

# Visualización y limpieza de datos con R

FSC

January 28, 2019

## Contents

R Basic plots: paquete <i>graphics()</i> . . . . .	1
Scatterplots (nubes de puntos) . . . . .	1
Histograms . . . . .	6
Boxplot . . . . .	9
Ejercicio #1: dataset <i>babies</i> . . . . .	11
Sumarizando datos y detectando outliers . . . . .	15
Medidas de resumen paramétricas: Media y desviación standard . . . . .	15
Medidas de resumen no paramétricas: Mediana, IQR . . . . .	15
Sumarizando data con <i>dplyr()</i> . . . . .	19
<i>summarize()</i> . . . . .	19
<i>dot</i> . . . . .	20
<i>group_by()</i> . . . . .	21
Ordenar data.frames: <i>arrange()</i> <i>top_n()</i> . . . . .	22
Ejercicio #2: Distribución de las alturas de los estudiantes . . . . .	23

## R Basic plots: paquete *graphics()*

Una de las grandes fortalezas de R consiste en la facilidad con la que podemos representar datos de diferentes formas y con formato de alta calidad. Visualizar los datos es esencial para entenderlos y proponer modelos.

Para seguir esta sesión necesitarás tener instaladas las siguientes librerías:

```
library(dslabs)
library(ggplot2)
library(dplyr)
library(tidyverse)
library(readr)
```

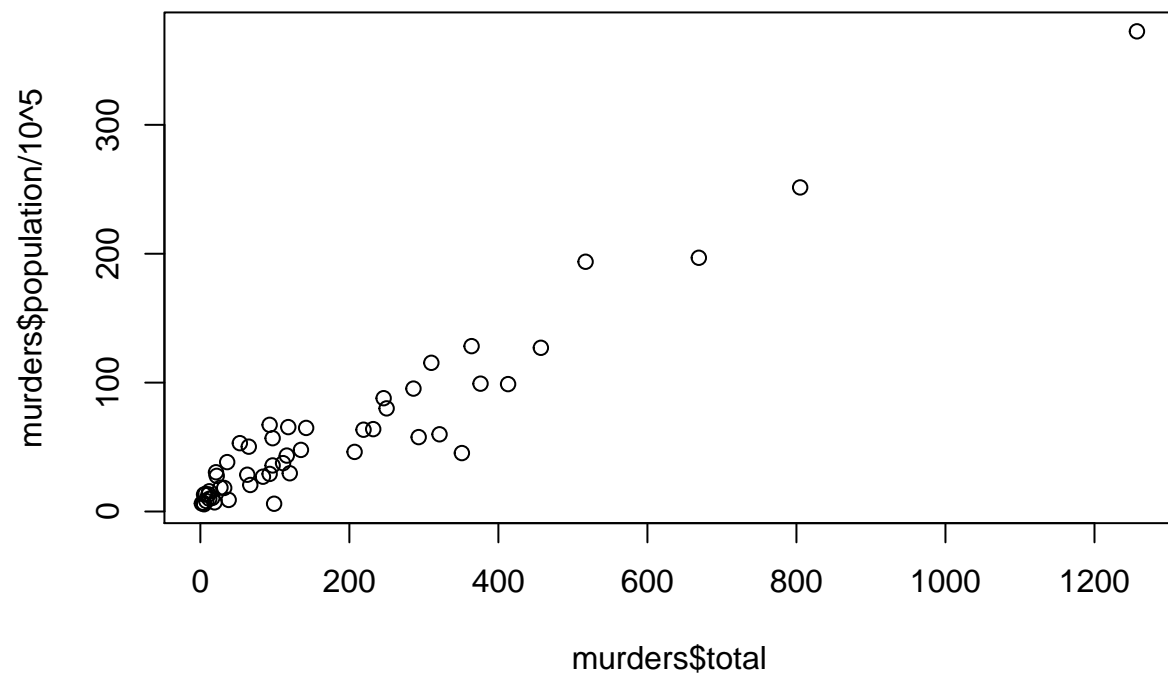
## Scatterplots (nubes de puntos)

El plot más básico en el que podemos pensar es una nube de puntos. Solemos utilizarlo con frecuencia si queremos entender la relación que existe entre dos variables. Por ejemplo, si queremos ver la relación entre el número de asesinatos en un estado y su población (por 100.000 habitantes) utilizamos la función *plot()*

```
data(murders)
?plot
```

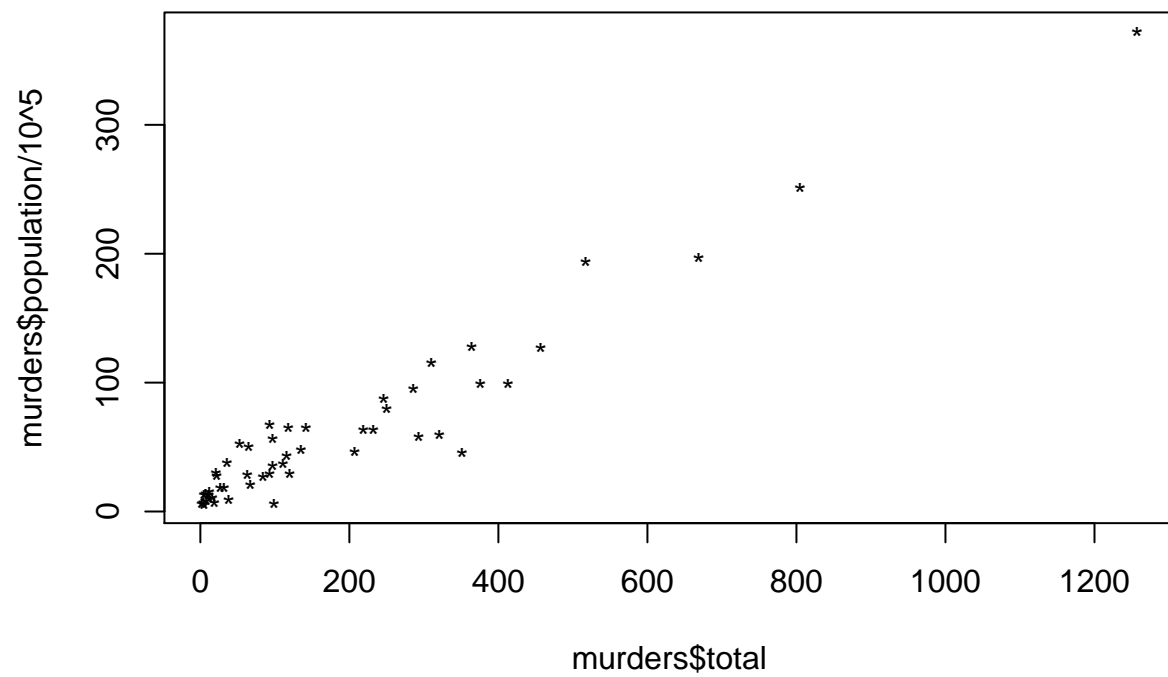
```
## starting httpd help server ... done
```

```
plot(murders$total, murders$population/10^5)
```



