

# R-PL1

Gabriel López, Sergio Sanz, Álvaro Zamorano

October 10, 2019

En esta parte de la práctica trabajaremos con el fichero `satelites.txt`.

En primer lugar hay que leer este fichero, para ello usamos la función:

```
> satelites<-read.table("satelites.txt")
```

Para trabajar con la variable `radio`, y hacer este trabajo más cómodo, la cargamos en una variable:

```
> Radio<-satelites$Radio
```

En el primer análisis de los datos se cuantifica la **frecuencia** de aparición de los mismos.

## 1. *Frecuencia absoluta:*

```
> frabsradio<-table(Radio)  
> write.table(frabsradio, "C:/Users/tromp/Documents/1.Curso_2019-20/1.Cuatrimetestre/  
>
```

---

| "Radio" | "Freq" |
|---------|--------|
| "13"    | 1      |
| "15"    | 1      |
| "16"    | 1      |
| "20"    | 2      |
| "22"    | 1      |
| "27"    | 1      |
| "29"    | 1      |
| "30"    | 1      |
| "33"    | 1      |
| "34"    | 1      |
| "42"    | 1      |

---

## 2. *Frecuencia absoluta acumulada:*

```
> frabsacumradio<-cumsum(table(Radio))  
> frabsacumradio
```

```
13 15 16 20 22 27 29 30 33 34 42
1  2  3  5  6  7  8  9 10 11 12
```

3. *Frecuencia relativa:* En este caso es necesario crear una función para poder calcular este valor. La función es:

```
> frecrel<-function(Radio){table(Radio)/length(Radio)}
> frecrel(Radio)
```

```
Radio
      13      15      16      20      22      27      29      30
0.08333333 0.08333333 0.08333333 0.16666667 0.08333333 0.08333333 0.08333333 0.08333333
```

4. *Frecuencia relativa acumulada:* Haremos uso de la función definida anteriormente:

```
> frecrelacum<-function(Radio){cumsum(table(Radio)/length(Radio))}
> frecrelacum(Radio)
```

```
      13      15      16      20      22      27      29      30
0.08333333 0.16666667 0.25000000 0.41666667 0.50000000 0.58333333 0.66666667 0.75000000
```

El segundo análisis de los datos se basa en calcular la **media aritmética**:

```
> mr=mean(Radio)
> mr
```

```
[1] 25.08333
```

El tercer análisis de los datos se basa en calcular las **medidas de dispersión**:

1. *Desviación típica:* Para corregir los resultados, se hace el cálculo a través de:

```
> sdr<-sd(Radio)/sqrt(12/11)
> sdr
```

```
[1] 8.47996
```

2. *Varianza:* Al igual que en el caso anterior es necesario corregir el resultado por lo que se usa:

```
> varr<-var(Radio)*11/12
> varr
```

```
[1] 71.90972
```

El cuarto análisis de los datos se basa en las **medidas de ordenación**, antes de los cálculos es necesario ordenar los datos en función de la variable usada, en este caso el radio.

```
> so<-satelites[order(Radio),]
```

Realmente no sería necesario ordenar los datos, ya que R se encarga de ello en caso de no hacerlo. Se procede a realizar los cálculos:

1. *Mediana:*

```
> mediant<-median(Radio)
> mediant
```

```
[1] 24.5
```

2. *Cuartiles:*

```
> cuar1<-quantile(Radio,0.25)
> cuar1
```

```
25%
19
```

```
> cuar2<-quantile(Radio,0.5)
> cuar2
```

```
50%
24.5
```

```
> cuar3<-quantile(Radio,0.75)
> cuar3
```

```
75%
30.75
```

```
> cuar54<-quantile(Radio,0.54)
> cuar54
```

```
54%
26.7
```

A continuación pasaremos a trabajar con un fichero generado por SPSS, `cardata.sav`.

En primer lugar hay que leer este fichero pero no disponemos de la librería necesaria para hacerlo, para cargarla usamos:

```
> library(foreign)
```

Esta librería se trata de una librería estándar de R.

