PRÁCTICA 1
BIOESTADÍSTICA

Autor: *Laura Sánchez Garzón*

Grado: *Ingeniería Biomédica*

Fecha: *17 de febrero de 2023*

Bibliografía y referencias:

https://bookdown.org/matiasandina/R-intro/primeros-pasos.html

https://www.uv.es/lejarza/mcaf/materialR/curso-R-xsjv.pdf

https://iqss.github.io/dss-workshops/R/Rintro/base-r-cheat-sheet.pdf

https://cheatography.com/macasalva/cheat-sheets/r-base/pdf/

http://publish.illinois.edu/johnrgallagher/files/2015/10/BaseGraphicsCheatsheet.pdf

ÍNDICE

1. INTRODUCCIÓN

Esta primera práctica de bioestadística introduce al alumno a un nuevo lenguaje de programación, lenguaje R.

Gracias al uso del programa R Studio, se le ha permito al alumno familiarizarse con los comandos y reglas básicas que éste incluye, de manera que ha podido aplicar los conocimientos teóricos estudiados previamente en clase de bioestadística, en un código relativamente sencillo con infinitas utilidades.

El alumno comprobará la rapidez y eficiencia que supone aprender a usar lengua en R, y le permitirá asentar conceptos clave en la estadística como aprender a utilizar y crear cada tipo de gráfico, extraer conclusiones tras ser analizado, y entender cómo funcionan operaciones básicas en la estadística como la media, la mediana, los cuantiles, etc.

Para facilitar al alumno enfrentarse a un lenguaje de programación nuevo, le han sido entregados seis códigos ya creados para ejecutarlos paso a paso, entendiendo qué ocurre en cada momento. Muchos de esos códigos implicaban descargarse la librería ISwR, que contiene datos biomédicos ya cargados, como concentraciones de folatos, de inmunoglobulina en suero, o de IGF-I.

Tras este primer contacto con R Studio, el alumno se ha debido enfrentar a la creación de dos códigos a trabajar sobre una tabla de Excel con una serie de datos que analizar y a graficar.

Dicho archivo csv. se adjuntará junto con esta memoria como pdf., junto a los códigos entregables.

2. CÓDIGOS ENTREGADOS COMENTADOS PASO A PASO

Los códigos adjuntados son códigos propuestos por el profesorado de bioestadística. Este bloque de la memoria va dirigido a la comprensión de cada línea de código ofrecida. En esta memoria se ha querido comentar el código con **#azules**, y se insertan figuras representativas que aclaran cada paso que se da.

2.1. Código 1

Ejemplo de cómo calcular fácilmente estadísticos de tendencia central (media, mediana y moda), de dispersión (varianza, desviación típica, rango), de forma (asimetría, curtosis), de posición (percentiles).

```
#Se guarda este primer código en la carpeta deseada

setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-1")


#Primero generaremos una distribución normal de 50 elementos.

# rnorm(y) crea y valores que siguen una distribución normal (0, 1)

x<-rnorm(50) # se almacenan los valores en un vector que se llama x.
```

```
> x
 [1] -0.59285769 -1.48046355  0.61870100  0.97607604 -0.31536479
 [6]  1.33969062  1.00594655 -0.45454351  0.81267278 -0.39630101
[11] -0.69624888  0.22281043 -0.87075340  2.46878716 -1.22499536
[16] -0.14489253  0.42503505 -0.99334699  0.45618154 -0.54903100
[21] -0.23402040 -1.57103595  1.34954436  0.17668073  0.43109022
[26] -1.33600267  0.80457775 -1.10712949 -0.69591848 -0.52887179
[31]  0.17538634  1.53684659  1.25178193 -0.44824643 -1.16673014
[36]  1.54889621  0.63406996  0.08616510  0.58337176  0.32858513
[41] -0.80987665  0.26647816 -0.35487029 -1.92597591  0.98774890
[46] -1.45513981  0.96723675  0.01889263  0.84113886  1.19157687
```

Figura 1: Vector x que representa una distribución normal de 50 elementos

```
media=mean(x) #almacena la media de x (vector de los 50 valores que siguen
una distribución normal (0,1))

mediana=median(x) #mediana almacena la mediana de x
```

```
> media
[1] 0.04306705
> mediana
[1] 0.1307757
```

Figura 2: Media y mediana de x

```
quantile(x) # Función para obtener los cuantiles empíricos (cuartiles)
```

```
> quantile(x)
        0%        25%        50%        75%       100%
-1.9259759 -0.6701533  0.1307757  0.8106490  2.4687872
```

Figura 3: Cuantiles empíricos de x

```
pvec<-seq(0,1,0.1) #en pvec se guarda la secuencia del 0 al 1, con saltos de 0.1
```

```
> pvec
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

Figura 4: pvec guarda la secuencia de 0 a 1 en saltos de 0.1

```
quantile(x,pvec) #calcular los cuantiles de x en porcentajes de 10 (deciles)
```

```
> quantile(x,pvec)
        0%        10%        20%        30%        40%        50%
-1.9259759 -1.2360961 -0.8220520 -0.5349196 -0.3311670  0.1307757
       60%        70%        80%        90%       100%
 0.3671651  0.6233117  0.9690046  1.2605728  2.4687872
```

Figura 5: Cuantiles de x, calculados con pvec

En este primer código se han visto los conceptos de media, mediana y cuantil.

La media ($\bar{x}$, media aritmética) es una variable cuantitativa que representa el valor que se obtiene al dividir la suma de un conglomerado de números entre la cantidad de ellos.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad \left(\bar{x} = \frac{\sum_{i=1}^{n} x_i f_i}{n}\right)$$

La mediana es una variable cuantitativa que representa el valor central del conjunto ordenado de observaciones (o media de los dos centrales).

Cuantil es una medida de dispersión que indica la variabilidad o dispersión de las observaciones.

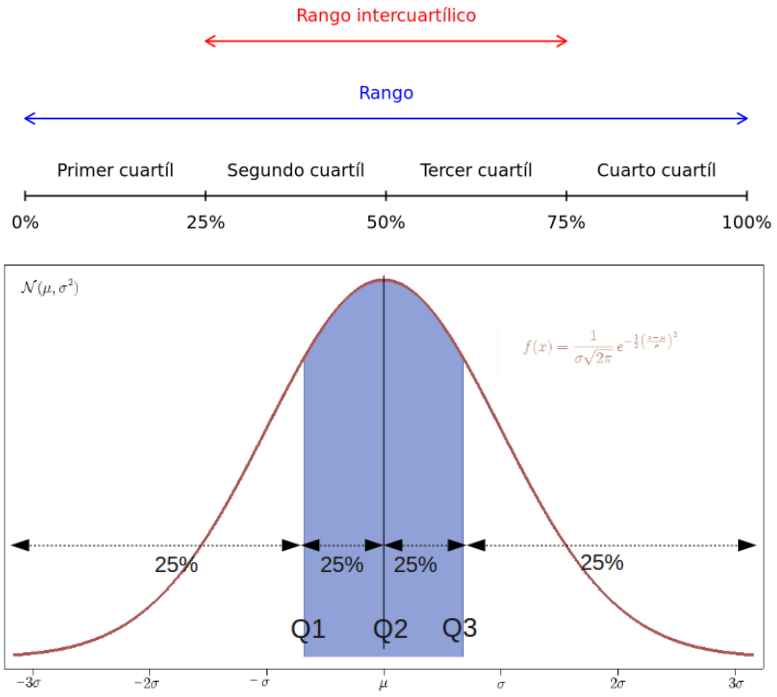

Figura 6: Imagen representativa del concepto de cuantil

2.2. Código 2

Exploración de la base de datos juul, y cálculo de estadísticos que permiten hacer un resumen de los datos (Código 2) (b) En la hoja de datos, denominada por R.

```
install.packages("ISwR") #se descarga la librería ISwR

library(ISwR)


data(juul) #mostrar los datos contenidos en la base de datos de juul
```

```
> force(juul)
     age menarche sex igf1 tanner testvol
1     NA       NA  NA   90     NA       NA
2     NA       NA  NA   88     NA       NA
3     NA       NA  NA  164     NA       NA
4     NA       NA  NA  166     NA       NA
5     NA       NA  NA  131     NA       NA
6   0.17       NA   1  101      1       NA
7   0.17       NA   1   97      1       NA
8   0.17       NA   1  106      1       NA
9   0.17       NA   1  111      1       NA
10  0.17       NA   1   79      1       NA
11  0.17       NA   1   43      1       NA
12  0.17       NA   1   64      1       NA
13  0.25       NA   1   90      1       NA
14  0.25       NA   1  141      1       NA
15  0.42       NA   1   42      1       NA
16  0.50       NA   1   43      1       NA
17  0.67       NA   1  132      1       NA
18  0.75       NA   1   43      1       NA
19  0.75       NA   1   36      1       NA
20  1.00       NA   1   86      1       NA
21  1.16       NA   1   44      1       NA
22  1.50       NA   1   68      1       NA
23  1.50       NA   1   89      1       NA
24  1.58       NA   1  101      1       NA
25  1.67       NA   1  115      1       NA
26  1.67       NA   1   53      1       NA
27  1.75       NA   1   94      1       NA
28  1.83       NA   1   95      1       NA
29  1.92       NA   1   76      1       NA
30  2.00       NA   1   79      1       NA
31  2.00       NA   1   71      1       NA
32  2.20       NA   1  121      1       NA
33  2.41       NA   1  201      1       NA
34  2.42       NA   1   96      1       NA
35  2.42       NA   1   29      1       NA
36  2.83       NA   1   80      1       NA
37  3.00       NA   1  117      1       NA
38  3.08       NA   1   38      1       NA
39  3.08       NA   1  100      1       NA
40  3.16       NA   1  108      1       NA
41  3.16       NA   1   52      1       NA
42  4.08       NA   1  106      1       NA
43  4.16       NA   1  182      1       NA
44  4.66       NA   1  195      1       NA