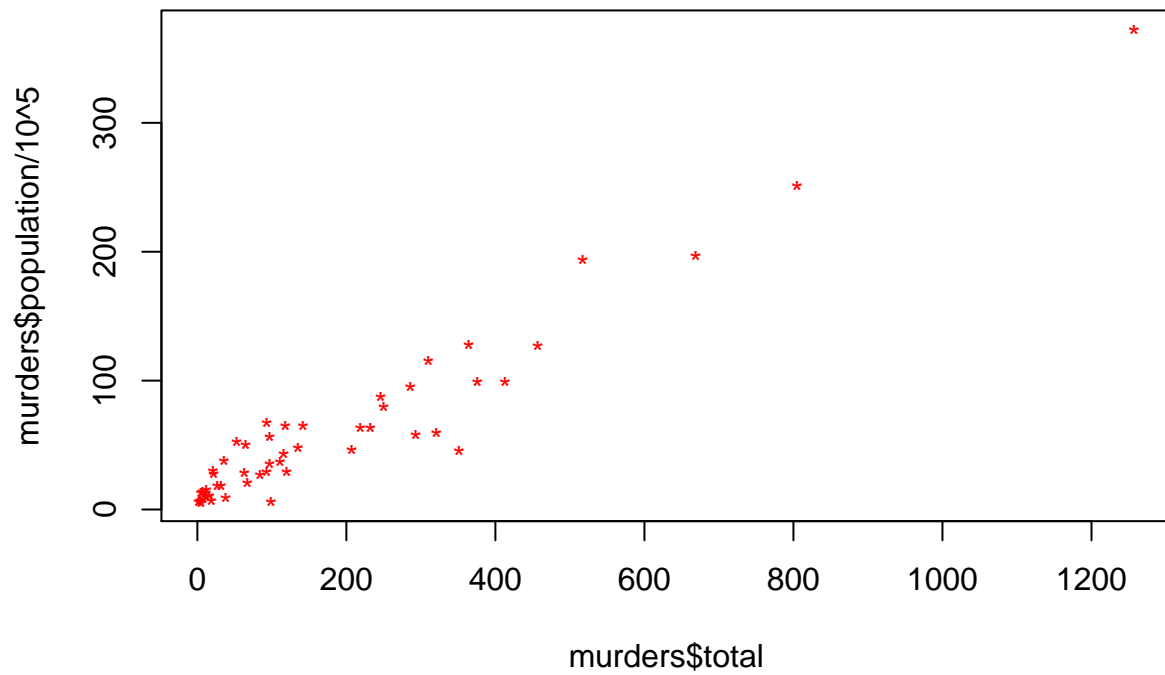
Para esta función podemos ajustar el tipo de punto con el parámetro *pch*:

```
plot(murders$total,murders$population/10^5,pch="*")
```



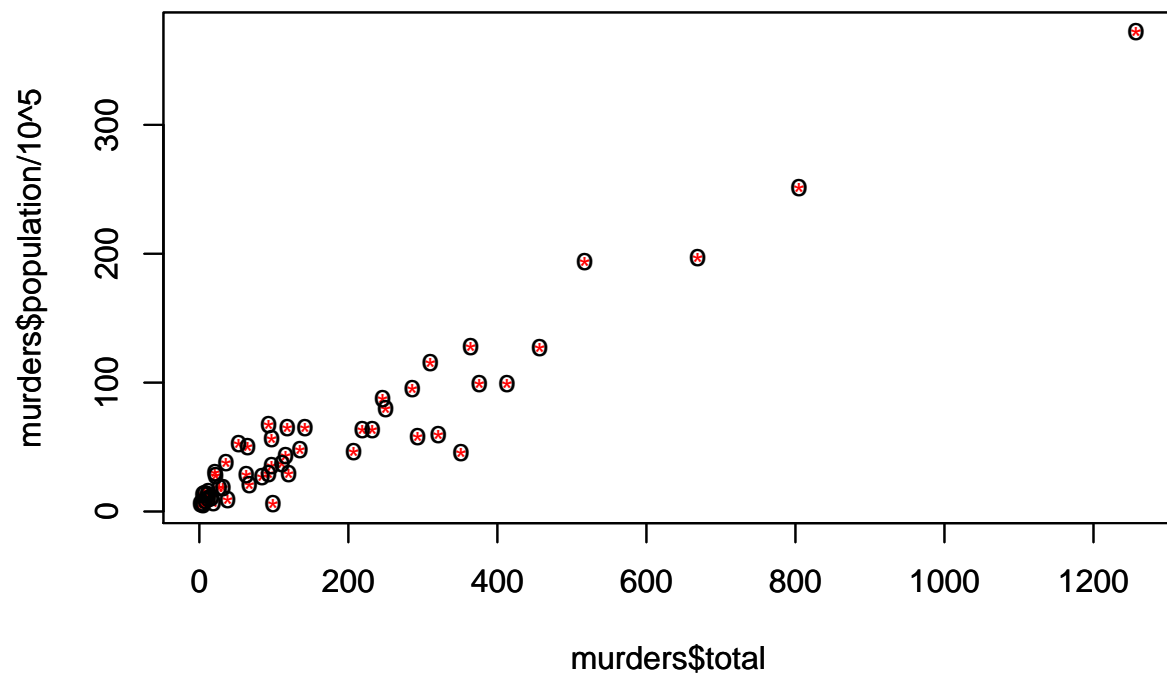
O el color:

```
plot(murders$total,murders$population/10^5,pch="*",col="red")
```



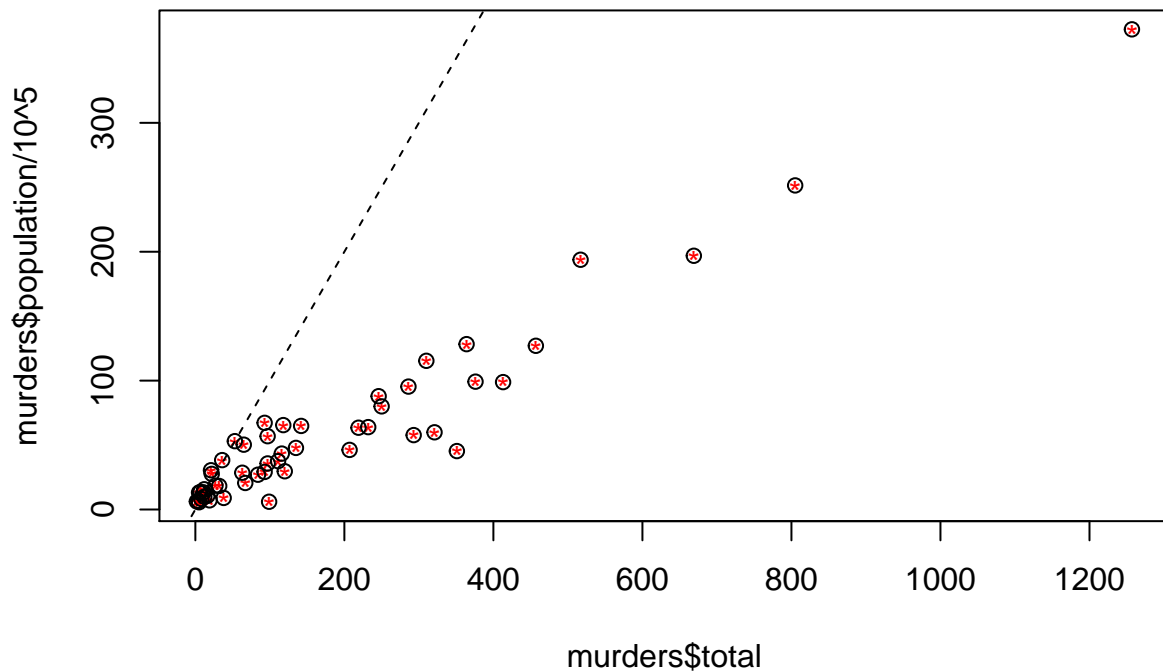
Cada vez que llamamos a la función `plot()` se abre una nueva ventana. Si queremos pintar en un gráfico ya existente tenemos dos opciones:

```
plot(murders$total,murders$population/10^5,pch="*",col="red")
par(new=T)
plot(murders$total,murders$population/10^5,pch="o")
```



o bien usar *points()* o *lines()* que pintan puntos o unen puntos por medio de líneas si los puntos ya están dibujados. La función *abline* pinta una línea de pendiente *b* y ordenada en el origen *a*.

```
plot(murders$total,murders$population/10^5,pch="*",col="red")
points(murders$total,murders$population/10^5,lty=2)
abline(a=0,b=1,lty=2)
```



El parámetro lty controla el tipo de línea (sólida, discontinua, etc)

