```
[1] 0.8429834
```

Pero podemos también calcular la matriz de correlaciones,

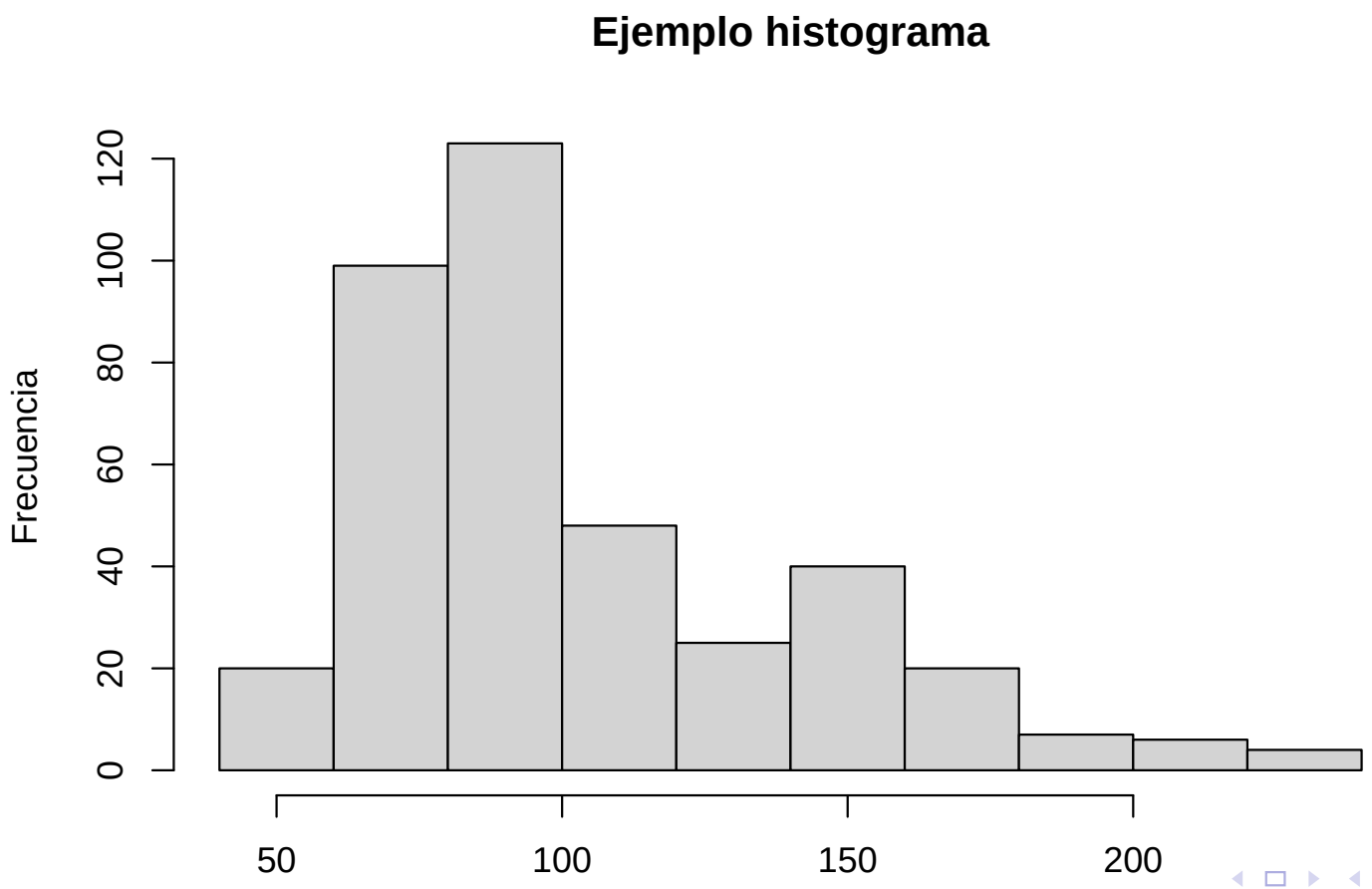
```
> cor(Auto[,-9]) # no tiene sentido correlaciones para variables categoricas
```

```
                    mpg  cylinders displacement horsepower
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834
displacement -0.8051269  0.9508233    1.0000000  0.8972570
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000
weight       -0.8322442  0.8975273    0.9329944  0.8645377
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955
year          0.5805410 -0.3456474   -0.3698552 -0.4163615
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715
                 weight acceleration       year     origin
mpg          -0.8322442    0.4233285  0.5805410  0.5652088
cylinders     0.8975273   -0.5046834 -0.3456474 -0.5689316
displacement  0.9329944   -0.5438005 -0.3698552 -0.6145351
horsepower    0.8645377   -0.6891955 -0.4163615 -0.4551715
weight        1.0000000   -0.4168392 -0.3091199 -0.5850054
acceleration -0.4168392    1.0000000  0.2903161  0.2127458
year         -0.3091199    0.2903161  1.0000000  0.1815277
origin       -0.5850054    0.2127458  0.1815277  1.0000000
```

Visualización: Histograma

Histograma: es una representación gráfica de una variable en forma de barras, donde la altura de cada barra es la frecuencia de los valores representados. Sirven para obtener una "primera vista"general de la distribución de una variable aleatoria continua.

```
> hist(Auto$horsepower,main="Ejemplo histograma",xlab="Potencia caballos",
+      ylab="Frecuencia",col="lightgray")
```