. . .
```

Figura 7: Tabla de datos contenidos en juul

```
?juul #help de juul
```

Description

The juul data frame has 1339 rows and 6 columns. It contains a reference sample of the distribution of insulin-like growth factor (IGF-I), one observation per subject in various ages, with the bulk of the data collected in connection with school physical examinations.

Usage

```
juul
```

Format

This data frame contains the following columns:

`age` a numeric vector (years).

`menarche` a numeric vector. Has menarche occurred (code 1: no, 2: yes)?

`sex` a numeric vector (1: boy, 2: girl).

`igf1` a numeric vector, insulin-like growth factor ($\mu g/l$).

`tanner` a numeric vector, codes 1–5: Stages of puberty ad modum Tanner.

`testvol` a numeric vector, testicular volume (ml).

Source

Original data.

Examples

```
plot(igf1~age, data=juul)
```

Figura 8: Descripción y características de juul

```
plot(igf1~age, data=juul) #se le da la instrucción de que muestre un gráfico
de puntos que relaciona igf1 con edad, a partir de la tabla de datos de juul
```

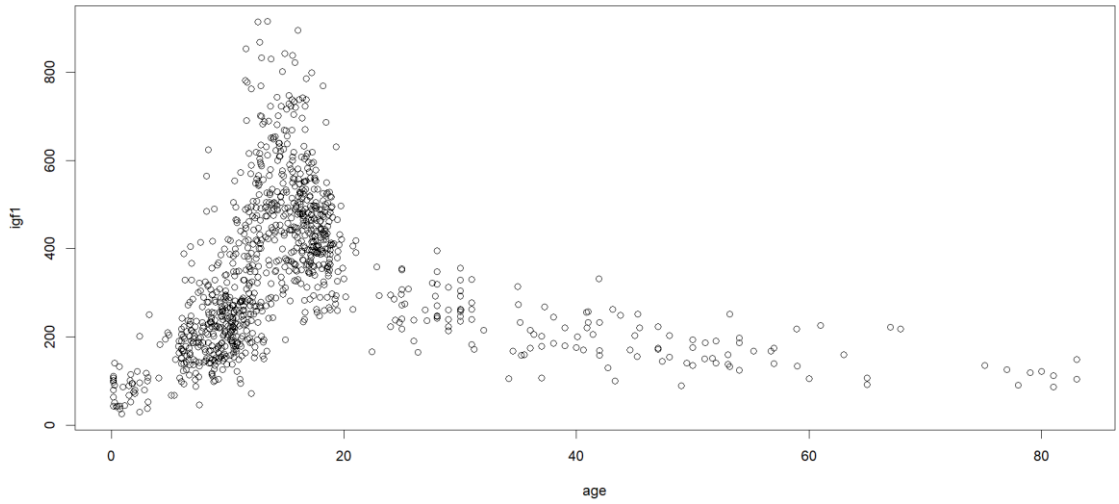

Figura 9: Gráfico de puntos que relaciona igf1 con edad

```
attach(juul) # attach() permite acceder fácilmente a las "columnas" de
un data frame. De modo que, en vez de escribir data.frame$columna, podemos
usar simplemente el nombre de la columna.
```

```
mean(igf1) #va a dar error puesto que hay valores perdidos (NA)
```

```
> mean(igf1)
[1] NA
```

Figura 10: Error porque hay valores NA

```
mean(igf1,na.rm=T) #debemos indicar que no tenga en cuenta valores perdidos
```

```
> mean(igf1,na.rm=T)
[1] 340.168
```

Figura 11: Media de igf1

```
sd(igf1,na.rm=T) #calcula la desviación estándar
```

```
> sd(igf1,na.rm=T)
[1] 171.0356
```

Figura 12: Desviación estándar de igf1

```
sum(!is.na(igf1)) #determina el número de valores no perdidos (número de NA)
```

```
> sum(!is.na(igf1))
[1] 1018
```

Figura 13: Número de valores no NA

```
summary(juul) #resumen del dataset juul
```

```
> summary(juul)
      age            menarche          sex             igf1           tanner         testvol      
 Min.   : 0.170   Min.   :1.000   Min.   :1.000   Min.   : 25.0   Min.   :1.00   Min.   : 1.000  
 1st Qu.: 9.053   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:202.2   1st Qu.:1.00   1st Qu.: 1.000  
 Median :12.560   Median :1.000   Median :2.000   Median :313.5   Median :2.00   Median : 3.000  
 Mean   :15.095   Mean   :1.476   Mean   :1.534   Mean   :340.2   Mean   :2.64   Mean   : 7.896  
 3rd Qu.:16.855   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:462.8   3rd Qu.:5.00   3rd Qu.:15.000  
 Max.   :83.000   Max.   :2.000   Max.   :2.000   Max.   :915.0   Max.   :5.00   Max.   :30.000  
 NA's   :5        NA's   :635     NA's   :5       NA's   :321     NA's   :240    NA's   :859     
```

Figura 14: resumen del dataset juul

En este segundo código surgen conceptos nuevos como desviación típica y gráfico de puntos.

La desviación típica (raíz cuadrada de la varianza) es una medida que se utiliza para cuantificar la variación o la dispersión de un conjunto de datos numéricos, es decir, se trata de la raíz cuadrada de la media de los cuadrados de las puntuaciones de desviación.

Por el otro lado, el gráfico de puntos permite hacerse una idea de cómo se distribuyen y concentran las frecuencias de cierta variable.

2.3. Código 3

Etiquetar los resúmenes de datos.

```
detach(juul) #usado para "desenganchar" un paquete que estaba enganchado a una librería

summary(juul) #recordemos cómo estaba resumido el juul en el código 2
```

```
> summary(juul)
      age           menarche          sex            igf1           tanner          testvol
 Min.   : 0.170   Min.   :1.000   Min.   :1.000   Min.   : 25.0   Min.   :1.00   Min.   : 1.000
 1st Qu.: 9.053   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:202.2   1st Qu.:1.00   1st Qu.: 1.000
 Median :12.560   Median :1.000   Median :2.000   Median :313.5   Median :2.00   Median : 3.000
 Mean   :15.095   Mean   :1.476   Mean   :1.534   Mean   :340.2   Mean   :2.64   Mean   : 7.896
 3rd Qu.:16.855   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:462.8   3rd Qu.:5.00   3rd Qu.:15.000
 Max.   :83.000   Max.   :2.000   Max.   :2.000   Max.   :915.0   Max.   :5.00   Max.   :30.000
 NA's   :5        NA's   :635     NA's   :5       NA's   :321     NA's   :240    NA's   :859
```

Figura 15: Resumen del dataset de juul

```
# factor estructura de datos para manejar variables categóricas

juul$sex<-factor(juul$sex,labels=c("M","F")) #en la columna sex se van a renombrar las filas como M y F
```

```
> juul$sex<-factor(juul$sex,labels=c("M","F"))
> summary(juul)
      age           menarche       sex            igf1           tanner          testvol
 Min.   : 0.170   Min.   :1.000   M   :621   Min.   : 25.0   Min.   :1.00   Min.   : 1.000
 1st Qu.: 9.053   1st Qu.:1.000   F   :713   1st Qu.:202.2   1st Qu.:1.00   1st Qu.: 1.000
 Median :12.560   Median :1.000   NA's:  5   Median :313.5   Median :2.00   Median : 3.000
 Mean   :15.095   Mean   :1.476              Mean   :340.2   Mean   :2.64   Mean   : 7.896
 3rd Qu.:16.855   3rd Qu.:2.000              3rd Qu.:462.8   3rd Qu.:5.00   3rd Qu.:15.000
 Max.   :83.000   Max.   :2.000              Max.   :915.0   Max.   :5.00   Max.   :30.000
 NA's   :5        NA's   :635                NA's   :321     NA's   :240    NA's   :859
```

Figura 16: Fijarse en que sex ha cambiado el nombre de sus filas

```
juul$menarche<-factor(juul$menarche,labels=c("No","Yes")) #en la columna menarche se van a renombrar las filas como No y Yes
```

```
> juul$menarche<-factor(juul$menarche,labels=c("No","Yes"))
> summary(juul)
      age           menarche      sex            igf1           tanner          testvol
 Min.   : 0.170   No  :369   M   :621   Min.   : 25.0   Min.   :1.00   Min.   : 1.000
 1st Qu.: 9.053   Yes :335   F   :713   1st Qu.:202.2   1st Qu.:1.00   1st Qu.: 1.000
 Median :12.560   NA's:635   NA's:  5   Median :313.5   Median :2.00   Median : 3.000
 Mean   :15.095                         Mean   :340.2   Mean   :2.64   Mean   : 7.896
 3rd Qu.:16.855                         3rd Qu.:462.8   3rd Qu.:5.00   3rd Qu.:15.000
 Max.   :83.000                         Max.   :915.0   Max.   :5.00   Max.   :30.000
 NA's   :5                              NA's   :321     NA's   :240    NA's   :859
```

Figura 16: Fijarse en que menarche ha cambiado el nombre de sus filas