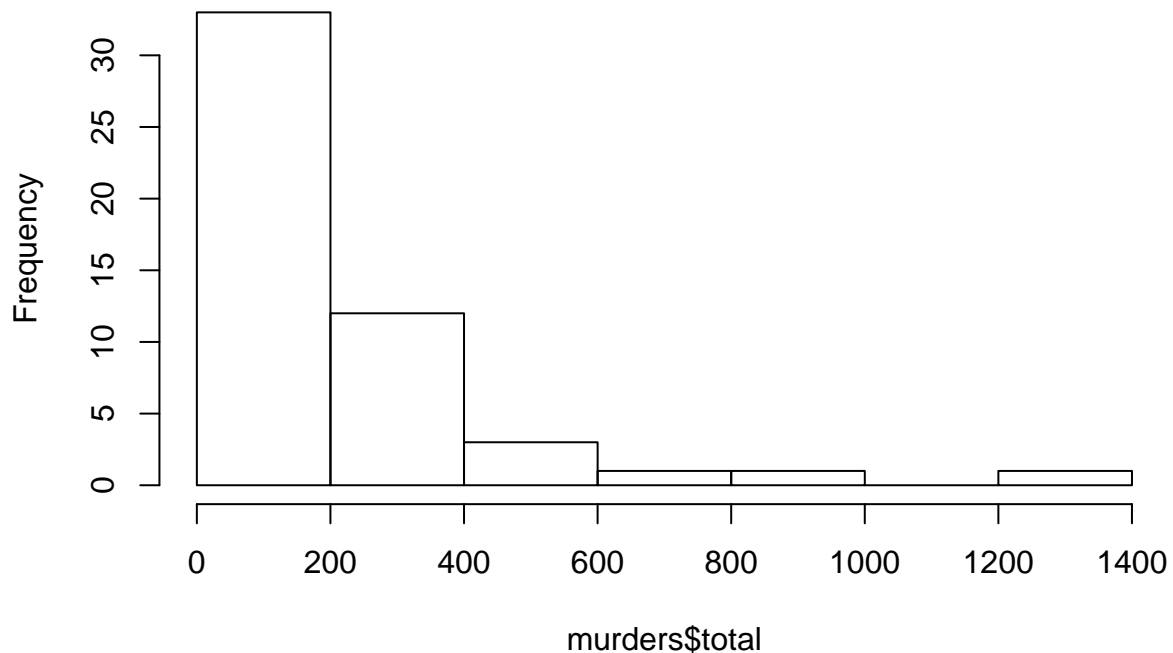
Estos simples gráficos ya nos muestran que hay una relación de tipo lineal entre el numero de asesinatos y la población total de un estado

## Histograms

Un histograma nos muestra la distribución de los elementos de una muestra. Es decir, nos dice cuántos elementos de cada tipo hay.

```
hist(murders$total)
```

## Histogram of murders\$total



Es decir, hay unos 32-33 estados con menos de 200 asesinatos; hay unos 10 estados con entre 200 y 400 asesinatos y el resto (~10 estados) sufrieron más de 400 asesinatos. EN particular parece que hay 1 estado con entre 1200 y 1400 asesinatos.

Vamos a comprobarlo con una tabla. Primero binarizamos nuestros resultados en bins de 200:

```
murders$total.bin=murders$total
murders$total.bin[which(murders$total<=200)]=200
murders$total.bin[which(murders$total>200 & murders$total<=400)]=400
murders$total.bin[which(murders$total>400 & murders$total<=600)]=600
murders$total.bin[which(murders$total>600 & murders$total<=800)]=800
murders$total.bin[which(murders$total>800 & murders$total<=1000)]=1000
murders$total.bin[which(murders$total>1000 & murders$total<=1200)]=1200
murders$total.bin[which(murders$total>1200 & murders$total<=1400)]=1400
table(murders$total.bin)
```

```
##
## 200 400 600 800 1000 1400
## 33 12 3 1 1 1
```

Arpovechamos para introducir la sintaxis de un loop en R. Podríamos haber utilizado un bucle para programar lo anterior:

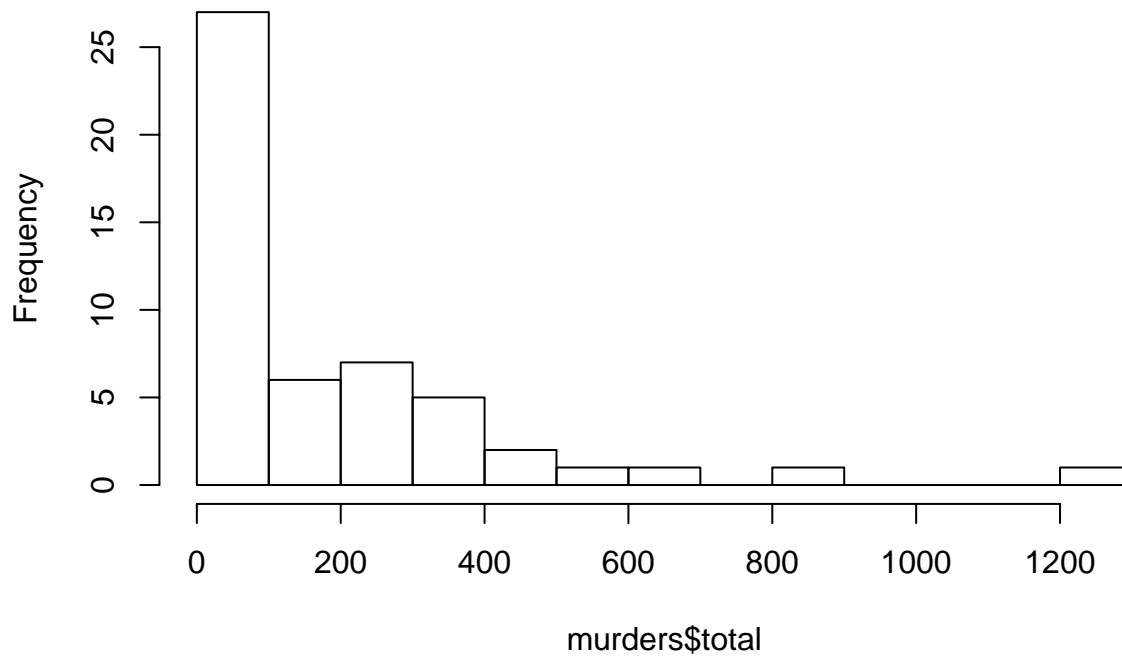
```
murders$total.bin=murders$total
for (i in 1:7){
  murders$total.bin[which(murders$total>200*(i-1) & murders$total<=200*i)]=200*i
}
table(murders$total.bin)
```

```
##
## 200  400  600  800 1000 1400
##  33   12   3    1    1    1
```

el número de bins puede cambiarse fácilmente con el comando *breaks()*

```
hist(murders$total,breaks=10)
```