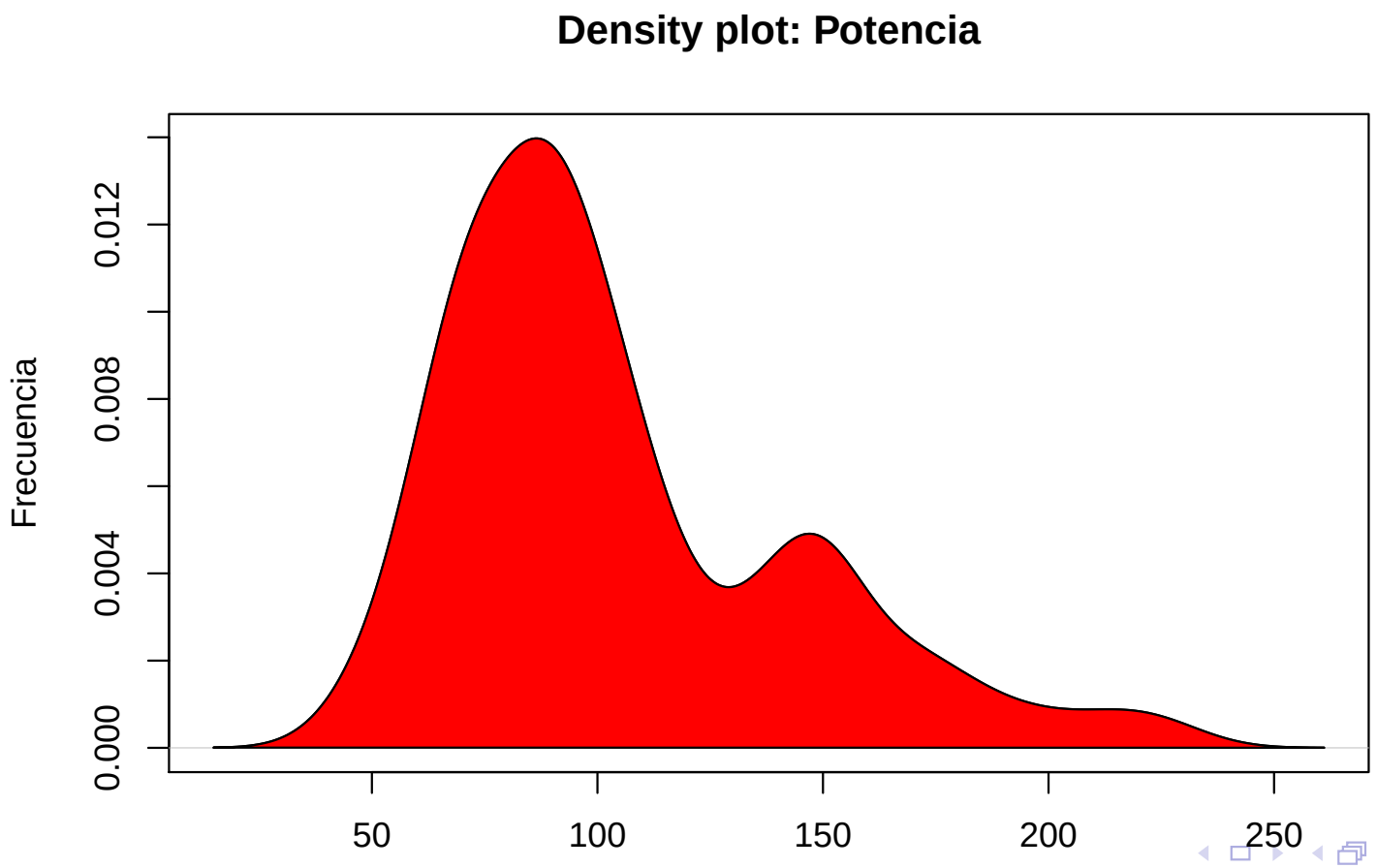


Visualización: Densidad

Universidad de Deusto
Deustuko Unibertsitatea

Podemos dibujar también la densidad estimada de cada una de las variables que sirve para, al igual que el histograma, determinar la forma de la variable.

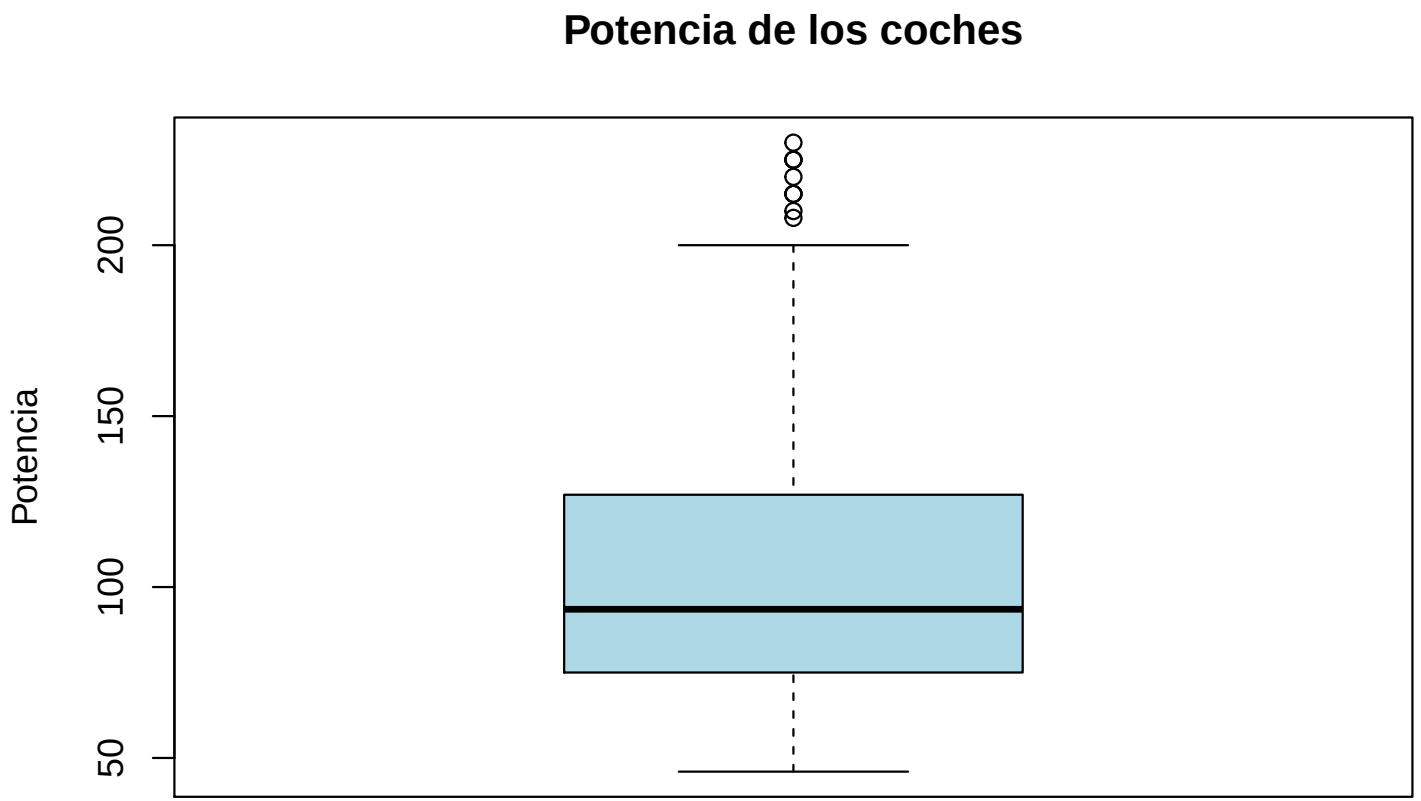
```
> plot(density(Auto$horsepower), main="Density plot: Potencia", ylab="Frecuencia"
+      ,xlab="")
> polygon(density(Auto$horsepower), col="red")
```



Visualización: Diagrama de cajas

Diagrama de cajas (boxplot): es una presentación visual que describe varias características importantes, al mismo tiempo, tales como la dispersión y simetría. Para su realización se representan los tres cuartiles y los valores mínimo y máximo de los datos, sobre un rectángulo, alineado horizontal o verticalmente.