Una vez cargada procedemos a su lectura

```
> A<-read.spss("cardata.sav")
```

Para trabajar con la variable mpg, y hacer este trabajo más cómodo, la cargamos en una variable:

```
> mpg<-A$mpg
```

La variable mpg contiene valores NA, es decir, valores que no se encuentran disponibles por lo que es imposible realizar cálculos con ella. Para eliminar estos valores usamos:

```
> mpg<-mpg[!is.na(mpg)]
```

En el primer análisis de los datos se cuantifica la **frecuencia** de aparición de los mismos.

1. *Frecuencia absoluta:*

```
> frabsmpg<-table(mpg)
> frabsmpg
```

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| mpg  | 15.5 | 16.2 | 16.5 | 16.9 | 17   | 17.5 | 17.6 | 17.7 | 18.1 | 18.2 | 18.5 | 18.6 | 19.1 | 19.2 | 19.4 | 19.8 | 19.9 | 20.3 |
|      | 1    | 1    | 1    | 1    | 2    | 1    | 2    | 1    | 2    | 1    | 1    | 1    | 1    | 3    | 2    | 1    | 1    | 1    |
| 23.8 | 23.9 | 24   | 24.2 | 24.3 | 25   | 25.1 | 25.4 | 25.8 | 26   | 26.4 | 26.6 | 26.8 | 27   | 27.2 | 27.4 | 27.5 | 27.6 | 27.7 |
|      | 1    | 2    | 1    | 1    | 1    | 1    | 1    | 2    | 1    | 1    | 2    | 1    | 1    | 4    | 3    | 1    | 1    | 1    |
| 31.6 | 31.8 | 31.9 | 32   | 32.1 | 32.2 | 32.3 | 32.4 | 32.7 | 32.8 | 32.9 | 33   | 33.5 | 33.7 | 33.8 | 34   | 34.1 | 34.2 | 34.3 |
|      | 1    | 1    | 1    | 3    | 1    | 1    | 1    | 2    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 2    | 2    | 2    |
| 38.1 | 39   | 39.1 | 39.4 | 40.8 | 40.9 | 41.5 | 43.1 | 43.4 | 44   | 44.3 | 44.6 | 46.6 |      |      |      |      |      |      |
|      | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |      |      |      |      |      |

2. *Frecuencia absoluta acumulada:*

```
> frabsacummpg<-cumsum(table(mpg))
> frabsacummpg
```

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 15.5 | 16.2 | 16.5 | 16.9 | 17   | 17.5 | 17.6 | 17.7 | 18.1 | 18.2 | 18.5 | 18.6 | 19.1 | 19.2 | 19.4 | 19.8 | 19.9 | 20.3 | 20.4 |
| 1    | 2    | 3    | 4    | 6    | 7    | 9    | 10   | 12   | 13   | 14   | 15   | 16   | 19   | 21   | 22   | 23   | 24   | 25   |
| 23.8 | 23.9 | 24   | 24.2 | 24.3 | 25   | 25.1 | 25.4 | 25.8 | 26   | 26.4 | 26.6 | 26.8 | 27   | 27.2 | 27.4 | 27.5 | 27.6 | 27.7 |
|      | 46   | 48   | 49   | 50   | 51   | 52   | 53   | 55   | 56   | 57   | 58   | 60   | 61   | 65   | 68   | 69   | 70   | 71   |
| 31.6 | 31.8 | 31.9 | 32   | 32.1 | 32.2 | 32.3 | 32.4 | 32.7 | 32.8 | 32.9 | 33   | 33.5 | 33.7 | 33.8 | 34   | 34.1 | 34.2 | 34.3 |
|      | 93   | 94   | 95   | 98   | 99   | 100  | 101  | 103  | 104  | 105  | 106  | 107  | 108  | 109  | 110  | 112  | 114  | 115  |
| 38.1 | 39   | 39.1 | 39.4 | 40.8 | 40.9 | 41.5 | 43.1 | 43.4 | 44   | 44.3 | 44.6 | 46.6 |      |      |      |      |      |      |
|      | 142  | 143  | 144  | 145  | 146  | 147  | 148  | 149  | 150  | 151  | 152  | 153  | 154  |      |      |      |      |      |