## Histogram of murders\$total

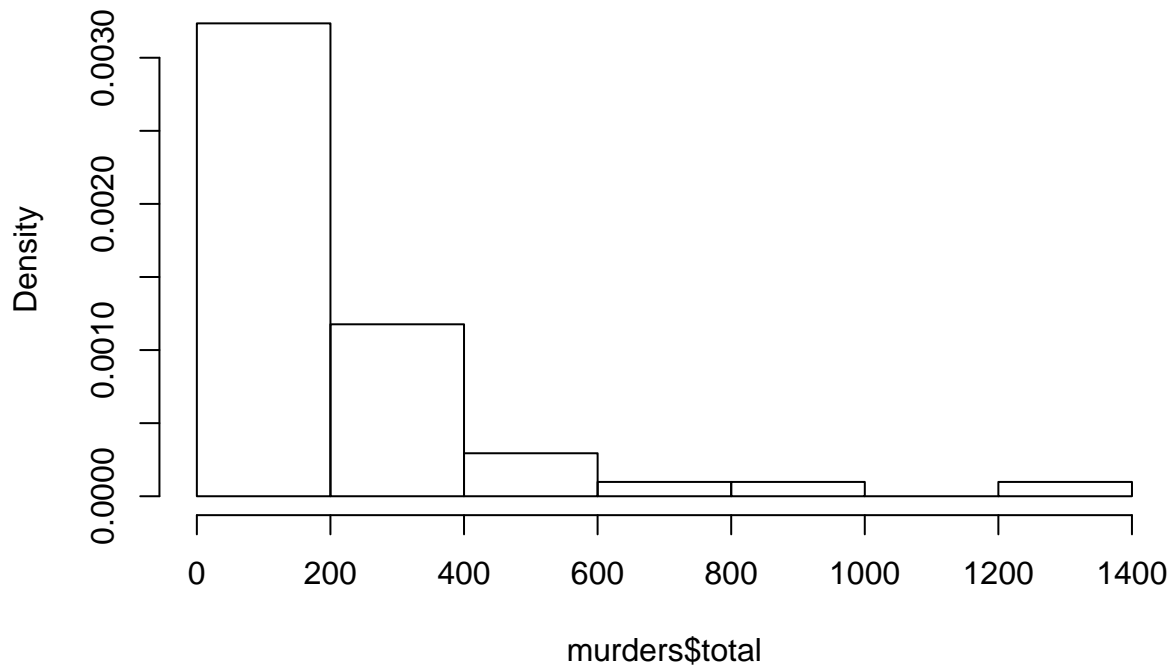


Y también podemos elegir ver la frecuencia (número de elementos en cada bin) o la probabilidad de tener un elemento en cada bin ( $\frac{\# \text{ elementos en bin}}{\text{total numero de elementos}}$ )

```
hist(murders$total,freq = F)
```



## Histogram of murders\$total

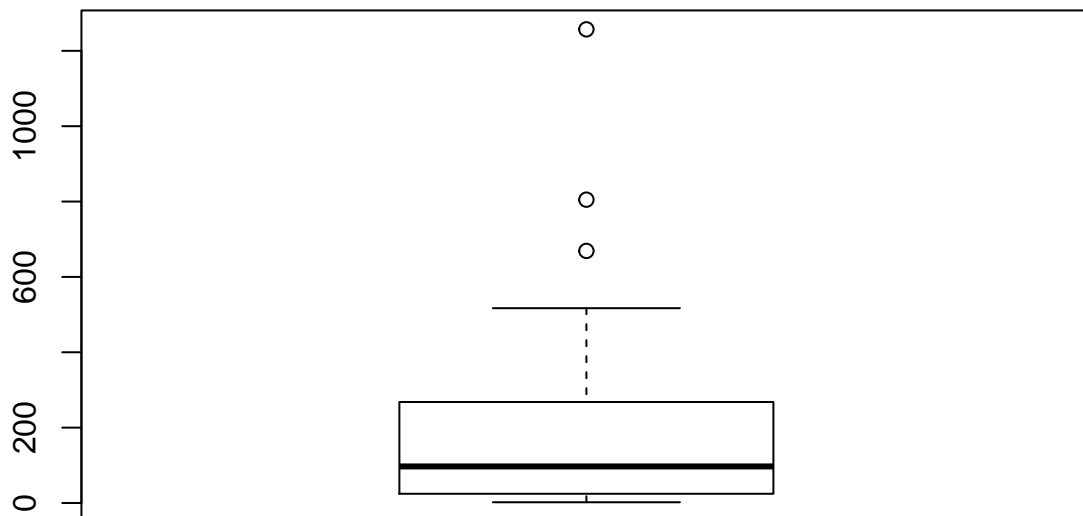


Con este tipo de plot obtenemos una idea de cómo es la distribución de los datos: no simétrica, con el valor mas probable entre 0 y 200 y con algunos valores muy distintos dl resto. Estas características serán las que miraremos más adelante cuando queramos caracterizar distribuciones de datos.

### Boxplot

Un boxplot también nos da pistas acerca de la distribución de un conjunto de datos

```
boxplot(murders$total)
```



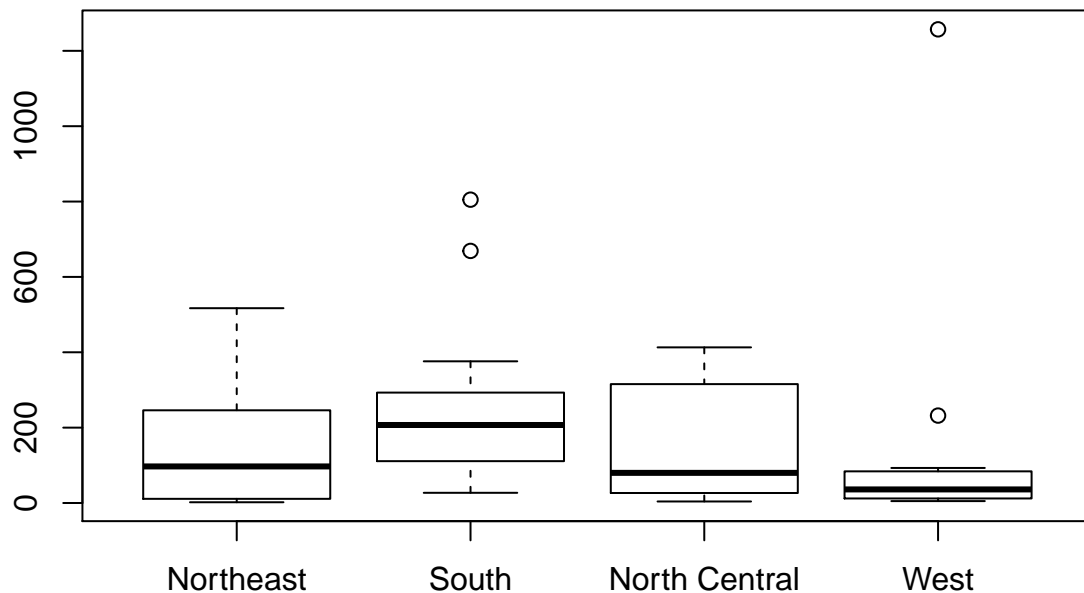
Vemos que la caja tiene una línea central (la mediana) que en este caso no está en el medio, lo cual significa que el 50% de los valores más pequeños están más cerca entre si que los valores del 50% superior. Además vemos tres puntos por encima de la línea (whishart). Estos son outliers. Tenemos 3 entre los datos más altos. Los outliers o valores extremos están más allá del valor que deja a su izquierda el 75% de la distribución multiplicado por 1.5. Todos estos datos se pueden observar usando la función `summary()` sobre un vector numérico:

```
summary(murders$total)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2.0   24.5    97.0   184.4   268.0   1257.0
```

Los boxplots son particularmente interesantes para comparar distribuciones de diferentes grupos de datos. Por ejemplo, como tenemos la información del número de asesinatos por region podemos hacer un boxplot del número de asesinatos para cada region:

```
boxplot(murders$total~murders$region)
```



La región con el menor número de asesinatos en general es “West”, aunque hay dos estados que son outliers. Podemos buscarlo:

```
murders.west<-filter(murders,region=="West")
murders.west[which.max(murders.west$total),]
```

```
##      state abb region population total total.bin
## 3 California CA   West   37253956  1257      1400
```

La región con un mayor número de asesinatos en general es “South”