```
juul$tanner<-factor(juul$tanner,labels=c("I","II","III","IV","V")) #en la columna tanner se van a renombrar las filas como I, II, III, IV, V
```

```
> juul$tanner<-factor(juul$tanner,labels=c("I","II","III","IV","V"))
> summary(juul)
      age          menarche     sex           igf1         tanner       testvol
 Min.   : 0.170   No  :369   M   :621   Min.   : 25.0   I   :515   Min.   : 1.000
 1st Qu.: 9.053   Yes :335   F   :713   1st Qu.:202.2   II  :103   1st Qu.: 1.000
 Median :12.560   NA's:635   NA's:  5   Median :313.5   III : 72   Median : 3.000
 Mean   :15.095                         Mean   :340.2   IV  : 81   Mean   : 7.896
 3rd Qu.:16.855                         3rd Qu.:462.8   V   :328   3rd Qu.:15.000
 Max.   :83.000                         Max.   :915.0   NA's:240   Max.   :30.000
 NA's   :5                              NA's   :321                NA's   :859
```

Figura 17: Fijarse en que menarche ha cambiado el nombre de sus filas

```
attach(juul) #se va a poder acceder a las columnas de juul simplemente dando su nombre

summary(juul)
```

```
> summary(juul)
      age          menarche     sex           igf1         tanner       testvol
 Min.   : 0.170   No  :369   M   :621   Min.   : 25.0   I   :515   Min.   : 1.000
 1st Qu.: 9.053   Yes :335   F   :713   1st Qu.:202.2   II  :103   1st Qu.: 1.000
 Median :12.560   NA's:635   NA's:  5   Median :313.5   III : 72   Median : 3.000
 Mean   :15.095                         Mean   :340.2   IV  : 81   Mean   : 7.896
 3rd Qu.:16.855                         3rd Qu.:462.8   V   :328   3rd Qu.:15.000
 Max.   :83.000                         Max.   :915.0   NA's:240   Max.   :30.000
 NA's   :5                              NA's   :321                NA's   :859
```

Figura 18: Resumen del dataset de juul

```
#También podríamos haber utilizado la función transform(), que permite escribirlo todo en la misma línea de código

juul<-transform(juul,
                sex=factor(sex,labels=c("M","F")),
                menarche=factor(menarche,labels=c("No","Yes")),
                tanner=factor(tanner,labels=c("I","II","III","IV","V")) )

summary(juul)
```

```
> summary(juul)
      age          menarche     sex           igf1         tanner       testvol
 Min.   : 0.170   No  :369   M   :621   Min.   : 25.0   I   :515   Min.   : 1.000
 1st Qu.: 9.053   Yes :335   F   :713   1st Qu.:202.2   II  :103   1st Qu.: 1.000
 Median :12.560   NA's:635   NA's:  5   Median :313.5   III : 72   Median : 3.000
 Mean   :15.095                         Mean   :340.2   IV  : 81   Mean   : 7.896
 3rd Qu.:16.855                         3rd Qu.:462.8   V   :328   3rd Qu.:15.000
 Max.   :83.000                         Max.   :915.0   NA's:240   Max.   :30.000
 NA's   :5                              NA's   :321                NA's   :859
```

Figura 19: Resumen del dataset de juul

2.4. Código 4

Aprender a representar gráficamente los distintos tipos de variables, con el objetivo de obtener una impresión razonable de la forma de la distribución.

```
hist(x) #Histogramas. Por defecto R, intenta hacer puntos de corte
"adecuados"
```

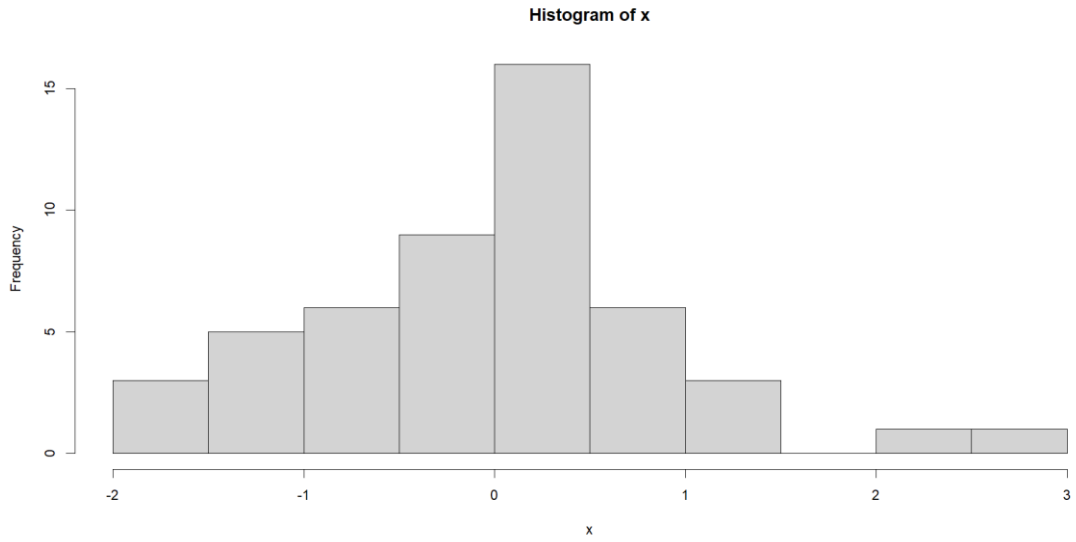

Figura 20: Histograma de x

```
#Ejemplo accidentes versus edad (0-4,5-9,10-15,16,17,18-19,20-24,25-59,60-79)

#se guardan valores de la edad media, el número de accidentes, y la
repetición de cada edad por accidente, en vectores diferentes

mid.age<-c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)

acc.count<-c(28,46,58,20,31,64,149,316,103)

age.acc<-rep(mid.age,acc.count)
```

```
> age.acc
  [1]  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5
 [13]  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5  2.5
 [25]  2.5  2.5  2.5  2.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5
 [37]  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5
 [49]  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5
 [61]  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5  7.5
 [73]  7.5  7.5 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
 [85] 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
 [97] 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
[109] 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
[121] 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
[133] 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5
[145] 16.5 16.5 16.5 16.5 16.5 16.5 16.5 16.5 17.5 17.5 17.5 17.5
[157] 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5
[169] 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5
[181] 17.5 17.5 17.5 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0
[193] 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0
[205] 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0
[217] 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0
[229] 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0 19.0
[241] 19.0 19.0 19.0 19.0 19.0 19.0 19.0 22.5 22.5 22.5 22.5 22.5
[253] 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5
[265] 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5
[277] 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5 22.5
```

Figura 21: age.acc es el vector del número de repeticiones de cada mid.age (media de edad), según el acc.account (número de accidentes)

```
brk<-c(0,5,10,16,17,18,20,25,60,80) #va a ser las divisiones del eje x al
crear el histograma
```

```
hist(age.acc,breaks=brk) #pueden apreciarse las divisiones (breaks)

#Nótese que automáticamente se obtiene de esta manera el histograma correcto
donde el área de una columna es proporcional a la frecuencia relativa de
manera que el área total del histograma es 1.
```

Figura 22: Histograma en el que se estima la densidad con la que han ocurrido accidentes con respecto al vector que aúna edad y número de accidentes

```
#Distribución empírica acumulada

n<-length(x)

plot(sort(x),(1:n)/n, type="s",ylim=c(0,1))
```

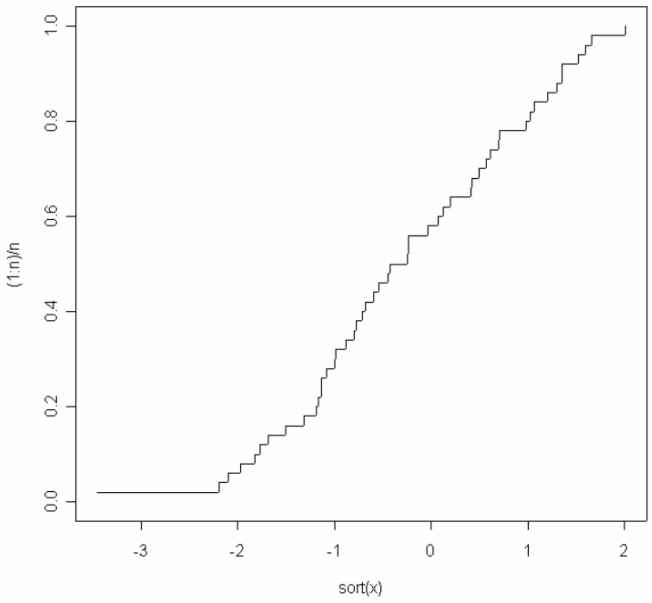

Figura 23: Distribución empírica acumulada

```
#Qqplot

qqnorm(x) # qqnormes una función genérica cuyo método predeterminado produce
un gráfico QQ normal de los valores en y. qqlineagrega una línea a un gráfico
cuantil-cuantil "teórico", por defecto normal, que pasa a través de
los probs cuantiles, por defecto el primer y tercer cuartiles.
```

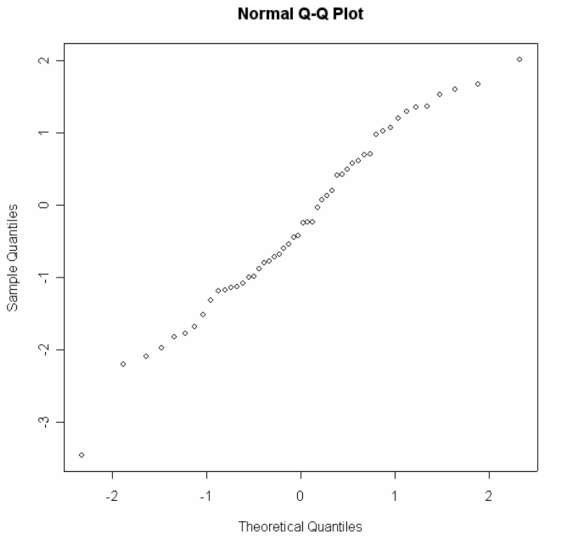

Figura 24: Gráfico de puntos