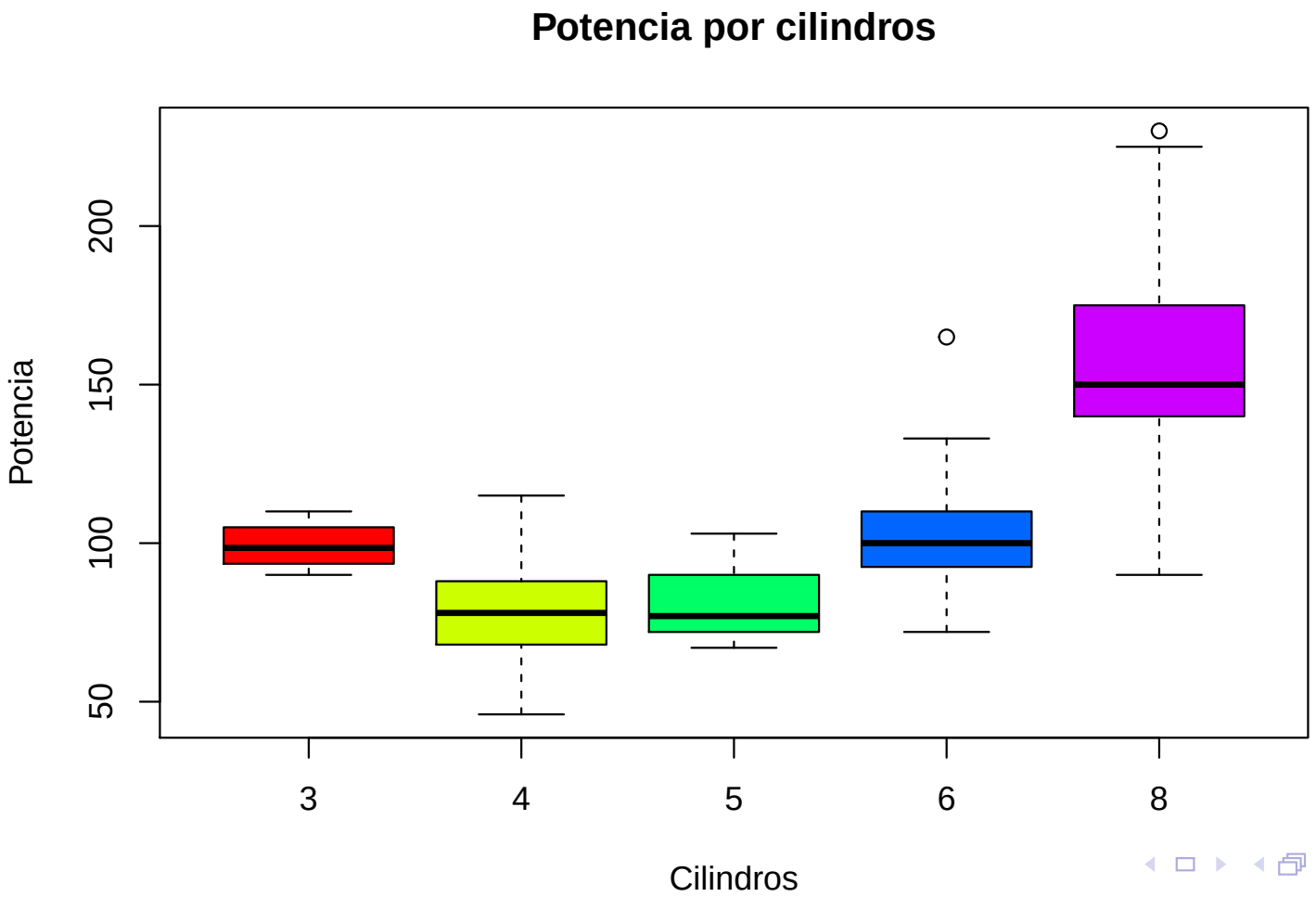
```
> boxplot(Auto$horsepower,xlab="",ylab="Potencia",
+         main="Potencia de los coches",col="lightblue")
```



Visualización: Diagrama de cajas

Diagrama de cajas por una variable categórica.

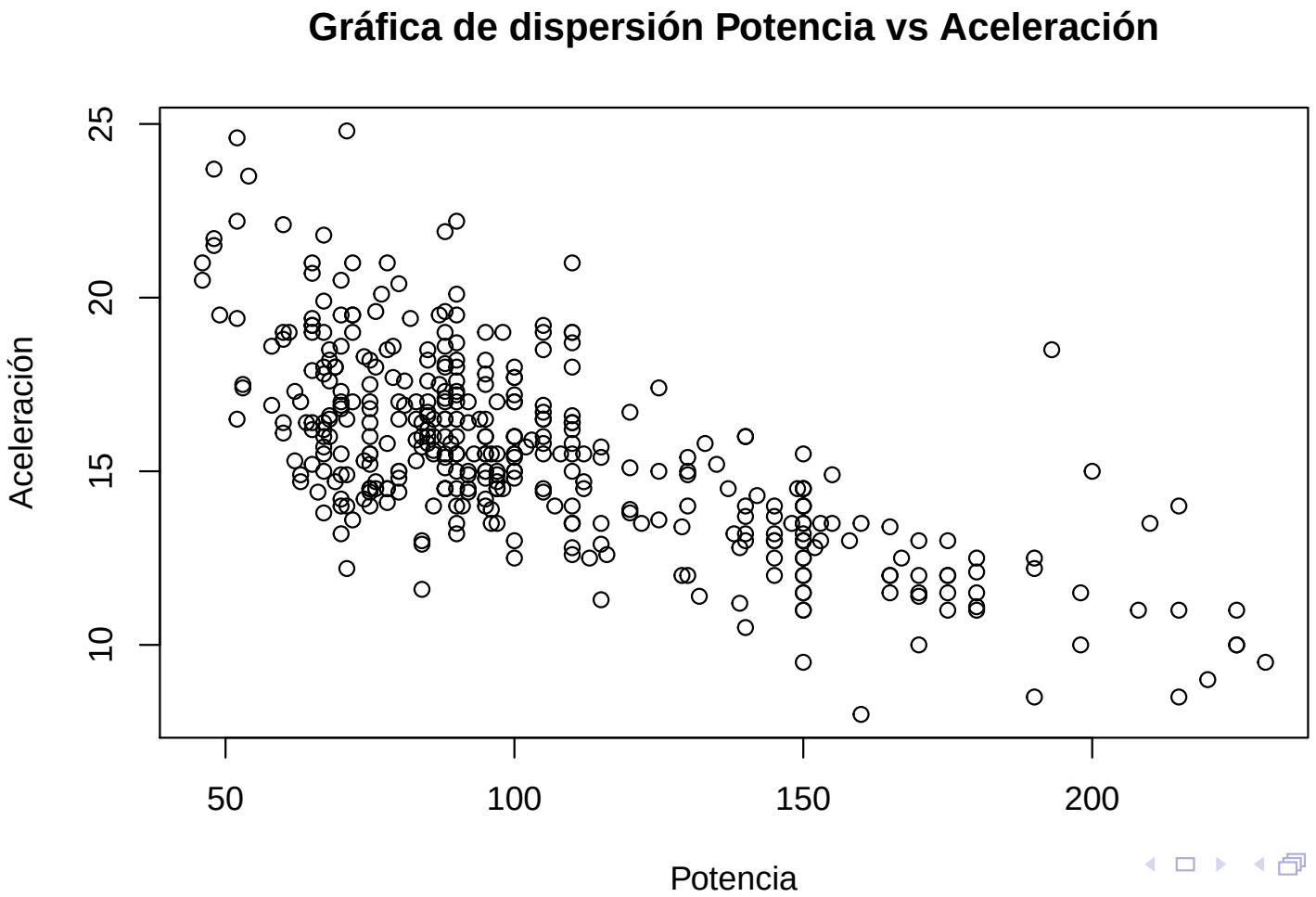
```
> library(RColorBrewer)
> boxplot(Auto$horsepower~Auto$cylinders, main="Potencia por cilindros",
+         xlab="Cilindros",ylab="Potencia",col=rainbow(5))
```