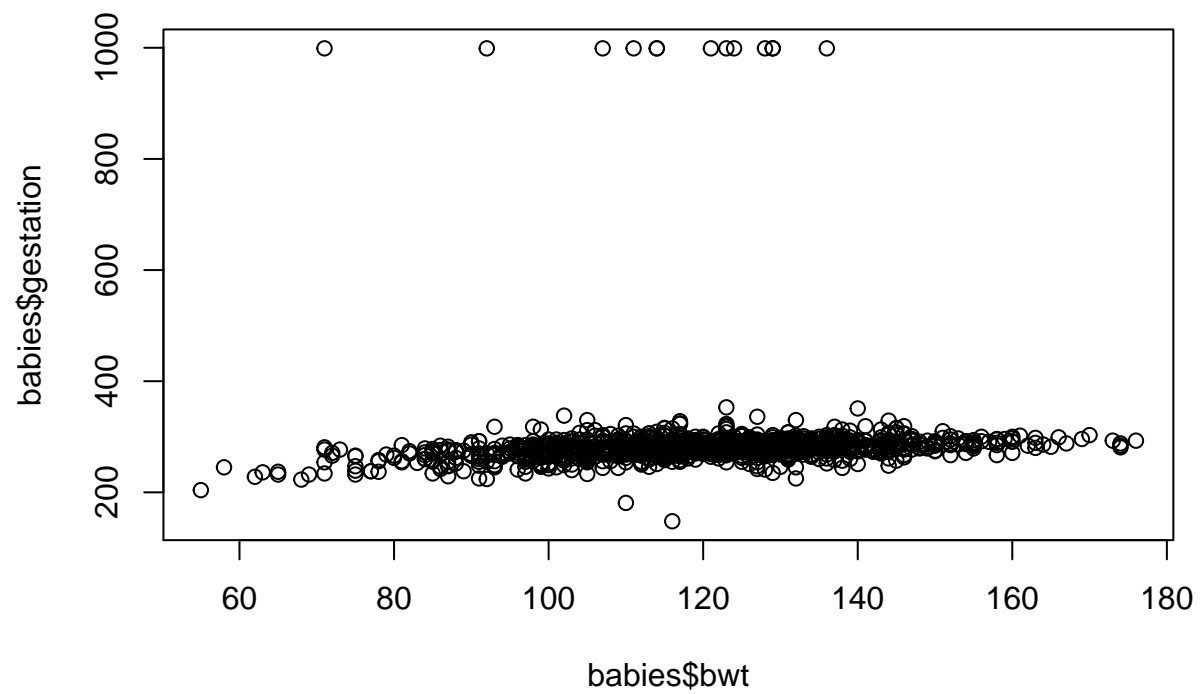
### Ejercicio #1: dataset *babies*

Utilizando el dataset “babies.txt”

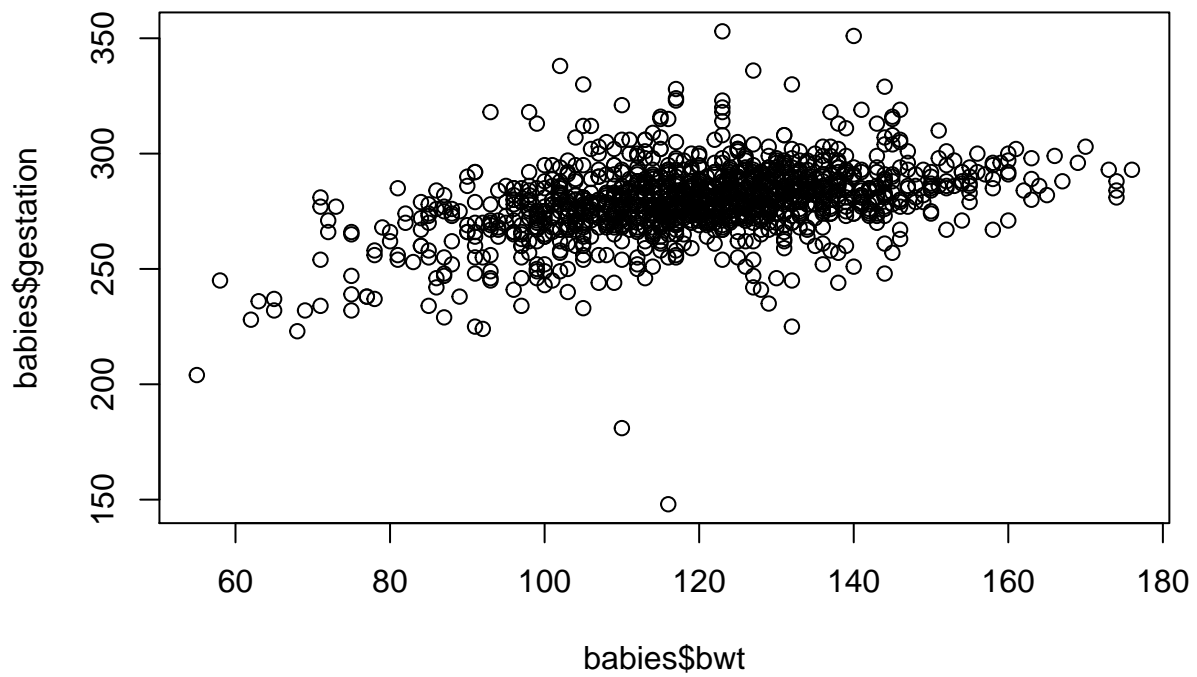
1. Utilizando un scatterplot *plot()* encontrar si existe una relación de algún tipo entre el peso al nacimiento de los bebés y la edad gestacional (en semanas)
2. Compara utilizando un boxplot la distribución de los pesos al nacer de los niños con madres fumadoras frente a aquellos con madres no fumadoras
3. Explora usando un histograma la distribución general de los pesos de los bebés.

- 
1. Utilizando un scatterplot *plot()* encontrar si existe una relación de algún tipo entre el peso al nacimiento de los bebés y la edad gestacional (en semanas)

```
setwd("C:/Users/fscabo/Desktop/MasterDataScience_KSchool/Ejercicios")
babies=read.delim("babies.txt",header=T,sep="\t",stringsAsFactors = F)
plot(babies$bwt,babies$gestation)
```

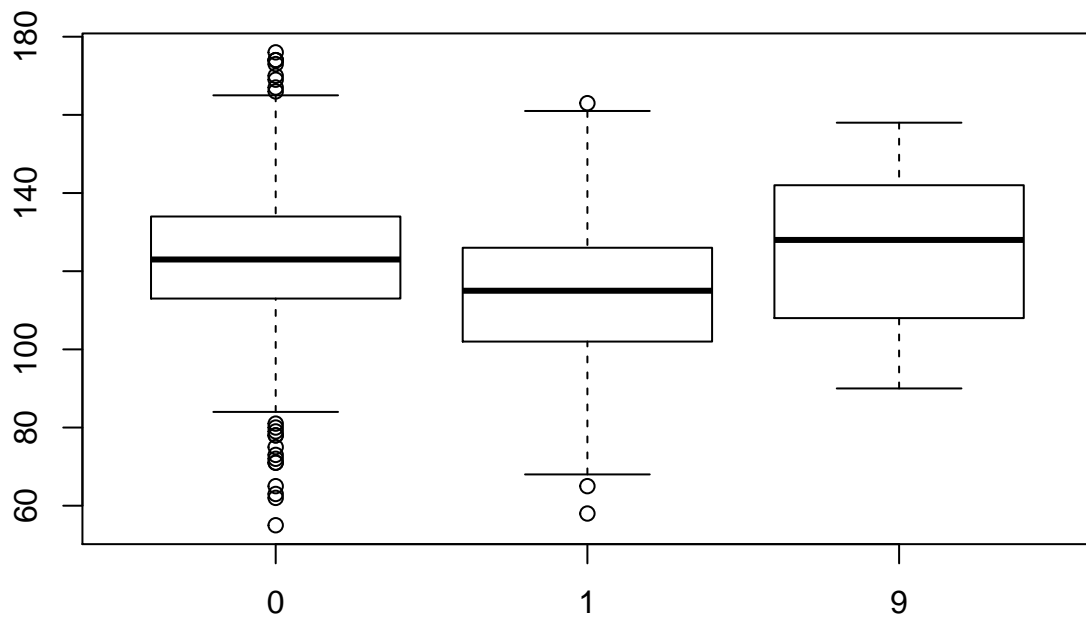


```
#necesitamos poner a NA los missing (999)  
babies$gestation[which(babies$gestation=="999")]=NA  
plot(babies$bwt,babies$gestation)
```



2. Compara utilizando un boxplot la distribución de los pesos al nacer de los niños con madres fumadoras frente a aquellos con madres no fumadoras