```
#Boxplots IgM (Concentraciones de IgM en suero de 298 niños de 6 meses-6 años
de edad

data(IgM) #se cargan los datos de IgM guardados en la dataset de la librería
ISwR

?IgM

#par(mfrow=c(2,1)) dibuja una matriz de gráficos 2x1: un gráfico debajo de
otro

par(mfrow=c(1,2)) #con mfrow los gráficos se organizarán por filas

boxplot(IgM) #boxplot sirve para crear un gráfico de cajas y bigotes
```

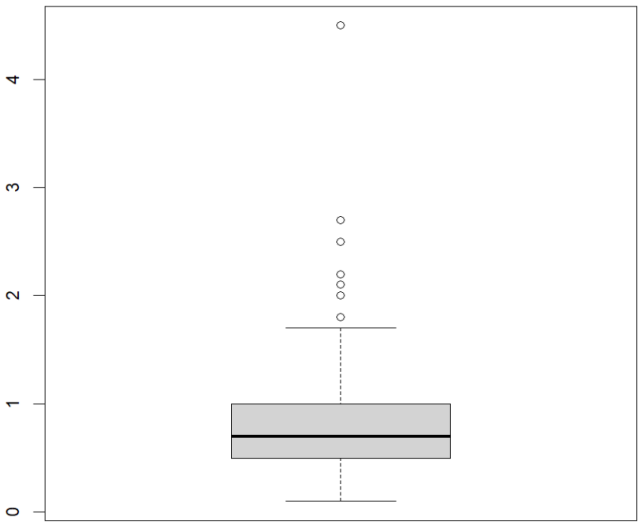

Figura 25: Gráfico de cajas y bigotes de la Inmunoglobulina M

```
boxplot(log(IgM)) #se representa el mismo grafico tras haberle aplicado
logaritmos
```

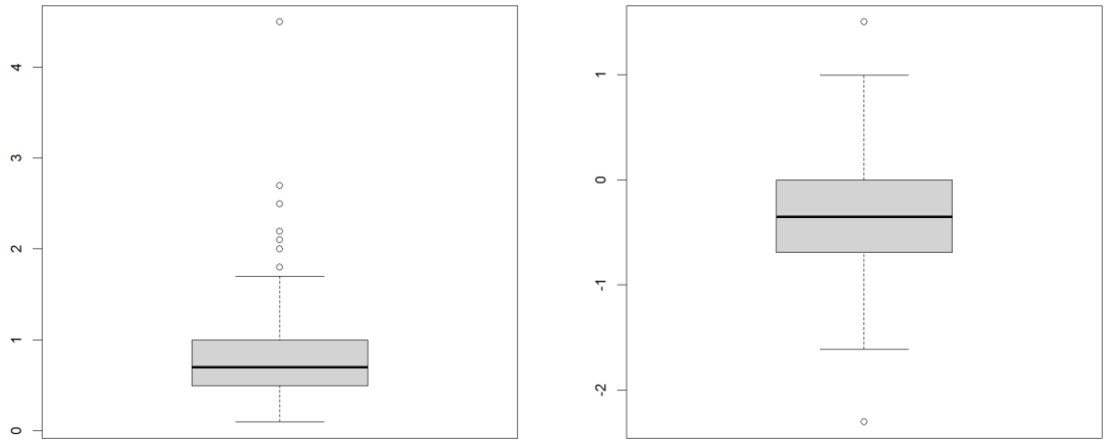

Figura 26: Gráfico de caja y bigotes de la IgM en versión normal y logarítmica

```
par(mfrow=c(2,1)) #se pretende graficar la misma información, pero mostrando
un gráfico encima del otro

boxplot(IgM)

boxplot(log(IgM))
```

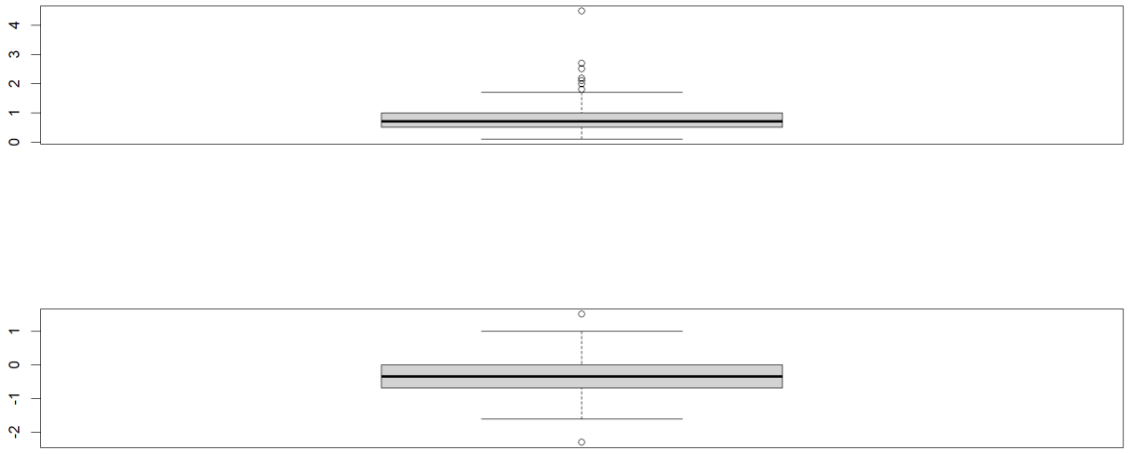

Figura 27: Gráfico de caja y bigotes de la IgM en versión normal y logarítmica

Los diagramas de Caja-Bigotes (boxplots) son 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.

2.5. Código 5

Ejemplos de cómo generar estadísticos y gráficos descriptivos por grupos

```
#Concentraciones de folatos en células sanguíneas en relación a

#tres tipos de ventilación durante la anestesia

install.packages("ISwR")

library(ISwR)

data(red.cell.folate) #se cargan los datos de las concentraciones de folatos
en células sanguíneas, guardadas en la librería ISwR

attach(red.cell.folate)

?red.cell.folate

summary(red.cell.folate)
```

```
> summary(red.cell.folate)
     folate          ventilation
 Min.   :206.0   N2O+O2,24h:8
 1st Qu.:249.5   N2O+O2,op :9
 Median :274.0   O2,24h    :5
 Mean   :283.2
 3rd Qu.:305.5
 Max.   :392.0
```

Figura 28: Resumen del dataset de red.cell.folate

```
tapply(folate,ventilation,mean) #media de los datos de la columna ventilación
```

```
> tapply(folate,ventilation,mean)
N2O+O2,24h  N2O+O2,op     O2,24h
  316.6250   256.4444   278.0000
```

Figura 29: Media de los datos de la columna ventilación

```
#Para tener más de un estadístico resumen por grupo

m<-tapply(folate,ventilation,mean) #media

s<-tapply(folate,ventilation,sd) #desviación típica

n<-tapply(folate,ventilation,length) #longitud de las columnas
```

```
> m
N2O+O2,24h  N2O+O2,op     O2,24h
  316.6250   256.4444   278.0000
> s
N2O+O2,24h  N2O+O2,op     O2,24h
  58.71709   37.12180   33.75648
> n
N2O+O2,24h  N2O+O2,op     O2,24h
         8          9          5
```

Figura 30: Media, Desviación típica y longitud de columnas por separado

```
cbind(mean=m,std.dev=s,n=n) #se aúnan todos los datos
```

```
> cbind(mean=m,std.dev=s,n=n)
                mean  std.dev n
N2O+O2,24h 316.6250 58.71709 8
N2O+O2,op  256.4444 37.12180 9
O2,24h     278.0000 33.75648 5
```

Figura 31: Matriz con todos los datos

```
data(juul) #para el dataset juul

plot(igf1~age, data=juul)

tapply(igf1,tanner,mean)
```

```
> tapply(igf1,tanner,mean)
 1  2  3  4  5
NA NA NA NA NA
```

Figura 32: Dado que hay datos con missing values, al intentar calcular la media, sale NA

```
tapply(igf1,tanner,mean,na.rm=T)
```

```
> tapply(igf1,tanner,mean,na.rm=T)
       1        2        3        4        5
207.4727 352.6714 483.2222 513.0172 465.3344
```

Figura 33: Quitando los missing values, obtenemos las respectivas medias de cada tanner con respecto al valor de igf1

```
data(energy) #Cargamos la base de datos energy

attach(energy)

summary(energy)
```

```
> summary(energy)
     expend        stature
 Min.   : 6.130   lean :13
 1st Qu.: 7.660   obese: 9
 Median : 8.595
 Mean   : 8.979
 3rd Qu.: 9.900
 Max.   :12.790
```

Figura 34: Resumen del dataset de energía

```
?energy
```

```
# Histogramas para cada grupo de mujeres

expend.lean<-expend[stature=="lean"] #se buscan aquellas mujeres calificadas como “lean” (delgadas).
```

```
> expend.lean
 [1]  7.53  7.48  8.08  8.09 10.15  8.40 10.88  6.13  7.90  7.05
[11]  7.48  7.58  8.11
```

Figura 35: Vector de valores en los que las mujeres quedan calificadas como “lean”

```
expend.obese<-expend[stature=="obese"]
```

```
> expend.obese
[1]  9.21 11.51 12.79 11.85  9.97  8.79  9.69  9.68  9.19
```

Figura 36: Vector de valores en los que las mujeres quedan calificadas como “obese”

```
par(mfrow=c(2,1))

hist(expend.lean,breaks=10,xlim=c(5,13),ylim=c(0,4),col="white") #xlim e ylim sirven para indicar desde qué valor hasta qué valor graficar.