3. *Frecuencia relativa:* En este caso es necesario crear una función para poder calcular este valor. La función es:

```
> frecrel<-function(mpg){table(mpg)/length(mpg)}
> frecrel(mpg)
```

| mpg         | 15.5        | 16.2        | 16.5        | 16.9        | 17          | 17.5        | 17.6 |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|
| 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.012987013 | 0.006493506 | 0.012987013 | 0    |
| 19.2        | 19.4        | 19.8        | 19.9        | 20.2        | 20.3        | 20.5        | 0    |
| 0.019480519 | 0.012987013 | 0.006493506 | 0.006493506 | 0.025974026 | 0.006493506 | 0.012987013 | 0    |
| 22.3        | 22.4        | 23          | 23.2        | 23.5        | 23.6        | 23.7        | 0    |
| 0.006493506 | 0.006493506 | 0.012987013 | 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0    |
| 25.1        | 25.4        | 25.8        | 26          | 26.4        | 26.6        | 26.8        | 0    |
| 0.006493506 | 0.012987013 | 0.006493506 | 0.006493506 | 0.006493506 | 0.012987013 | 0.006493506 | 0    |
| 28.1        | 28.4        | 28.8        | 29          | 29.5        | 29.8        | 29.9        | 0    |
| 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.012987013 | 0.006493506 | 0    |
| 31.5        | 31.6        | 31.8        | 31.9        | 32          | 32.1        | 32.2        | 0    |
| 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.019480519 | 0.006493506 | 0.006493506 | 0    |
| 33.5        | 33.7        | 33.8        | 34          | 34.1        | 34.2        | 34.3        | 0    |
| 0.006493506 | 0.006493506 | 0.006493506 | 0.012987013 | 0.012987013 | 0.006493506 | 0.006493506 | 0    |
| 36          | 36.1        | 36.4        | 37          | 37.2        | 37.3        | 37.7        | 0    |
| 0.032467532 | 0.012987013 | 0.006493506 | 0.019480519 | 0.006493506 | 0.006493506 | 0.006493506 | 0    |
| 40.9        | 41.5        | 43.1        | 43.4        | 44          | 44.3        | 44.6        | 0    |
| 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0.006493506 | 0    |

4. *Frecuencia relativa acumulada:* Haremos uso de la función definida anteriormente:

```
> frecrelacum<-function(mpg){cumsum(table(mpg)/length(mpg))}
> frecrelacum(mpg)
```

|             | 15.5        | 16.2        | 16.5        | 16.9        | 17          | 17.5        | 17.6 |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|
| 0.006493506 | 0.012987013 | 0.019480519 | 0.025974026 | 0.038961039 | 0.045454545 | 0.058441558 | 0    |
| 19.2        | 19.4        | 19.8        | 19.9        | 20.2        | 20.3        | 20.5        | 0    |
| 0.123376623 | 0.136363636 | 0.142857143 | 0.149350649 | 0.175324675 | 0.181818182 | 0.194805195 | 0    |
| 22.3        | 22.4        | 23          | 23.2        | 23.5        | 23.6        | 23.7        | 0    |
| 0.246753247 | 0.253246753 | 0.266233766 | 0.272727273 | 0.279220779 | 0.285714286 | 0.292207792 | 0    |
| 25.1        | 25.4        | 25.8        | 26          | 26.4        | 26.6        | 26.8        | 0    |
| 0.344155844 | 0.357142857 | 0.363636364 | 0.370129870 | 0.376623377 | 0.389610390 | 0.396103896 | 0    |
| 28.1        | 28.4        | 28.8        | 29          | 29.5        | 29.8        | 29.9        | 0    |
| 0.487012987 | 0.493506494 | 0.500000000 | 0.506493506 | 0.512987013 | 0.525974026 | 0.532467532 | 0    |
| 31.5        | 31.6        | 31.8        | 31.9        | 32          | 32.1        | 32.2        | 0    |
| 0.597402597 | 0.603896104 | 0.610389610 | 0.616883117 | 0.636363636 | 0.642857143 | 0.649350649 | 0    |
| 33.5        | 33.7        | 33.8        | 34          | 34.1        | 34.2        | 34.3        | 0    |
| 0.701298701 | 0.707792208 | 0.714285714 | 0.727272727 | 0.740259740 | 0.746753247 | 0.753246753 | 0    |
| 36          | 36.1        | 36.4        | 37          | 37.2        | 37.3        | 37.7        | 0    |
| 0.831168831 | 0.844155844 | 0.850649351 | 0.870129870 | 0.876623377 | 0.883116883 | 0.889610390 | 0    |
| 40.9        | 41.5        | 43.1        | 43.4        | 44          | 44.3        | 44.6        | 0    |
| 0.954545455 | 0.961038961 | 0.967532468 | 0.974025974 | 0.980519481 | 0.987012987 | 0.993506494 | 1    |