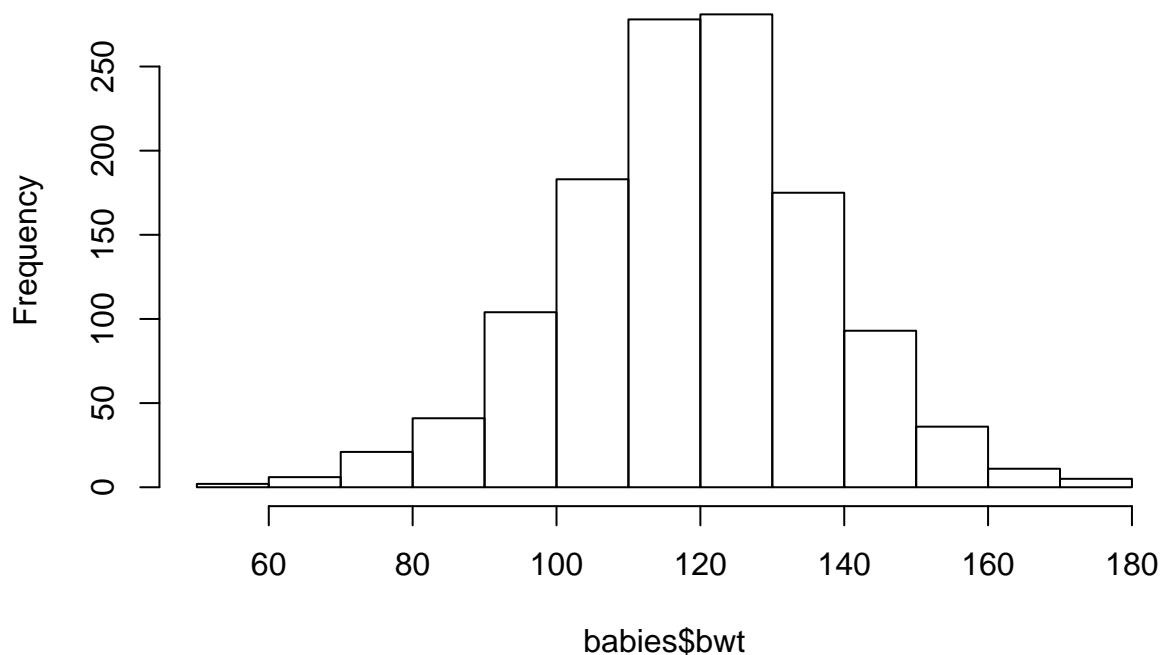
```
boxplot(babies$bwt~babies$smoke)
```



3. Explora usando un histograma la distribución general de los pesos de los bebés.

```
hist(babies$bwt)
```

## Histogram of babies\$bwt



## Sumarizando datos y detectando outliers

Cuando tenemos muchos datos lo primero que solemos querer hacer es intentar resumir la información en un sólo número. En principio esto debería de darnos una idea acerca de algunas de las características importantes de nuestros datos. Quéé medidas de sumarización se nos ocurren? La media, la mediana, la desviación estandar... ahora vamos a ver cuando usar cada una de ellas, cuando son y cuando no son informativas.

### Medidas de resumen paramétricas: Media y desviación standard

Utilizando el ejemplo “babies.txt”, calcular la media y la desviación estandar para los peso de los bebes:

```
mean(babies$bwt)
```

```
## [1] 119.5769
```

```
sqrt(var(babies$bwt))
```

```
## [1] 18.23645
```

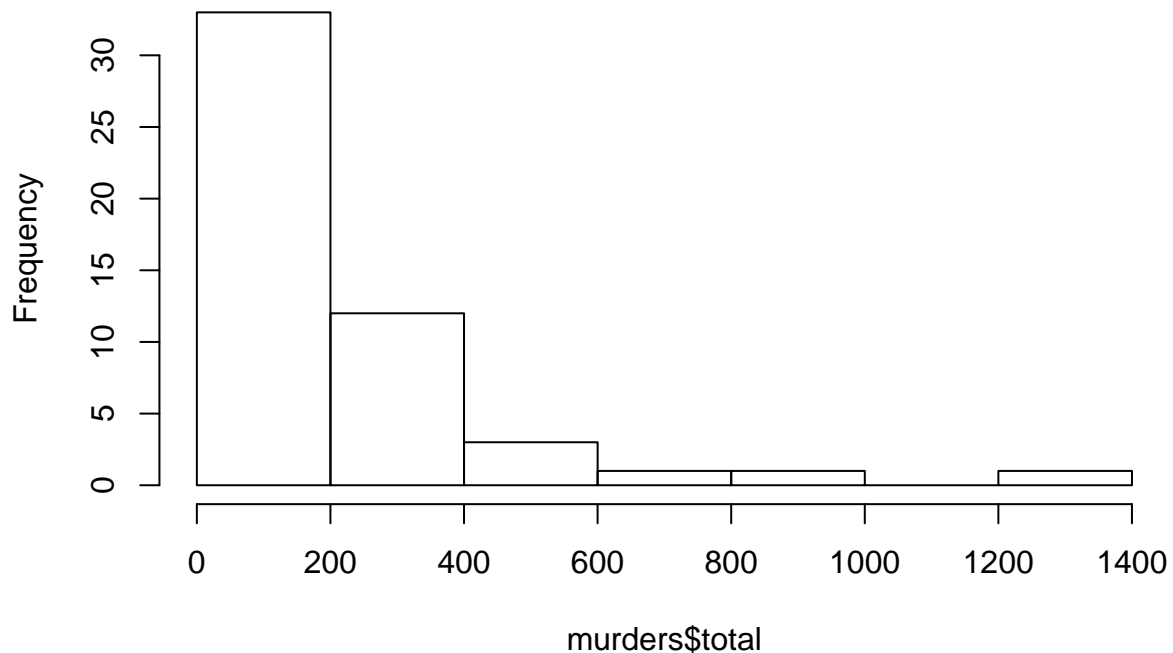
Recordando cómo era su distribución (simétrica, un sólo valor más probable...) parece que estos valores son bastantes informativos acerca de estos datos

### Medidas de resumen no paramétricas: Mediana, IQR

Sin embargo, si miramos el histograma del total de asesinatos del ejemplo murder y calculamos los mismos parámetros

```
hist(murders$total)
```

## Histogram of murders\$total



```
mean(murders$total)
```

```
## [1] 184.3725
```

```
sqrt(var(murders$total))
```

```
## [1] 236.1261
```

Como veis estos dos valores nos harian pensar que hay numeros de asesinatos negativos en algunos estados, no nos da pistas acerca del numero mas habitual de asesinatos que encontramos y no podriamos saber que el numero de asesinatos es tan alto como 1400 en otros. Hay otros parámetros para sumarizar datos que no siguen una distribución “normal”:

```
median(murders$total)
```

```
## [1] 97
```

```
IQR(murders$total)
```

```
## [1] 243.5
```

La mediana nos dice cual es el valor que deja el 50% de los datos a la izquierda de el y el 50% a su derecha. El IQR (Interquantile Range) nos da una idea de en que intervalo cae el 50% de los datos. Podemos calcular los quantiles de una distribucion usando:

```
summary(murders$total)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2.0   24.5   97.0  184.4  268.0 1257.0
```



```
summary(babies$bwt)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      55.0   108.8   120.0   119.6   131.0   176.0
```

Si un conjunto de datos es aproximadamente normal (simétrico, con un sólo valor más probable, etc) en ese caso la media y la mediana son muy parecidas, como en el caso del peso de los bebés. Sin embargo, cuando una distribución no es parecida a una normal (murders\$total) la media y la mediana son muy distintas y la media no suele ser muy informativa. Tampoco la desviación estándar. El IQR se calcula como  $Q_3 - Q_1$ . Todos los valores que estén alejados 1.5 veces el IQR del  $q_1$  o de  $q_3$  se consideran outliers:

```
summary(murders$total)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2.0    24.5    97.0   184.4   268.0  1257.0
```

```
q1=quantile(murders$total, p=0.25)
q1
```

```
## 25%
## 24.5
```

```
q3=quantile(murders$total, p=0.75)
q3
```

```
## 75%
## 268
```

```
iqr=(q3-q1)
iqr
```

```
## 75%
## 243.5
```

```
r <- c(q1 - 1.5*iqr, q3 + 1.5*iqr)
r
```

```
##      25%      75%
## -340.75  633.25
```