#col sirve para indicar de qué color rellenar el gráfico

hist(expend.obese,breaks=10,xlim=c(5,13),ylim=c(0,4),col="black")
```

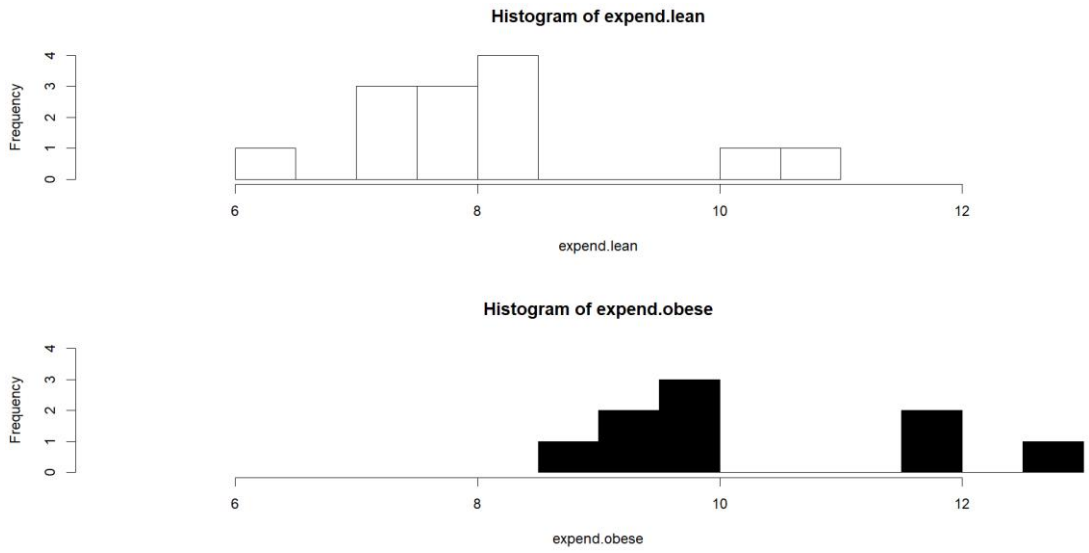

Figura 37: Histogramas que representan la frecuencia absoluta de mujeres que sufren delgadez y obesidad

```
#Boxplots para cada grupo

par(mfrow=c(1,1))

boxplot(expend~stature) #relacionar expend con estatura. Al poner el nombre de las columnas, sale el gráfico etiquetado.
```

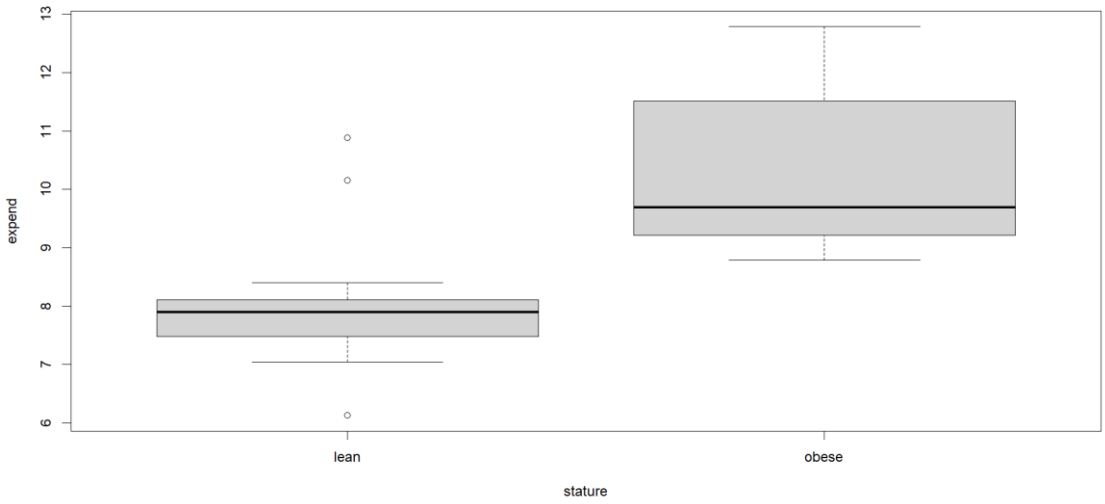

Figura 38: Gráfico de cajas y bigotes que representa stature con relación a expend

```
boxplot(expend.lean,expend.obese) #el gráfico no va a salir etiquetado por haber utilizado directamente el nombre de las variables
```

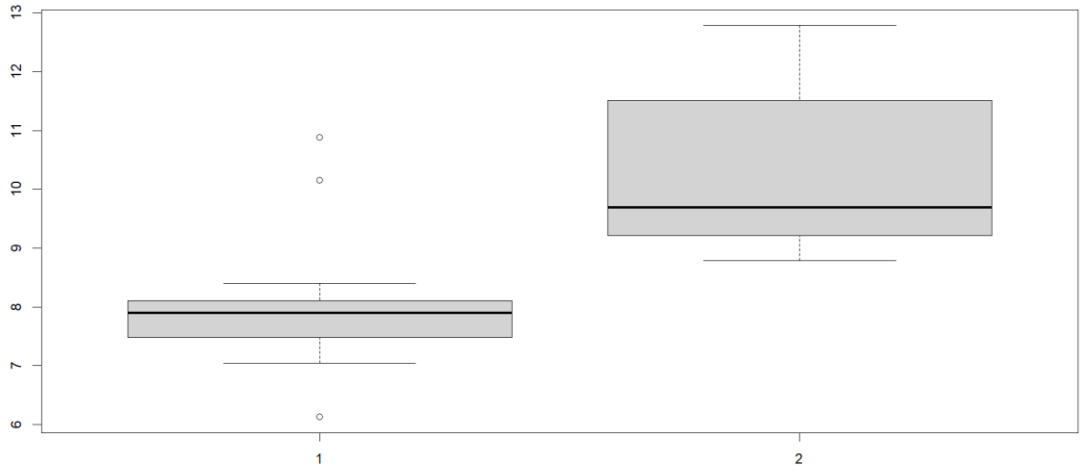

Figura 39: Gráfico de cajas y bigotes que representa stature con relación a expend

```
#Con muestras tan pequeñas, los boxplots pueden resultar engañosos

#Se puede realizar gráficos de los datos originales, punto a punto

opar<-par(mfrow=c(2,2),mex=0.8,mar=c(3,3,2,1)+0.1) #mex y mar son cuestiones de diseño de márgenes
```

```
> opar
$mfrow
[1] 1 1

$mex
[1] 1

$mar
[1] 5.1 4.1 4.1 2.1
```

Figura 40: Qué guarda opar

```
stripchart(expend~stature) # stripchart() coge un vector numérico y dibuja un gráfico de tiras sobre el mismo

stripchart(expend~stature,method="jitter") # La función jitter añade ruido a un vector numérico

stripchart(expend~stature,method="stack") # produce un data frame con dos columnas

stripchart(expend~stature,method="stack",jitter=0.03)
```

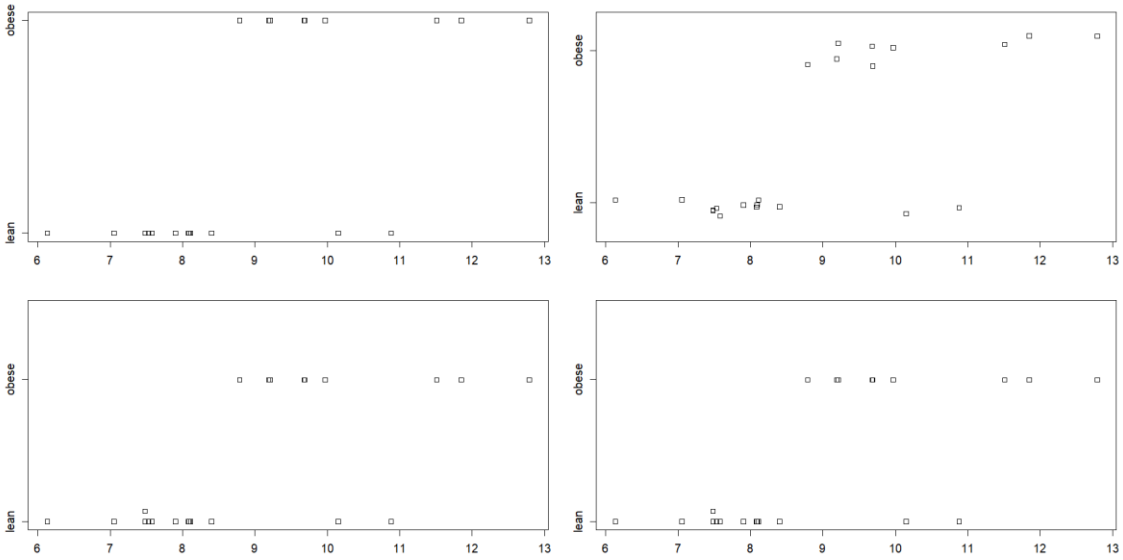

Figura 41: maneras de representar el gráfico de tiras

2.6. Código 6

Aprender a obtener información descriptiva para la elaboración de tablas