Visualización: Scatterplot

Gráfica de dispersión (Scatterplot): representa las coordenadas de los individuos (filas) en dos o más variables. Sirve para visualizar la posible relación que pueda existir entre las variables.

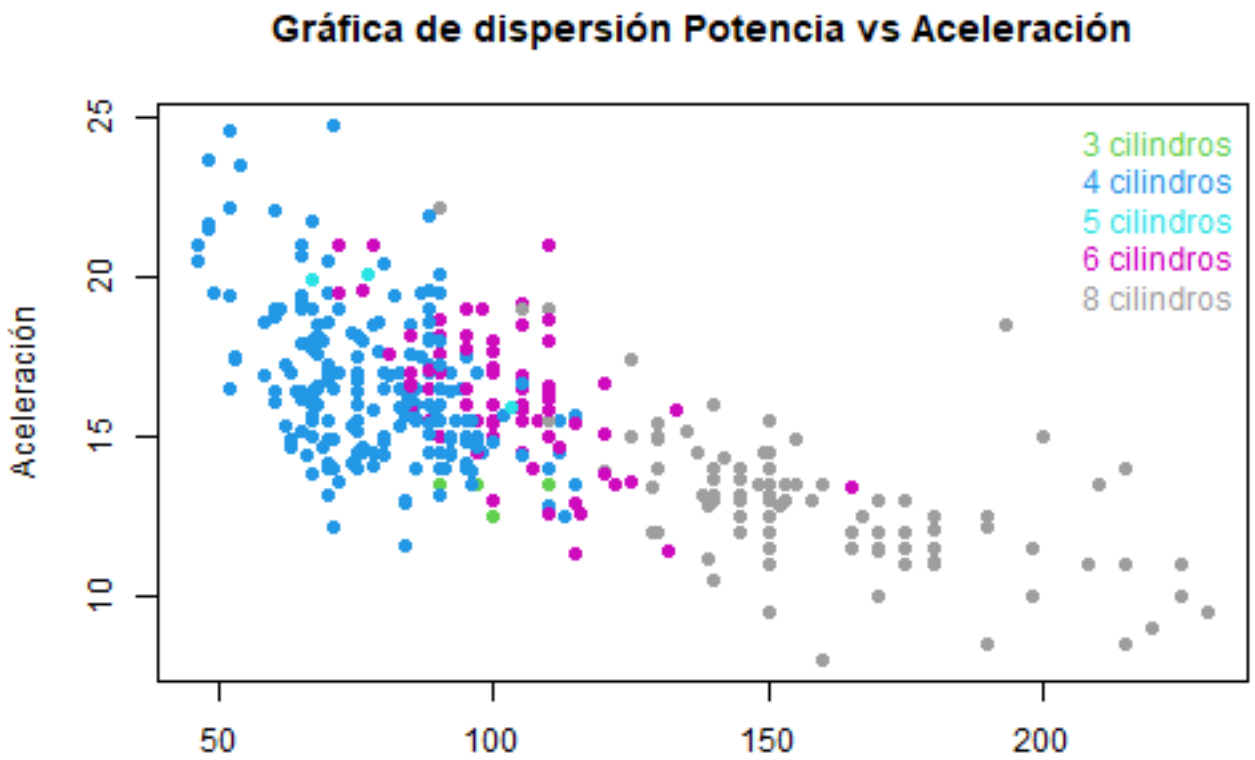
```
> plot(Auto$horsepower,Auto$acceleration,xlab="Potencia",ylab="Aceleración",
+      main="Gráfica de dispersión Potencia vs Aceleración")
```



Visualización: Scatterplot

Gráfica de dispersión (Scatterplot): representa las coordenadas de los individuos (filas) en dos o más variables. Sirve para visualizar la posible relación que pueda existir entre las variables.

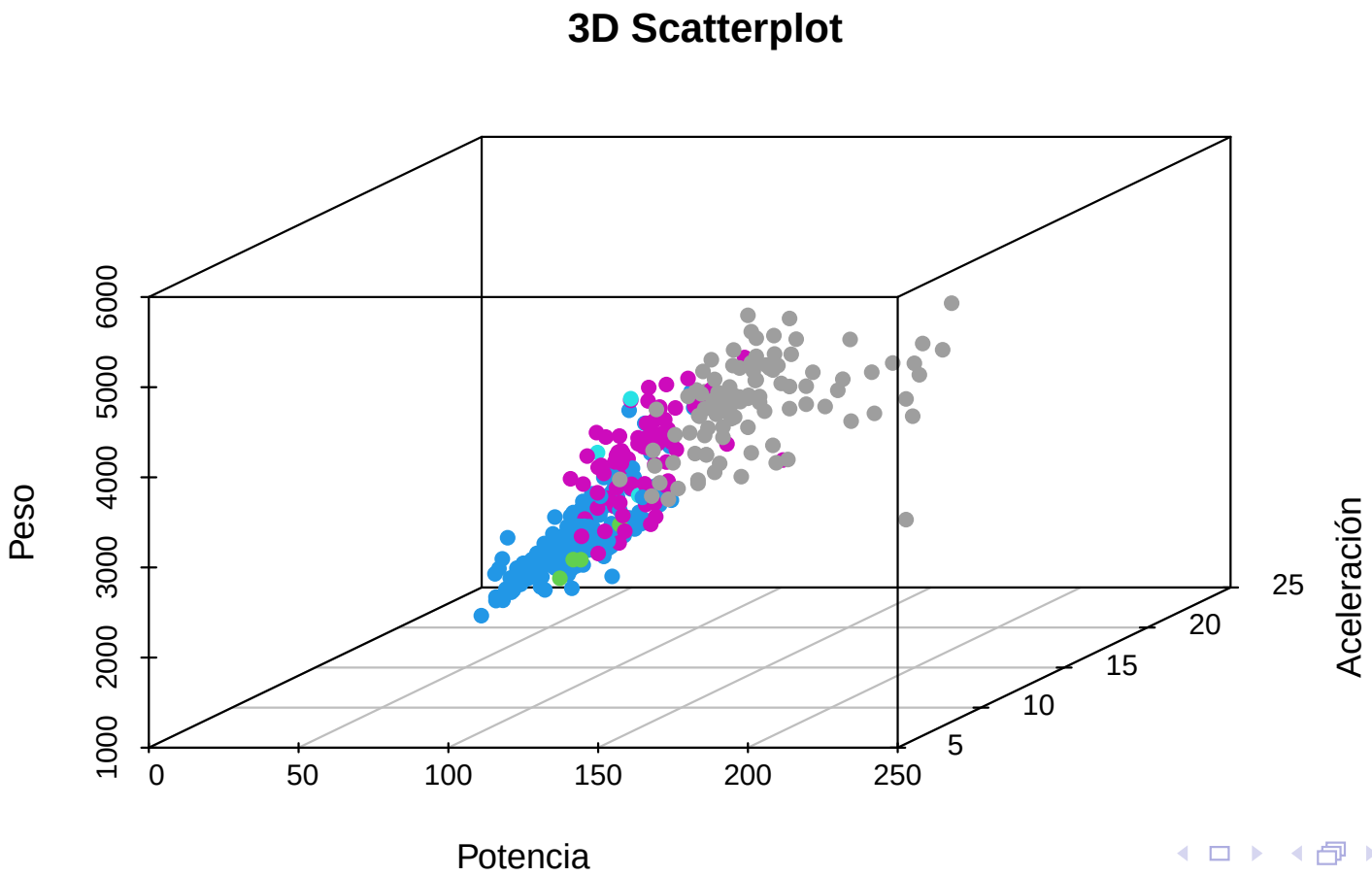
```
> plot(Auto$horsepower,Auto$acceleration,xlab="Potencia",ylab="Aceleración",
+      main="Gráfica de dispersión Potencia vs Aceleración",col=Auto$cylinders,
+      pch=16)
> legend("topright",legend = paste0(sort(unique(Auto$cylinders))," cilindros"),
+        text.col = sort(unique(Auto$cylinders)),bty="n")
```