Buscamos los outliers:

```
which(murders$total<=r[1])
```

```
## integer(0)
```

```
which(murders$total>=r[2])
```

```
## [1]  5 10 44
```

```
murders[which(murders$total>=r[2]),]
```

```
##      state abb region population total total.bin
## 5  California CA  West   37253956  1257      1400
## 10   Florida  FL  South   19687653   669       800
## 44    Texas   TX  South   25145561   805      1000
```

Hay otro tipo de outliers aún más lejanos que son los *far\_out* outliers

```
r2 <- c(q1 - 3*iqr, q3 + 3*iqr)
r2
```

```
##      25%      75%
## -706.0  998.5
```

```
which(murders$total>=r2[2])
```

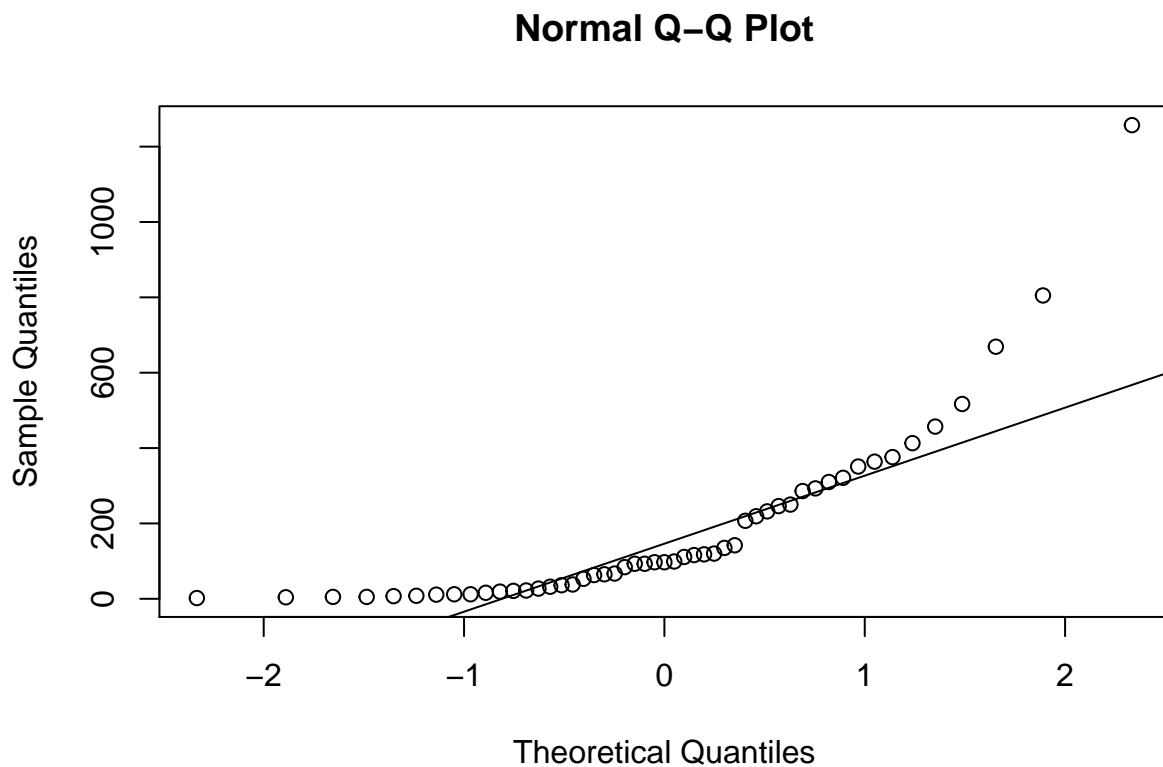
```
## [1] 5
```

```
murders[which(murders$total>=r2[2]),]
```

```
##      state abb region population total total.bin
## 5 California CA  West  37253956  1257      1400
```

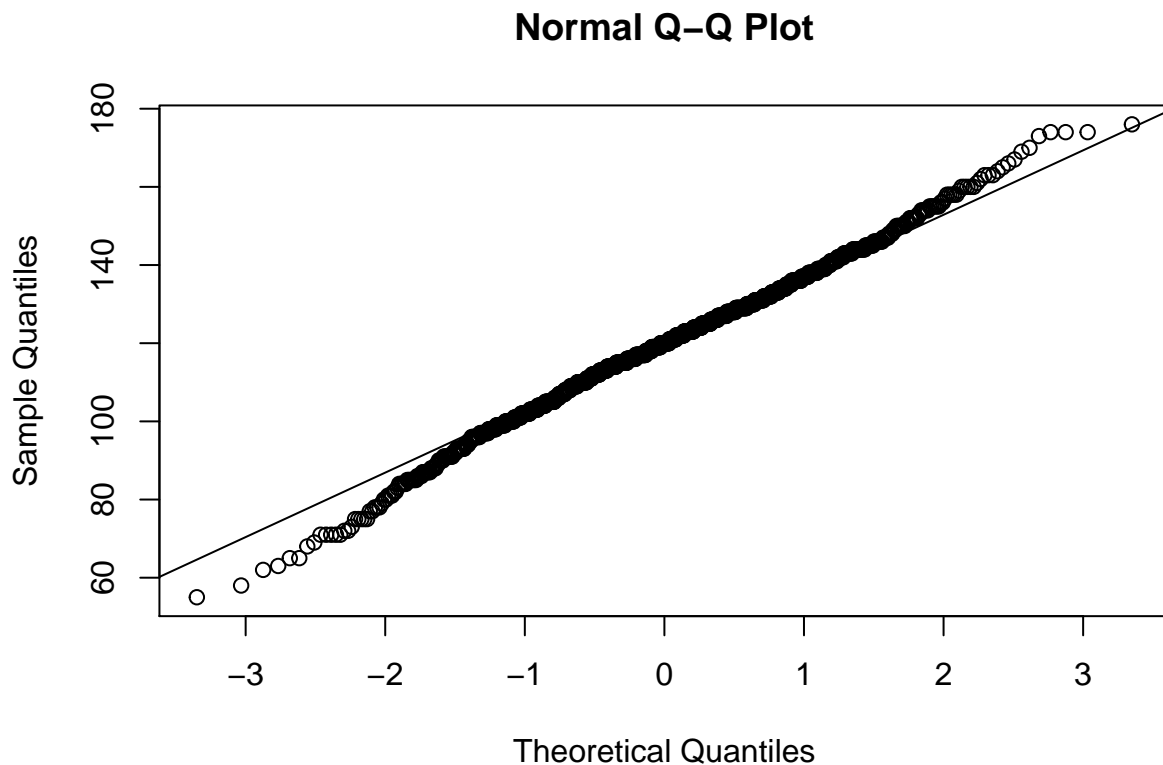
Un tipo de plot que no habíamos visto y que también es muy informativo es el que compara los quantiles de nuestros datos con los de una distribución normal:

```
qqnorm(murders$total)
qqline(murders$total)
```



Frente a esto, la distribución de los pesos de los bebés tienen una distribución mas o menos normal, con la excepción de que las colas son mas pesadas.

```
qqnorm(babies$bwt)
qqline(babies$bwt)
```



Finalmente, para datos no normales hay otra medida que nos da una idea de la dispersión, la *mad*: median absolute deviation. Se trata de la suma de la distancia absoluta entre cada valor y la mediana.

```
mad(babies$bwt)
```

```
## [1] 16.3086
```

```
mad(murders$total)
```

```
## [1] 126.021
```

## Sumarizando data con *dplyr()*

*summarize()*

Vamos a utilizar los datos de alturas del paquete *dslabs*

```
library(dslabs)
data(heights)
head(heights)
```

```
##      sex height
## 1  Male     75
## 2  Male     70
## 3  Male     68
## 4  Male     74
## 5  Male     61
## 6 Female    65
```

```
str(heights)
```

```
## 'data.frame':  1050 obs. of  2 variables:
## $ sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 1 1 1 1 2 ...
## $ height: num  75 70 68 74 61 65 66 62 66 67 ...
```