El segundo análisis de los datos se basa en calcular la **media aritmética:**

```
> mm<-mean(mpg)
> mm
```

```
[1] 28.79351
```

El tercer análisis de los datos se basa en calcular las **medidas de dispersión**:

1. *Desviación típica*: Para corregir los resultados, se hace el cálculo a través de:

```
> sdm<-sd(mpg)/sqrt(12/11)
> sdm
```

```
[1] 7.063141
```

2. *Varianza*: Al igual que en el caso anterior es necesario corregir el resultado por lo que se usa:

```
> varm<-var(mpg)*11/12
> varm
```

```
[1] 49.88796
```

El cuarto análisis de los datos se basa en las **medidas de ordenación**.,

1. *Mediana*:

```
> mediantm<-median(mpg)
> mediantm
```

```
[1] 28.9
```

2. *Cuartiles*:

```
> cuar1m<-quantile(mpg,0.25)
> cuar1m
```

```
25%
22.55
```

```
> cuar2m<-quantile(mpg,0.5)
> cuar2m
```

```
50%
28.9
```

```
> cuar3m<-quantile(mpg,0.75)
> cuar3m
```

```
75%
34.275
```

```
> cuar54m<-quantile(mpg,0.54)
> cuar54m
```

```
54%
30
```

En la segunda parte de la práctica vamos a trabajar con una base de datos descargada de *Kaggle* cuyos datos pertenecen a los jugadores de FIFA 19.

El archivo se encuentra en formato *csv*, para proceder a su lectura vamos a usar la librería **readr** por lo que es necesario instalarla mediante:

```
> install.packages("readr")
```

Una vez instala se carga en R haciendo uso de:

```
> library("readr")
```

Por último leemos el archivo.

```
> fifa<-read_csv("fifa19.csv")
```

Trabajaremos con la variable **Age** por lo que vamos a cargarla en una variable local para facilitar el trabajo.

```
> edad<-fifa$Age
```

Con el objetivo de observar las **frecuencias absolutas** de la variable crearemos un histograma con ellas. Hemos delegado la creación del histograma en una función externa la cual almacena este en un archivo *png* para posteriormente poder incluirlo correctamente al documento.