Visualización: Scatterplot

También se pueden representar gráficas de dispersión con tres variables, pero en este caso no se aprecian tan bien las relaciones.

```
> library(scatterplot3d)
> scatterplot3d(Auto$horsepower,Auto$acceleration,Auto$weight, main="3D Scatterplot"
+                 xlab="Potencia",ylab="Aceleración",zlab="Peso",pch=16,color = Auto$c
```



Visualización: Correlaciones

Antes hemos presentado la forma de calcular la matriz de correlaciones entre las variables en una base de datos. Sin embargo, cuando el número de variables es muy grande, resulta complicado obtener resultados.
Podemos mostrar las correlaciones visualmente con la función `corrplot`.

```
> library(corrplot)
> corrplot(cor(Auto[,-9]))
```

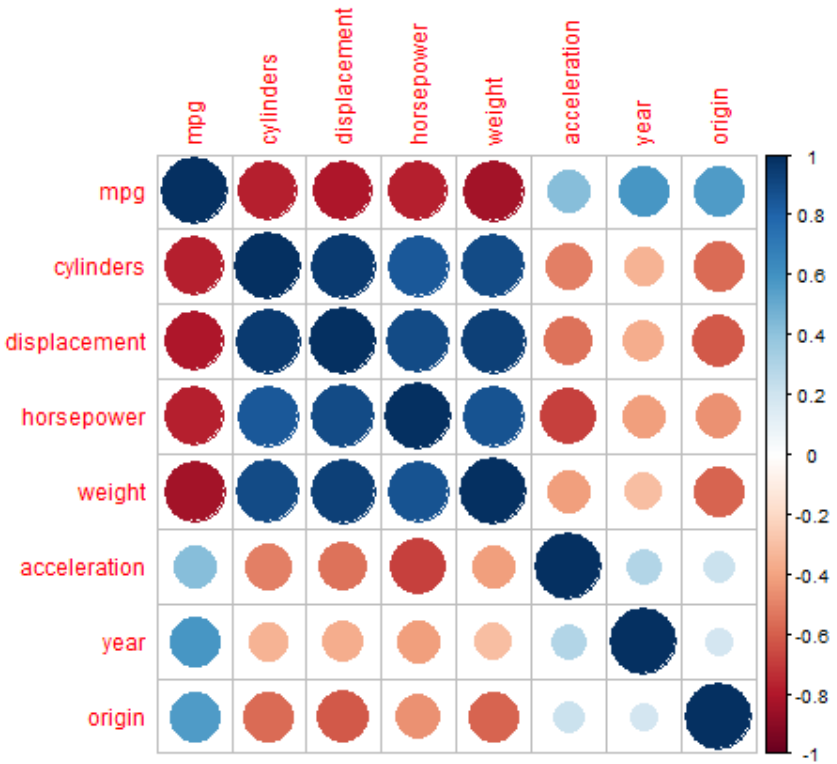


Table of Contents

1. Instalación de R y Rstudio
2. Importar datos
3. Análisis preliminar
   - Introducción
   - Medidas analíticas
   - Visualización
4. Regresión Lineal

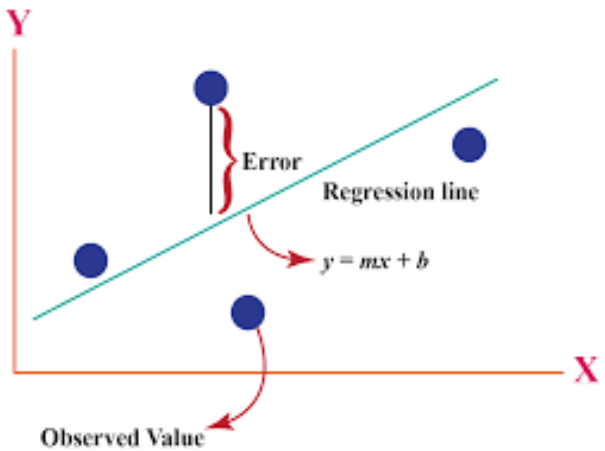
- **Objetivo:** Describir una variable independiente Y mediante variables dependientes $X_1, \ldots, X_k$.
- La relación es lineal,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- Los errores, ϵ, son independientes y siguen una distribución normal con media cero,

$$\epsilon \sim \mathcal{N}(0, \sigma I_k)$$

- β_i son los coefficientes que determinan la relación y que deberemos de estimar;
  - *estimar:* obtener el valor que mejor se ajusta al modelo basado en los datos que tenemos.
    - ⋆ Mínimos cuadrados (minimiza los errores de la estimación).