```
#Una tabla debe estar en un objecto tipo matriz

#Ejemplo: consumo de cafeína en mujeres según estado civil

caff.marital<-
matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67),nrow=3,byrow=T)

caff.marital
```

```
> caff.marital
     [,1] [,2] [,3] [,4]
[1,]  652 1537  598  242
[2,]   36   46   38   21
[3,]  218  327  106   67
```

Figura 42: Matriz que representa qué hay guardado en caff.marital

```
colnames(caff.marital)<-c("0","1-150","151-300",">300") #se renombran las filas y
columnas

rownames(caff.marital)<-c("Married","Prev.married","Single")

caff.marital
```

```
> colnames(caff.marital)<-c("0","1-150","151-300",">300")
> rownames(caff.marital)<-c("Married","Prev.married","Single")
> caff.marital
              0 1-150 151-300 >300
Married     652  1537     598  242
Prev.married  36    46      38   21
Single      218   327     106   67
```

Figura 43: Matriz que representa qué hay guardado en caff.marital

```
install.packages("ISwR")

library(ISwR)

detach(juul)

attach(juul)

juul<-transform(juul,

                sex=factor(sex,labels=c("M","F")),

                menarche=factor(menarche,labels=c("No","Yes")),

                tanner=factor(tanner,labels=c("I","II","III","IV","V")) )
```

```
#También podemos crearla a partir de variables categóricas de un dataset
table(sex)
table(sex,menarche)
table(menarche,tanner)
```

```
> table(sex)
sex
  M   F
621 713
> table(sex,menarche)
   menarche
sex  No Yes
  M   0   0
  F 369 335
> table(menarche,tanner)
        tanner
menarche   I  II III  IV   V
     No  221  43  32  14   2
     Yes   1   1   5  26 202
```

Figura 44: Matrices que representan sex, combinación de sex y menarche, y combinación de menarche y tanner

```
t(caff.marital) #Podemos transponer las tablas
```

```
> t(caff.marital)
        Married Prev.married Single
0           652           36    218
1-150      1537           46    327
151-300     598           38    106
>300        242           21     67
```

Figura 45: Matriz traspuesta de caff.marital

```
#Para calcular las frecuencias marginales, por fila o columna
tanner.sex<-table(tanner,sex)
margin.table(tanner.sex,1)
```

```
> tanner.sex<-table(tanner,sex)
> margin.table(tanner.sex,1)
tanner
  I  II III  IV   V
515 103  72  81 328
```

Figura 46: Tanner según sex

```
margin.table(tanner.sex,2)
```

```
> margin.table(tanner.sex,2)
sex
  M   F
545 554
```

Figura 47: Tabla marginal de sex

```
prop.table(tanner.sex,1) #combinar sex y tanner

prop.table(tanner.sex,1)*100 #multiplicar por 100 dicha combinación
```

```
> prop.table(tanner.sex,1)
      sex
tanner         M         F
   I   0.5650485 0.4349515
   II  0.5339806 0.4660194
   III 0.4722222 0.5277778
   IV  0.5061728 0.4938272
   V   0.3780488 0.6219512
> prop.table(tanner.sex,1)*100
      sex
tanner        M        F
   I   56.50485 43.49515
   II  53.39806 46.60194
   III 47.22222 52.77778
   IV  50.61728 49.38272
   V   37.80488 62.19512
```

Figura 48: Matriz que combina tanner y sex

```
tanner.sex/sum(tanner.sex)
```

```
> tanner.sex/sum(tanner.sex)
      sex
tanner          M          F
   I   0.26478617 0.20382166
   II  0.05004550 0.04367607
   III 0.03093722 0.03457689
   IV  0.03730664 0.03639672
   V   0.11282985 0.18562329
```

Figura 49: Matriz que combina sex y tanner, y divide las frecuencias entre el sumatorio total

```
#También se pueden representar gráficamente tablas como por ejemplo con el
diagrama de barras

total.caff<-margin.table(caff.marital,2)

total.caff
```

```
> total.caff
   0  1-150 151-300  >300
 906  1910    742   330
```

Figura 50: Total de cafeína

```
barplot(total.caff,col="white")
```

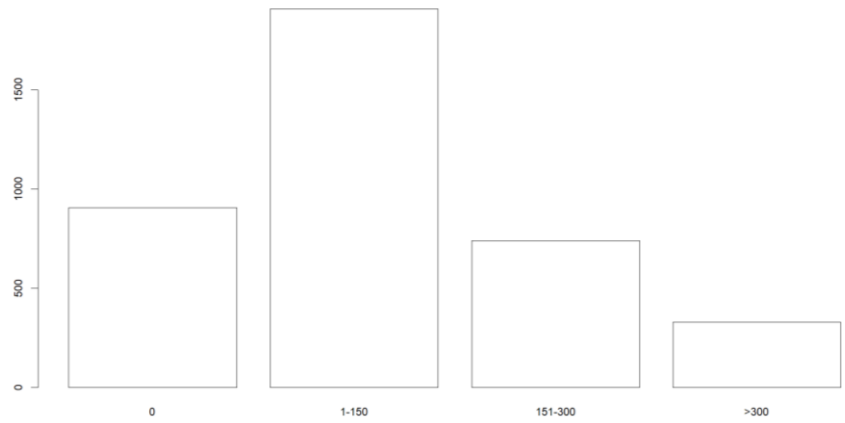

Figura 51: Gráfico de barras que representa el total de café por intervalos

```
#Diagramas de barras para una tabla de contingencia
par(mfrow=c(2,2))
barplot(caff.marital,col="white")
barplot(t(caff.marital),col="white")
barplot(t(caff.marital),col="white",beside=T)
barplot(prop.table(t(caff.marital),2),col="white",beside=T)
```

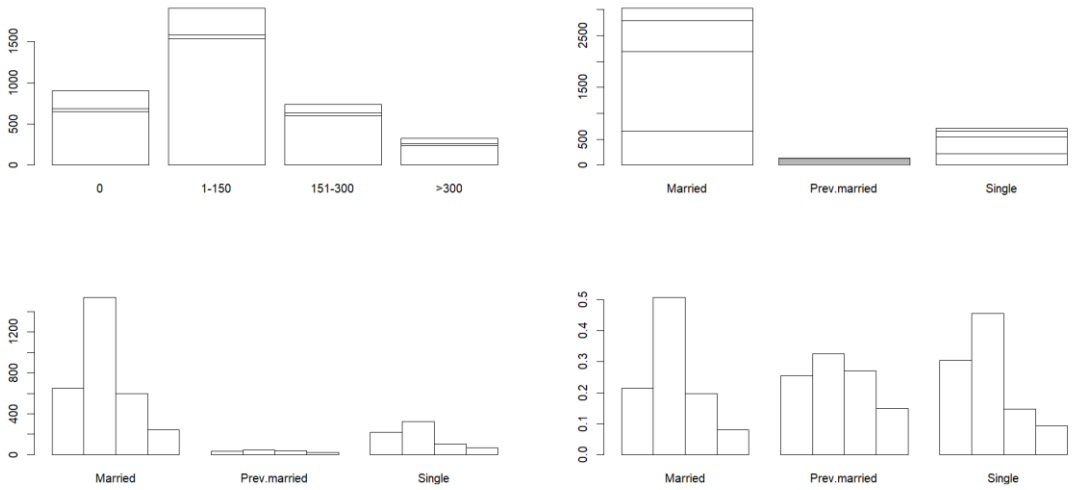

Figura 52: Diferentes maneras de representar el gráfico de barras

```
par(mfrow=c(1,1))
#Otro diagrama de barras para una tabla de contingencia
barplot(prop.table(t(caff.marital),2),beside=T,
legend.text=colnames(caff.marital), #se inserta la leyenda
        col=c("white","grey80","grey50","black"))
```

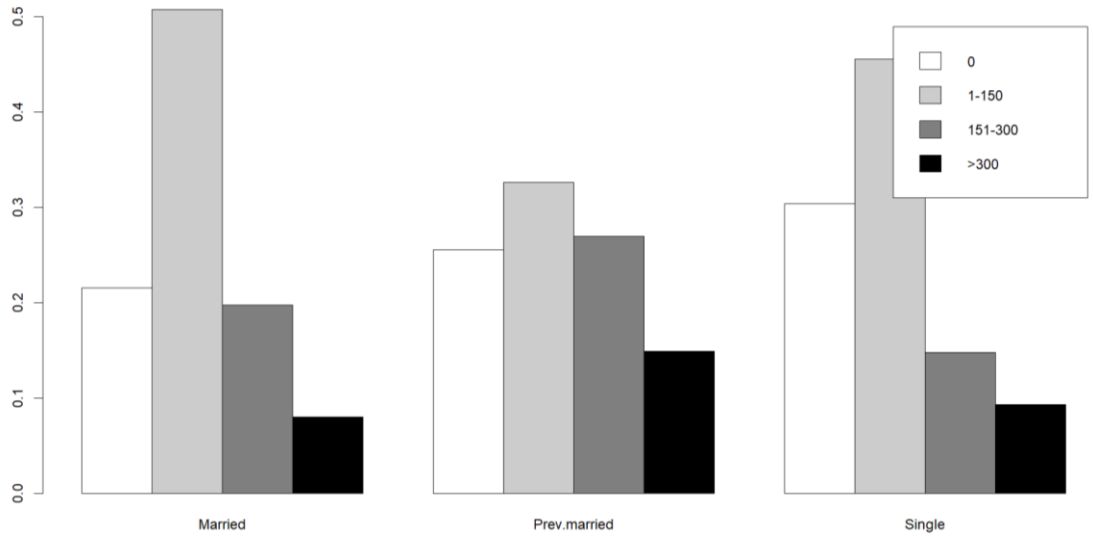

Figura 53: Gráfico de barras utilizando diferentes tonos de la escala de grises