La función es la siguiente:

```
> source("histograma.R")
> histograma

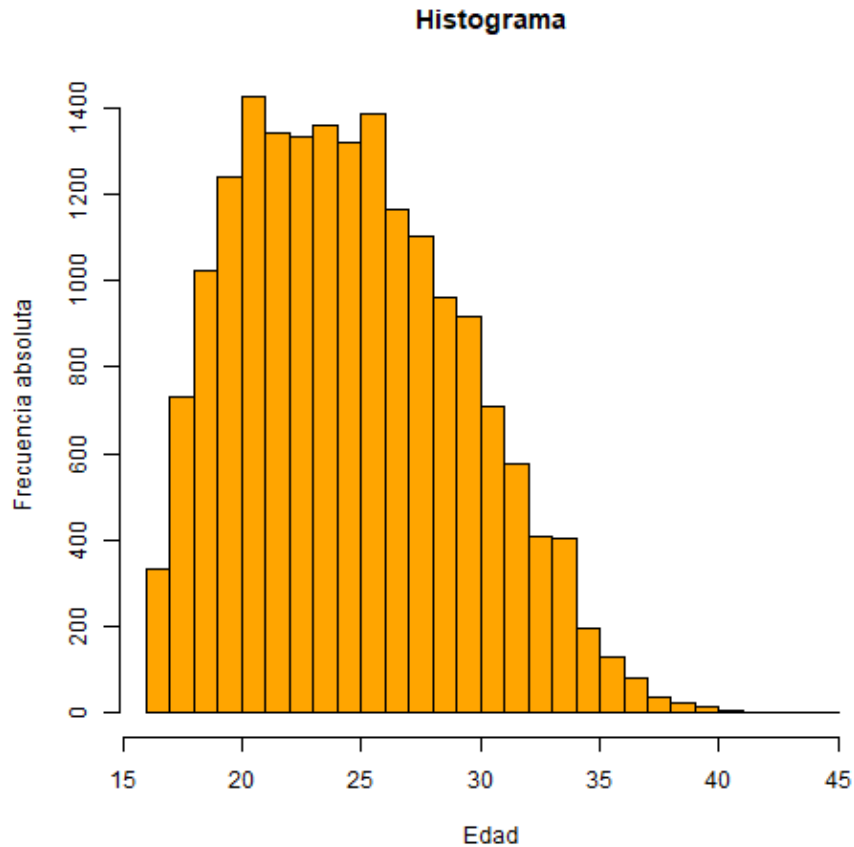
function(var,name,ruta) {
  png(ruta)

  h<-hist(var, col='orange', breaks=40, xlab=name,
          ylab="Frecuencia absoluta", main ="Histograma")

  dev.off()
}
```

Procedemos a su realización:

```
> h<-histograma(edad, "Edad", "./hist.png")
```



Procedemos a la separación de la variable en **clases de equivalencia** para calcular de ellas los distintos tipos de frecuencia estudiados. Esto se hace gracias al uso de la librería **fdth** y para poder utilizarla, como se ha dicho anteriormente es necesario realizar lo siguiente:

```
> install.packages("fdth")
> library("fdth")
```

Procedemos al cálculo en sí:

```
> dist <- fdt(edad,breaks="Sturges")
> dist
```

| Class limits | f    | rf   | rf(%) | cf    | cf(%) |
|--------------|------|------|-------|-------|-------|
| [16,18)      | 331  | 0.02 | 1.82  | 331   | 1.82  |
| [18,20)      | 1756 | 0.10 | 9.64  | 2087  | 11.46 |
| [20,21)      | 2663 | 0.15 | 14.63 | 4750  | 26.09 |
| [21,23)      | 2672 | 0.15 | 14.68 | 7422  | 40.76 |
| [23,25)      | 2677 | 0.15 | 14.70 | 10099 | 55.47 |
| [25,27)      | 1387 | 0.08 | 7.62  | 11486 | 63.09 |
| [27,29)      | 2263 | 0.12 | 12.43 | 13749 | 75.51 |



|         |      |      |       |       |        |
|---------|------|------|-------|-------|--------|
| [29,31) | 1876 | 0.10 | 10.30 | 15625 | 85.82  |
| [31,32) | 1281 | 0.07 | 7.04  | 16906 | 92.85  |
| [32,34) | 812  | 0.04 | 4.46  | 17718 | 97.31  |
| [34,36) | 323  | 0.02 | 1.77  | 18041 | 99.09  |
| [36,38) | 119  | 0.01 | 0.65  | 18160 | 99.74  |
| [38,40) | 25   | 0.00 | 0.14  | 18185 | 99.88  |
| [40,42) | 18   | 0.00 | 0.10  | 18203 | 99.98  |
| [42,44) | 1    | 0.00 | 0.01  | 18204 | 99.98  |
| [44,45) | 3    | 0.00 | 0.02  | 18207 | 100.00 |

- Frecuencia absoluta -> f
- Frecuencia relativa -> rf
- Frecuencia relativa porcentual -> rf(%)
- Frecuencia abs acumulada -> cf
- Frecuencia rel acumulada porcentual -> cf(%)