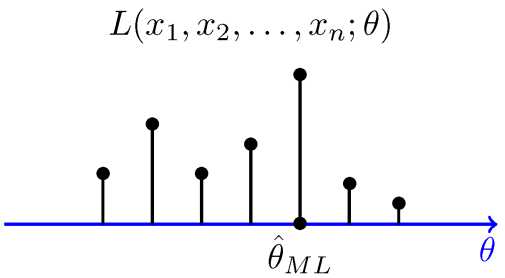
- **Objetivo:** Describir una variable independiente Y mediante variables dependientes $X_1, \ldots, X_k$.
- La relación es lineal,

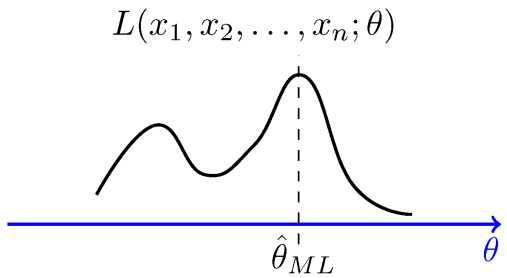
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- Los errores, ϵ, son independientes y siguen una distribución normal con media cero,

$$\epsilon \sim \mathcal{N}(0, \sigma I_k)$$

- β_i son los coefficientes que determinan la relación y que deberemos de estimar;
  - *estimar:* obtener el valor que mejor se ajusta al modelo basado en los datos que tenemos.
    - ⋆ Máxima verosimilitud (calcula los valores más probables de los parámetros basado en los datos observados).





- **Objetivo:** Describir una variable independiente Y mediante variables dependientes $X_1, \ldots, X_k$.
- La relación es lineal,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- Los errores, ϵ, son independientes y siguen una distribución normal con media cero,

$$\epsilon \sim \mathcal{N}(0, \sigma I_k)$$

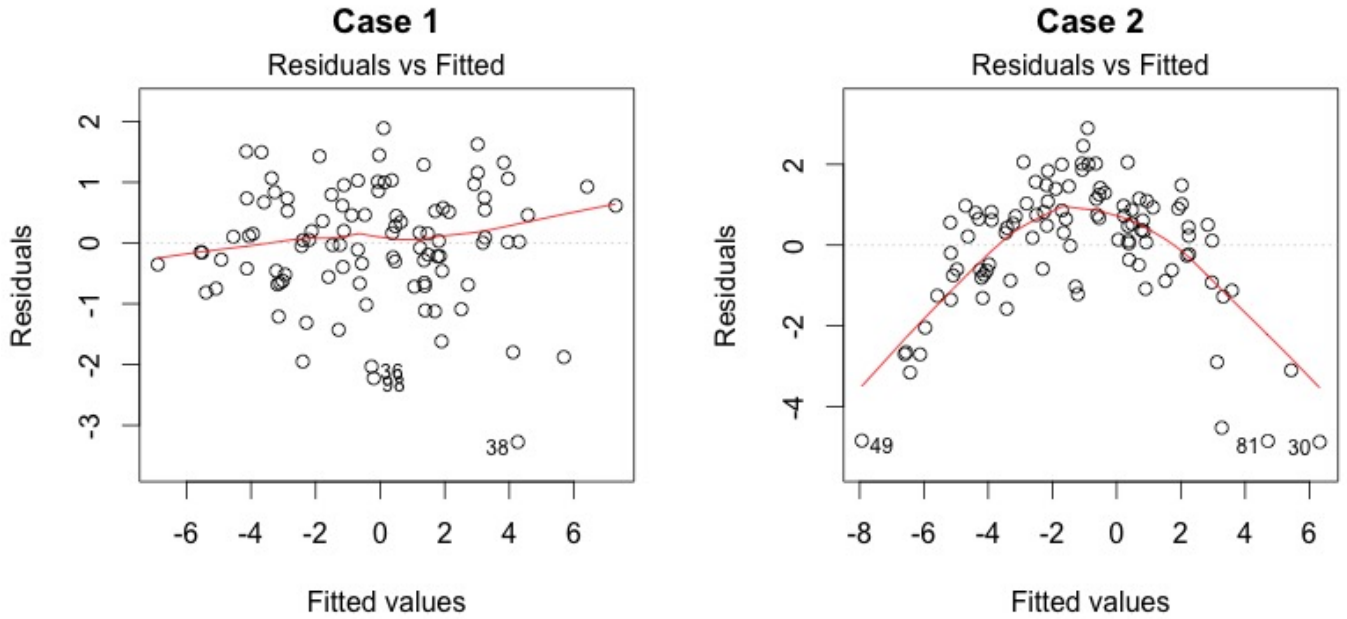
- β_i son los coefficientes que determinan la relación y que deberemos de estimar;
  - *estimar:* obtener el valor que mejor se ajusta al modelo basado en los datos que tenemos.
    - Cuando se realiza una estimación se realiza también un contraste de hipótesis donde se determina la significación del parámetro estimado,

$$\begin{cases} H_0 : & \beta_i = 0 \\ H_1 : & \beta_i \neq 0 \end{cases}$$

Una vez que el modelo es ajustado y los coeficientes estimados, hay que verificar que se cumplen las hipótesis planteadas.

- **la relación es lineal.**

  Residuals vs Fitted. Se usa para chequear la relación lineal. Una linea horizontal, sin ningún tipo de patrón, es un indicador de la existencia de una relación lineal.

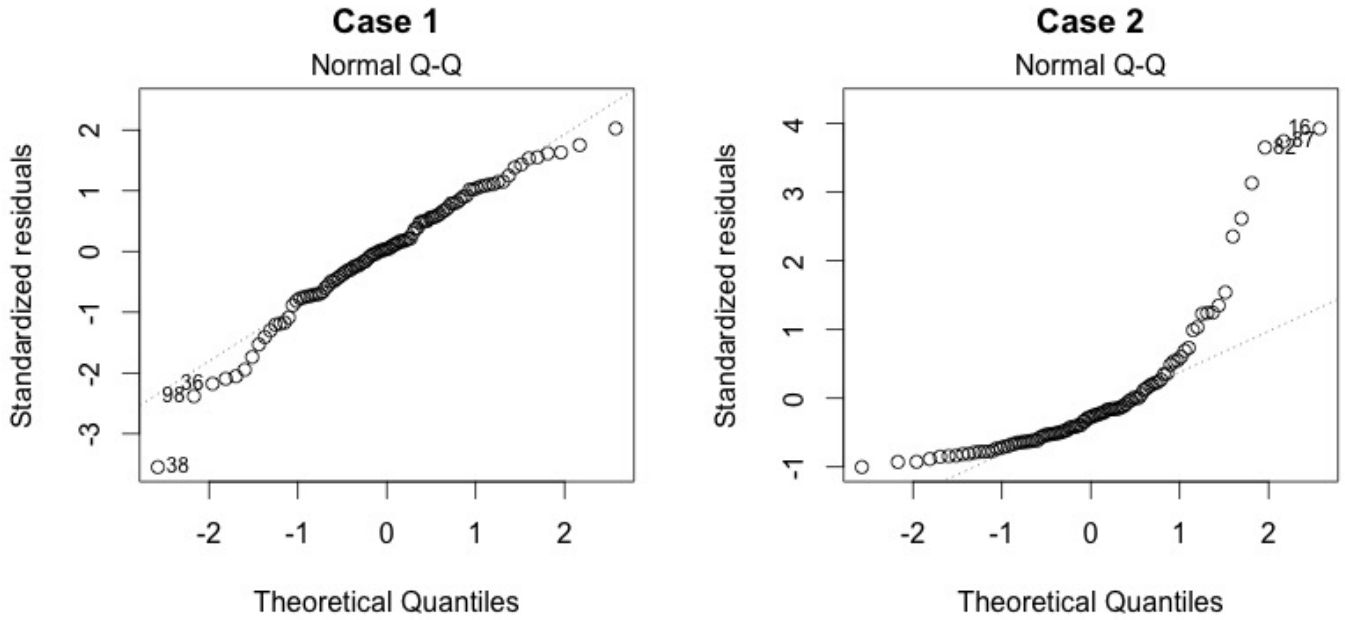


- **los residuos siguen una distribución normal.**
- **la varianza de los residuos es constante (homocedasticidad).**