```
#Diagrama de sectores para una tabla de contingencia
opar<-par(mfrow=c(2,2),mex=0.8,mar=c(1,1,2,1))
slices<-c("white","grey80","grey50","black")
pie(caff.marital["Married",],main="Married",col=slices)
pie(caff.marital["Prev.married",],main="Previouslymarried",col=slices)
pie(caff.marital["Single",],main="Single",col=slices)
par(opar)
```

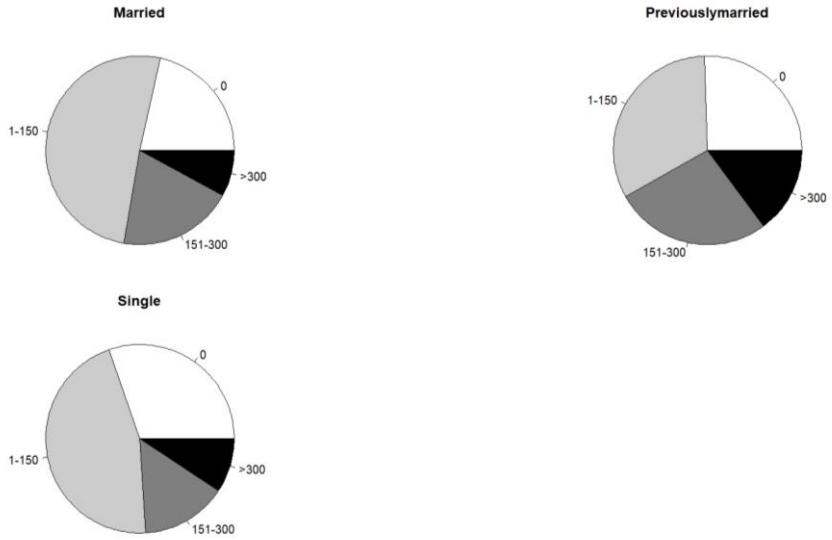

Figura 54: Gráfico de sectores de cada variable cualitativa (Married, Prev.Married, Single)

3. EJERCICIOS PROPUESTOS

En esta sección de la memoria, se va a describir el proceso de la creación de los códigos propios, utilizados para solucionar una serie de problemas bioestadísticos.

3.1. Ejercicio 1

Sobre la base de datos (Datos.csv) adjuntada, se ha trabajado sobre ella para dividir los datos recogidos en columnas y filas. Ésta contenía registros de 200 hombres y mujeres de España, de los cuales se les ha recogido información de variables sociodemográficas (entorno de residencia, sexo, estado civil, nivel de educación y tiempo de educación) y valores de presión arterial (tanto sistólica como diastólica), peso y altura.

El ejercicio consiste en dos partes. Una primera, requiere realizar un código que permita rellenar las siguientes tablas:

	Media	Desv.estándar	Mediana	P25	P75	Missing
Edad						
Peso						
Presión arterial sistólica						
Presión arterial diastólica						

	Frecuencia (n)	%
Sexo		
1		
2		
Estado civil		
Nunca casado		
Actualmente casado		
Viviendo en pareja		
Separado/divorciado		
Viudo		

Figura 55: Tablas propuestas a rellenar utilizando el fichero csv.

La segunda parte consiste en analizar un gráfico que relacione una variable continua (ya sea presión arterial, peso o edad), con variables sociodemográficas (como serían sexo y estado civil), y a partir de ahí reflexionar sobre los datos obtenidos.

El código creado para resolver la primera parte del ejercicio es el siguiente:

```
getwd()

setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-1")

datos <- read.csv("Datos.csv", sep=";") #se abre el fichero csv.

nombres <- c("id", "entorno.residencia", "sexo", "edad", "estado.civil",
"nivel.educacion", "tiempo.educacion", "sistolica", "diastolica", "altura",
"peso") #creamos los nombres de nuestras columnas

options(max.print=999999) #sin este comando, se leían sólo las primeras 60 filas
del csv.

datos <- read.table(file = "Datos.csv", header = FALSE, sep = ";",skip=1,
col.names = nombres) #se guarda en datos la matriz con la información contenida
en el csv.
```

```
> datos
    id entorno.residencia sexo edad estado.civil nivel.educacion tiempo.educacion  sistolica diastolica altura  peso
1    1                  1    2   85            2               0               NA          NA         NA     NA    NA
2    2                  1    2   93            5               0               NA          NA         NA     NA    NA
3    3                  2    1   83            2               1               NA          NA         NA     NA    NA
4    4                  1    2   71            5               2               NA          NA         NA     NA    NA
5    5                  1    1   92            2               2               NA          NA         NA     NA    NA
6    6                  1    2   79            2               2               NA          NA         NA     NA    NA
7    7                  1    2   62            1               0                0          90         63     NA    NA
8    8                  2    2   81            5               1                7         136         70     NA    NA
9    9                  1    2   67            2               2               10          NA         NA     NA    NA
10  10                  1    2   61            5               3               13         160         77     NA    NA
11  11                  1    2   86            5               2               14         130         70     NA    NA
12  12                  1    2   86            4               1                2         167         92  144.0  57.6
13  13                  1    2   79            2               1                8         143         74  144.2  53.4
14  14                  1    2   78            5               1                0         147         77  144.5  82.0
15  15                  1    2   88            5               2                8         137        100  144.5  81.5

...
```

Figura 56: Extracción de la información del csv. como parte de RStudio

```
#media, se calcula con mean

mean(datos$edad,na.rm=T) #na.rm=T sirve para despreciar los missing values

mean(datos$peso,na.rm=T) #datos$_____ sirve para acceder a una columna concreta

mean(datos$sistolica,na.rm=T)

mean(datos$diastolica,na.rm=T)
```

```
> mean(datos$edad,na.rm=T) #na.rm=T sirve para despreciar los missing values
[1] 61.625
> mean(datos$peso,na.rm=T) #datos$_____ sirve para acceder a una columna concreta
[1] 74.35082
> mean(datos$sistolica,na.rm=T)
[1] 131.1311
> mean(datos$diastolica,na.rm=T)
[1] 79.66667
```

Figura 57: Medias de edad, peso, PAM sistólica y PAM diastólica

```
#desviacion estándar, se calcula con sd

sd(datos$edad,na.rm=T)

sd(datos$peso,na.rm=T)

sd(datos$sistolica,na.rm=T)

sd(datos$diastolica,na.rm=T)
```

```
> sd(datos$edad,na.rm=T)
[1] 16.07502
> sd(datos$peso,na.rm=T)
[1] 14.75374
> sd(datos$sistolica,na.rm=T)
[1] 21.17183
> sd(datos$diastolica,na.rm=T)
[1] 11.5146
```

Figura 58: Desviación estándar de edad, peso, PAM sistólica y PAM diastólica

```
#mediana, se calcula con median
median(datos$edad,na.rm=T)
median(datos$peso,na.rm=T)
median(datos$sistolica,na.rm=T)
median(datos$diastolica,na.rm=T)
```

```
> median(datos$edad,na.rm=T)
[1] 63
> median(datos$peso,na.rm=T)
[1] 73
> median(datos$sistolica,na.rm=T)
[1] 130
> median(datos$diastolica,na.rm=T)
[1] 80
```

Figura 59: Medianas de edad, peso, PAM sistólica y PAM diastólica

```
#cuantiles, se calcula con quantile
quantile(datos$edad,na.rm=T, probs=c(0.25, 0.75))
quantile(datos$peso,na.rm=T, probs=c(0.25, 0.75))
quantile(datos$sistolica,na.rm=T, probs=c(0.25, 0.75))
quantile(datos$diastolica,na.rm=T, probs=c(0.25, 0.75))
```

```
> quantile(datos$edad,na.rm=T, probs=c(0.25, 0.75))
25% 75%
 52  75
> quantile(datos$peso,na.rm=T, probs=c(0.25, 0.75))
 25%  75%
65.0 82.3
> quantile(datos$sistolica,na.rm=T, probs=c(0.25, 0.75))
  25%   75%
119.0 141.5
> quantile(datos$diastolica,na.rm=T, probs=c(0.25, 0.75))
25% 75%
 70  87
```

Figura 60: Cuantiles de25 y 75 de edad, peso, PAM sistólica y PAM diastólica

```
#missing, se calcula sumando los datos NA

sum(is.na(datos$edad))

sum(is.na(datos$peso))

sum(is.na(datos$sistolica))

sum(is.na(datos$diastolica))
```

```
> sum(is.na(datos$edad))
[1] 0
> sum(is.na(datos$peso))
[1] 17
> sum(is.na(datos$sistolica))
[1] 17
> sum(is.na(datos$diastolica))
[1] 17
```

Figura 61: Valores perdidos de edad, peso, PAM sistólica y PAM diastólica