Para calcular la **media** de edad hemos construido una función en R la cual es:

```
> source("media.R")
> media

function(var) {
  sum<-0

  for (data in var) {
    sum<-sum+data
  }

  return(sum/length(var))
}
```

Procedemos al cálculo de la media haciendo uso de la misma:

```
> mf<-media(fifa$Age)
> mf

[1] 25.12221
```

En cuanto a los **cuantiles**, para su visualización hemos optado por el uso de un diagrama de bigotes, donde además aparecen representados los límites inferior y superior de la variable de estudio. Para su realización se ha hecho lo mismo que con el histograma de la siguiente manera:

```
> source("bigotes.R")
> bigotes
```

```

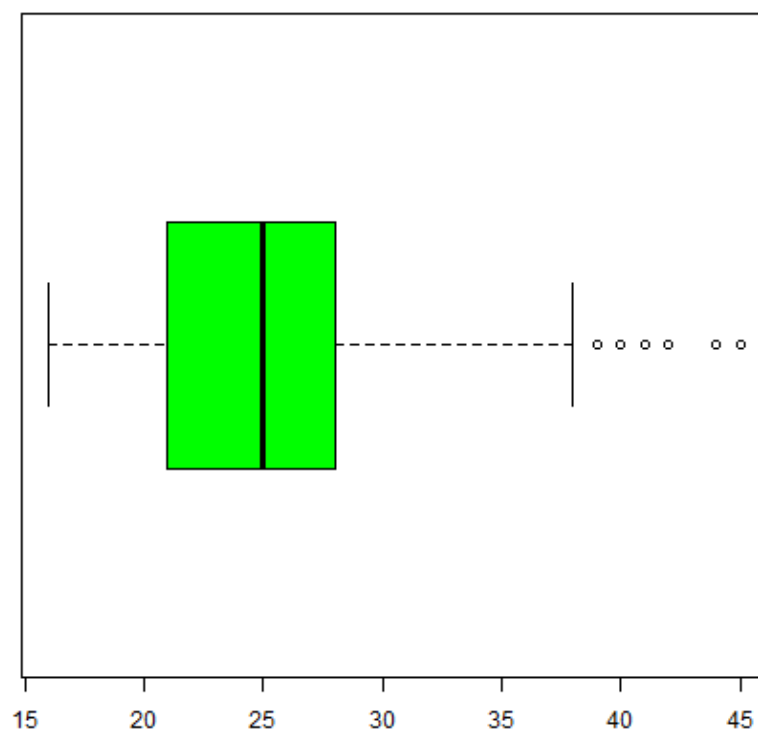
function(var,ruta) {
  png(ruta)

  boxplot(var, col='green', horizontal=T)

  dev.off()
}

> b<-bigotes(edad,"./bigotes.png")

```



Para calcular la **desviación típica** hemos construido una función de la siguiente forma:

```

> source("desviacion.R")
> desviacion

function(var,media) {
  num<-0
  for (data in var){
    num<-num+((data-media)^2)
  }
}

```

```

    s<-sqrt(num/length(var))

    return(s)
}

```

El cálculo de la misma es:

```

> s<-desviacion(edad,mf)
> s

[1] 4.669814

```

De acuerdo al teorema de *Tchebychev* (realizado en una función aparte) el intervalo en el que se encuentran el 75% de los datos, es decir para  $k=2$ , es:

```

> source("tche.R")
> t<-tche(s,mf,2)
> toString(t)

[1] "15, 34"

```

Dicha función es:

```

> tche

function(des,media,k) {

    rango<-des*k
    inter<-list(as.integer(media-rango),as.integer(media+rango))

    return(inter)
}

```

Observando el intervalo tan amplio necesario para englobar el 75% de los datos se puede concluir con que la media no es una buena medida representante del conjunto de datos estudiado.

El cálculo de la **varianza** es sencillo, únicamente es la desviación al cuadrado:

```

> v<-s^2
> v

[1] 21.80717

```