Una vez que el modelo es ajustado y los coeficientes estimados, hay que verificar que se cumplen las hipótesis planteadas.

- **la relación es lineal.**
- **los residuos siguen una distribución normal.**

  Normal Q-Q. Se usa para examinar si los residuos siguen una distribución normal. Que los residuos estén cerca de la linea discontinua es un muy buen indicador.



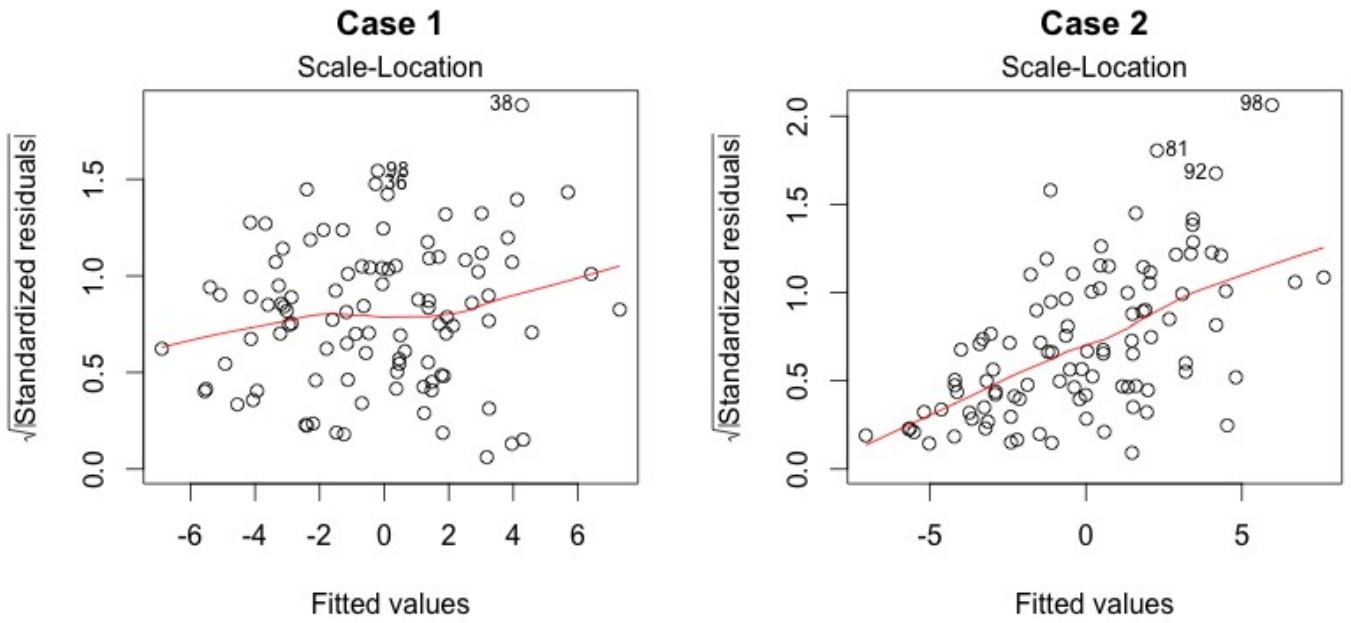
- **la varianza de los residuos es constante (homocedasticidad).**

Regresión lineal: Introducción

Una vez que el modelo es ajustado y los coeficientes estimados, hay que verificar que se cumplen las hipótesis planteadas.

- **la relación es lineal.**
- **los residuos siguen una distribución normal.**
- **la varianza de los residuos es constante (homocedasticidad).**

  Scale-Location. Usado para chequear la homogeneidad de la varianza (homocedastiscidad). El ajuste de una linea horizontal con puntos distribuidos uniformente es un buen indicador.



Regresión lineal: Aplicación

Ajustemos el siguiente modelo:

Modelo

$$mpg = \beta_0 + \beta_1 cylinders + \beta_2 displacement + \beta_3 horsepower + \beta_4 weight + \beta_5 acceleration + \beta_6 origin + \epsilon$$

y lo guardamos en un objeto que llamaremos modelo.

```
> Auto$origin <- as.factor(Auto$origin)
> modelo <- lm(mpg~cylinders+displacement+horsepower+weight+acceleration+origin,
+              data=Auto)
```

Podemos mostrar el resultado de las estimaciones pidiendo a R que nos muestre el objeto `modelo`.

```
> modelo

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration + origin, data = Auto)

Coefficients:
 (Intercept)     cylinders  displacement    horsepower
   44.768791     -0.566188      0.011427     -0.061334
      weight  acceleration       origin2       origin3
   -0.004812     -0.031984      1.125545      2.932540
```

Regresión lineal: Significación

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

```
> summary(modelo)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration + origin, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.6303  -2.8009  -0.2871   2.0945  14.8931

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  44.7687911  2.6398457  16.959  < 2e-16 ***
cylinders    -0.5661876  0.4042069  -1.401 0.162100
displacement  0.0114270  0.0095737   1.194 0.233376
horsepower   -0.0613339  0.0168679  -3.636 0.000314 ***
weight       -0.0048119  0.0008089  -5.948  6.1e-09 ***
acceleration -0.0319841  0.1232529  -0.259 0.795389
origin2       1.1255451  0.7015566   1.604 0.109458
origin3       2.9325397  0.6955675   4.216  3.1e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.161 on 384 degrees of freedom
Multiple R-squared:  0.7209,        Adjusted R-squared:  0.7158
```