La función `summarize()` del paquete `dplyr` nos calcula cualquier agregado que le pidamos de un vector de un `data.frame` o de un `tibble`. Como el input era un `data.frame()` el output también lo es.

```
s <- heights %>%
  filter(sex == "Male") %>%
  summarize(average = mean(height), standard_deviation = sd(height))
s
```

```
##   average standard_deviation
## 1 69.31475          3.611024
```

```
str(s)
```

```
## 'data.frame':  1 obs. of  2 variables:
## $ average      : num 69.3
## $ standard_deviation: num 3.61
```

Como ya comentamos anteriormente, para datos que no siguen una distribución normal o gaussiana es mejor utilizar la mediana, `mad` o `IQR`. Podemos usar otra vez la función `summarize` para el data de total murders:

```
s <- murders %>%
  summarize(median = median(total), mad=mad(total),min=min(total),max=max(total))
s
```

```
##   median      mad min  max
## 1     97 126.021  2 1257
```

```
str(s)
```

```
## 'data.frame':  1 obs. of  4 variables:
## $ median: num 97
## $ mad   : num 126
## $ min   : num 2
## $ max   : num 1257
```

NOTA: con la función `summarize` solo podemos llamar funciones que devuelvan un solo valor.

### *dot*

Recordemos en el último ejercicio de la sesión II necesitábamos sumarizar el rate por estado del dataset `murders` para poder unirlo a la tabla con el rate de todos los países del mundo.

```
s <- murders %>%
  mutate(rate=total/population*100000) %>%
  summarize(mean(rate))
s
```

```
##   mean(rate)
## 1    2.779125
```

```
str(s)
```

```
## 'data.frame':  1 obs. of  1 variable:
## $ mean(rate): num 2.78
```

Como las funciones de dplyr devuelven el mismo tipo de objeto que su input en este caso queremos acceder sólo al valor que tienen almacenado. Podemos hacerlo así:

```
s

##   mean(rate)
## 1    2.779125
```

```
s %>% .$rate
```

```
## NULL
```

“.” simplemente reemplaza al objeto que pasamos por el pipe, en este caso s que es un data frame. Por eso accedemos su información con \$

```
s <- murders %>%
  summarize(rate=mean(total)/mean(population)*100000) %>%
  .$rate
s
```

```
## [1] 3.034555
```

group\_by()

```
babies.new<-babies %>%
  select(bwt, smoke) %>%
  group_by(smoke)
str(babies.new)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  1236 obs. of  2 variables:
## $ bwt : int  120 113 128 123 108 136 138 132 120 143 ...
## $ smoke: int   0 0 1 0 1 0 0 0 0 1 ...
## - attr(*, "vars")= chr "smoke"
## - attr(*, "drop")= logi TRUE
## - attr(*, "indices")=List of 3
## ..$ : int   0 1 3 5 6 7 8 10 14 15 ...
## ..$ : int   2 4 9 11 12 13 16 17 19 21 ...
## ..$ : int  169 218 254 255 432 600 641 665 671 920
## - attr(*, "group_sizes")= int  742 484 10
## - attr(*, "biggest_group_size")= int 742
## - attr(*, "labels")='data.frame':  3 obs. of  1 variable:
## ..$ smoke: int   0 1 9
## ..- attr(*, "vars")= chr "smoke"
## ..- attr(*, "drop")= logi TRUE
```

```
babies.new
```

```
## # A tibble: 1,236 x 2
## # Groups:   smoke [3]
##       bwt smoke
##   <int> <int>
## 1   120     0
## 2   113     0
## 3   128     1
## 4   123     0
## 5   108     1
## 6   136     0
## 7   138     0
## 8   132     0
```

```
## 9 120 0
## 10 143 1
## # ... with 1,226 more rows
```

```
babies %>%
  select(bwt, smoke) %>%
  group_by(smoke) %>%
  summarize(mean(bwt))
```

```
## # A tibble: 3 x 2
##   smoke `mean(bwt)`
##   <int>     <dbl>
## 1     0     123.
## 2     1     114.
## 3     9     127.
```

Ordenar data.frames: *arrange()* *top\_n()*

La función *arrange()* ordena tablas enteras por una variable

```
murders %>% arrange(population) %>% head()
```

```
##           state abb      region population total total.bin
## 1      Wyoming  WY      West      563626      5      200
## 2 District of Columbia DC      South      601723     99      200
## 3      Vermont  VT      Northeast      625741      2      200
## 4 North Dakota ND North Central      672591      4      200
## 5      Alaska  AK      West      710231     19      200
## 6 South Dakota SD North Central      814180      8      200
```

```
murders %>% mutate(rate=total/population*100000)%>%
  arrange(rate) %>%
  head()
```

```
##           state abb      region population total total.bin      rate
## 1      Vermont  VT      Northeast      625741      2      200 0.3196211
## 2 New Hampshire NH      Northeast      1316470      5      200 0.3798036
## 3      Hawaii  HI      West      1360301      7      200 0.5145920
## 4 North Dakota ND North Central      672591      4      200 0.5947151
## 5      Iowa  IA North Central      3046355     21      200 0.6893484
## 6      Idaho  ID      West      1567582     12      200 0.7655102
```

Si tenemos empates podemos usar una segunda columna para deshacer dicho empate:

```
murders %>% mutate(rate=total/population*100000)%>%
  arrange(total,rate) %>%
  head()
```

```
##           state abb      region population total total.bin      rate
## 1      Vermont  VT      Northeast      625741      2      200 0.3196211
## 2 North Dakota ND North Central      672591      4      200 0.5947151
## 3 New Hampshire NH      Northeast      1316470      5      200 0.3798036
## 4      Wyoming  WY      West      563626      5      200 0.8871131
## 5      Hawaii  HI      West      1360301      7      200 0.5145920
## 6 South Dakota SD North Central      814180      8      200 0.9825837
```

Por último podemos seleccionar las primeras filas de un data.frame o de un tibble usando la función *top\_n()*. Nota que la función *desc()* indica que se ordena de manera descendente el data.frame.

```
murders %>% mutate(rate=total/population*100000)%>%
  arrange(desc(rate)) %>%
  top_n(10)
```

```
## Selecting by rate
```

```
##           state abb      region population total total.bin
## 1 District of Columbia DC      South      601723      99      200
## 2      Louisiana LA      South     4533372     351      400
## 3      Missouri MO North Central     5988927     321      400
## 4      Maryland MD      South     5773552     293      400
## 5      South Carolina SC      South     4625364     207      400
## 6      Delaware DE      South      897934      38      200
## 7      Michigan MI North Central     9883640     413      600
## 8      Mississippi MS      South     2967297     120      200
## 9      Georgia GA      South     9920000     376      400
## 10     Arizona AZ      West      6392017     232      400
##           rate
## 1 16.452753
## 2  7.742581
## 3  5.359892
## 4  5.074866
## 5  4.475323
## 6  4.231937
## 7  4.178622
## 8  4.044085
## 9  3.790323
## 10 3.629527
```

## Ejercicio #2: Distribución de las alturas de los estudiantes

Recogemos la altura, el género y la edad de todos los miembros de la clase.

- Escribir los datos en un fichero y guardarlo en tu directorio de trabajo con el nombre: “alturas.txt”
- Leelo en R en el objeto “altura”
- Haz un scatterplot que relacione la altura con la edad
- Que distribución tiene la altura? Y la edad? Cual sería la mejor forma de sumarizar la altura? Y la edad?
- Haz un boxplot que muestre la distribución de alturas en hombres frente a mujeres
- Hay algun outlier?
- Utilizando *dplyr* sumariza la altura de los hombres y de las mujeres por separado
- quienes son los tres hombres mas altos de la clase? Y las tres mujeres más jóvenes?