Regresión lineal: Significación

Deusto
Universidad de Deusto
Deustuko Unibertsitatea

Modelo

$$mpg = \beta_0 + \beta_1 horsepower + \beta_2 weoght + \beta_{32} origin2 + \beta_{33} origin3 + \epsilon$$

```
> modelo.v2 <- lm(mpg~horsepower+weight+origin,
+               data=Auto)
> summary(modelo.v2)

Call:
lm(formula = mpg ~ horsepower + weight + origin, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.3772  -2.9241  -0.3991   2.2345  14.8164

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.7399583  1.1011112  38.815  < 2e-16 ***
horsepower  -0.0535442  0.0109835  -4.875 1.59e-06 ***
weight      -0.0048427  0.0005455  -8.878  < 2e-16 ***
origin2      0.9611172  0.6402552   1.501    0.134
origin3      2.7422385  0.6517055   4.208 3.21e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La interpretación de cada coeficiente estimado depende de la naturaleza de la covariable la cual acompaña:

- Si la covariable es **continua**, por ejemplo, el peso en nuestro modelo, el coeficiente β_2 muestra el aumento medio de millas por galón, `mpg`, por unidad de peso que aumente el vehículo.

$$\begin{aligned} mpg &= \beta_0 + \beta_1 horsepower + \beta_2(weight + 1) + \beta_{32} origin2 + \beta_{33} origin3 \\ &= (\beta_0 + \beta_1 horsepower + \beta_2 weight + \beta_{32} origin2 + \beta_{33} origin3) + \beta_2 \end{aligned}$$

  Por lo tanto, el aumento de 1kg en el peso del vehículo conlleva un aumento medio de $-0{,}0048$ millas por galón, es decir, con un galón se realizan 0,0048 millas menos en media.

La interpretación de cada coeficiente estimado depende de la naturaleza de la covariable la cual acompaña:

- Si la covariable es **categorica**, por ejemplo, el origen, no tiene ningún sentido hace rla interpretación previa, por el aumento de una unidad en el origen... ?¿?¿?.
  - Dummy variables: Tomando una categoría como referencia, se crean variables binarias a partir de una variable categorica.

Variable original	Variable codificada	Dummy origin.2	Dummy origin.3
América	1	0	0
Europa	2	1	0
Japón	3	0	1

$$mpg = \beta_0 + \beta_1 horsepower + \beta_2 weight + \beta_{32} origin2 + \beta_{33} origin3$$

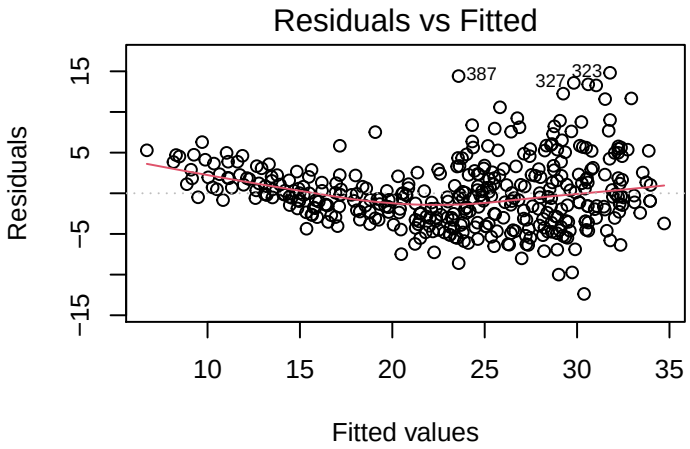
  - ⋆ Si `origin=America` $\Longrightarrow mpg = \beta_0 + \beta_1 horsepower + \beta_2 weight$.
  - ⋆ Si `origin=Europa` $\Longrightarrow mpg = \beta_0 + \beta_1 horsepower + \beta_2 weight + \beta_{32}$.
  - ⋆ Si `origin=Japón` $\Longrightarrow mpg = \beta_0 + \beta_1 horsepower + \beta_2 weight + \beta_{33}$.

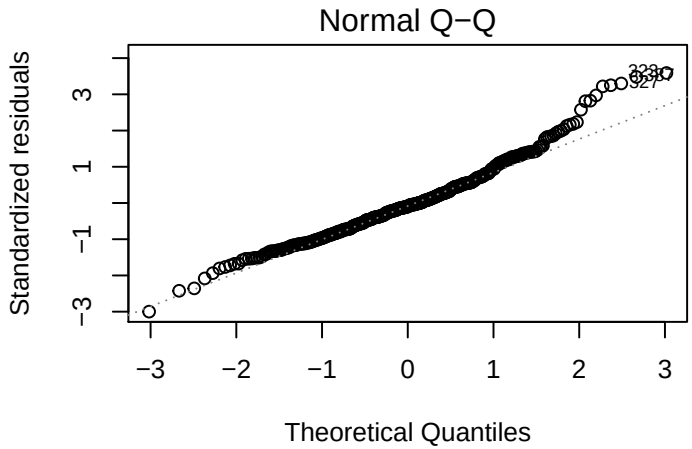
  Por lo tanto, $\beta_{32} = 0{,}961$ es el incremento medio de millas en un vehículo europeo comparado con uno americano; y $\beta_{33} = 2{,}742$ el aumento de un coche japonés comparado con un coche americano.

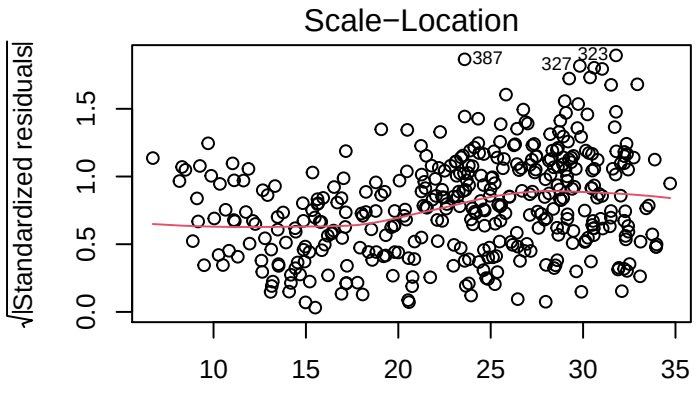
Regresión lineal: Diagnosis del modelo

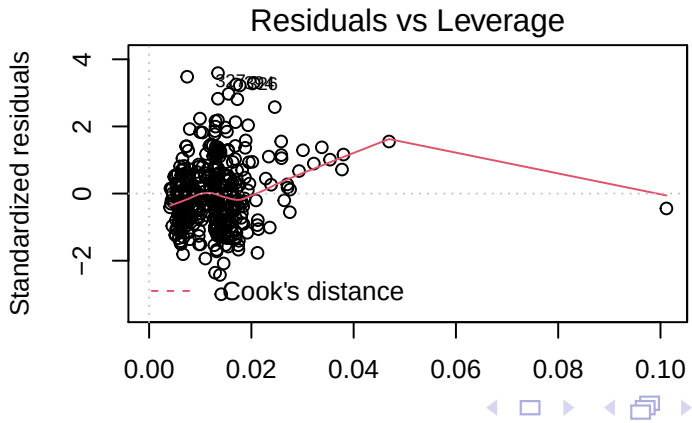
Para completar el diagnosis del modelo, usamos la función `plot()`.

```
> par(mfrow=c(2,2))
> plot(modelo.v2)
```









Eskerrik asko

Muchas gracias