```
#frecuencias

sex.freq<-table(datos$sexo) #table nos crea una tabla cuyas filas son el número de veces que aparece cada valor diferente

estado.freq<-table(datos$estado.civil)

uno<-sex.freq[1] #queremos acceder a las dos columnas por separado que gurda la tabla de frecuencias de sex

dos<-sex.freq[2]
```

```
> sex.freq

  1   2
 92 108
> estado.freq

  1   2   3   4   5
 36 116   6  15  27
```

Figura 62: Se cuentan las filas de cada tipo de variable (de sexo y estado civil)

```
#porcentajes

porc.uno<- (uno/(uno+dos))*100 #el porcentaje de 1 es la frecuencia de 1 entre el total

porc.dos<- (dos/(uno+dos))*100

sex.freq.total<-uno+dos #número total de muestras

sex.porc.total<-porc.uno+porc.dos #porcentaje total (debería devolver "100")

nunc.casad<-estado.freq[1]  #queremos acceder a las dos columnas por separado que gurda la tabla de frecuencias de estado.civil
```

```
> porc.uno
 1
46
> porc.dos
 2
54
> sex.freq.total
  1
200
> sex.porc.total
  1
100
```

Figura 63: Porcentajes, total de muestras y porcentaje total de la columna sex

```
actual.casad<-estado.freq[2]

pareja<-estado.freq[3]

divorciado<-estado.freq[4]

viud<-estado.freq[5]

porcent.casad<-nunc.casad/(nunc.casad+alctual.casad+pareja+divorciado+viud))*100
#el porcentaje de casados es la frecuencia de 1 entre el total

porcent.actual<-
(actual.casad/(nunc.casad+alctual.casad+pareja+divorciado+viud))*100

porcent.parej<- (pareja/(nunc.casad+alctual.casad+pareja+divorciado+viud))*100

porcent.divorc<-
(divorciado/(nunc.casad+alctual.casad+pareja+divorciado+viud))*100

porcent.viud<- (viud/(nunc.casad+alctual.casad+pareja+divorciado+viud))*100

estado.civil.freq.total<-nunc.casad+alctual.casad+pareja+divorciado+viud #número
total de muestras

estado.civil.porcentaj.total<-
porcent.casad+porcent.actual+porcent.parej+porcent.divorc+porcent.viud
#porcentaje total (debería devolver "100")
```

```
> porcent.casad
 1
18
> porcent.actual
 2
58
> porcent.parej
3
3
> porcent.divorc
  4
7.5
> porcent.viud
   5
13.5
> estado.civil.freq.total
  1
200
> estado.civil.porcentaj.total
  1
100
```

Figura 63: Porcentajes, total de muestras y porcentaje total de la columna estado.civil

Gracias a este código, se han obtenido los valores con los que hemos rellenado las tablas propuestas.

	Media	Desviación es	Mediana	P25	P75	Missing
Edad	61'625	16'07502	63	52	75	0
Peso	74'35082	14'75374	73	65	82,3	17
Presión arter	131'1311	21'17183	130	119	141,5	17
Presión arter	79'66667	11'5146	80	70	87	17

	Frecuencia (n)	Porcentaje
Sexo	200	100
1	92	46
2	108	54
Estado civil	200	100
Nunca casado	36	18
Actualmente casado	116	58
Viviendo en pareja	6	3
Separado/divorciado	15	7,5
Viudo	27	13,5

Figura 64: Tablas rellenas con los datos recopilados mediante programación en R

La segunda parte del ejercicio 1 consiste en crear gráficos que relacionen una de las variables continuas, en esta memoria se ha optado por elegir peso, y compararla con cada una de las variables demográficas.

Se ha optado por dividir el peso de la población muestreada en intervalos, para permitirnos obtener más información, según cada variable cualitativa. Por ello, en vez de crear histogramas de tres variables (frecuencia, variable continua y variable categórica), se ha optado por crear varios histogramas a comparar, variando el número según los subtipos de cada variable cualitativa.

El código con el que se ha realizado dicho análisis es el siguiente:

```
#segunda parte del ejercicio propuesto 1

int_pesos <- seq(min(datos$peso,na.rm=TRUE),max(datos$peso,na.rm=TRUE),10)
#secuenciar los intervalos de pesos desde el mínimo hasta el máximo, de 10 en 10


#estado civil

nunca.casado <- datos[datos$estado.civil=="1",]$peso #se guarda en el vector
nunca.casado los valores del peso de aquellas personas casadas ("1")

actualmente.casado <- datos[datos$estado.civil=="2",]$peso

viviendo.pareja <- datos[datos$estado.civil=="3",]$peso

separado.divorciado <- datos[datos$estado.civil=="4",]$peso

viudo <- datos[datos$estado.civil=="5",]$peso
```

```
#sexo

sex1 <- datos[datos$sex=="1",]$peso

sex2 <- datos[datos$sex=="2",]$peso


#estudios

no.escuela <- datos[datos$nivel.educacion=="0",]$peso

inf.prim <- datos[datos$nivel.educacion=="1",]$peso

est.prim <- datos[datos$nivel.educacion=="2",]$peso

sec <- datos[datos$nivel.educacion=="3",]$peso

bach <- datos[datos$nivel.educacion=="4",]$peso

univ <- datos[datos$nivel.educacion=="5",]$peso

mast <- datos[datos$nivel.educacion=="6",]$peso


#entorno de residencia

urbano <- datos[datos$entorno.residencia=="1",]$peso

rural <- datos[datos$entorno.residencia=="2",]$peso


#estado civil - histogramas

par(mfrow=c(3,2)) #crear un gráfico de gráficos (3x2)

hist(pesos1, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Nunca casado",
col="darkslategray1") #breaks son divisomes, x e ylab permiten etiquetar los
ejes, main permite etiquetar el gráfico

hist(pesos2, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Actualmente
casado", col="darkslategray2")

hist(pesos3, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Viviendo en
pareja", col="darkslategray3")

hist(pesos4, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Separado/a,
Divorciado", col="darkslategray4")

hist(pesos5, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Viudo",
col="darkslategray")


#sexo - histogramas

par(mfrow=c(2,1))

hist(sex1, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Hombres",
col="firebrick")

hist(sex2, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Mujeres",
col="firebrick1")
```

```
#estudios - histogramas

par(mfrow=c(3,3))

hist(no.escuela, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Nunca ha
ido a la escuela", col="palevioletred1")

hist(inf.prim, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Inferior a
estudios primarios", col="palevioletred2")

hist(est.prim, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Estudios
primarios", col="palevioletred3")

hist(sec, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Secundaria",
col="palevioletred")

hist(bach, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Bachillerato (o
equivalente) completado", col="pink1")

hist(univ, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Estudios
universitarios", col="pink2")

hist(mast, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Máster o
Doctorado", col="pink3")


#entorno de residencia - histogramas

par(mfrow=c(1,2))

hist(urbano, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Entorno de
residencia urbano", col="springgreen1")

hist(rural, breaks=int_pesos,xlab="Peso",ylab="Frecuencia",main="Entorno de
residencia rural", col="springgreen3")
```

Los histogramas creados son los que se van a analizar a continuación:

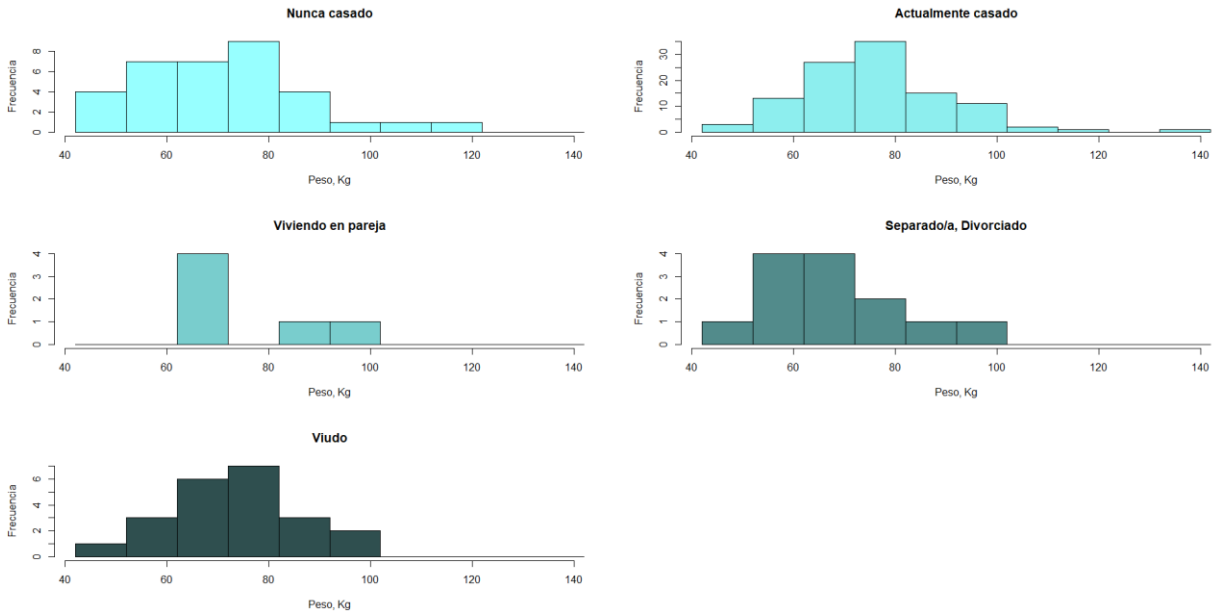

Figura 65: Histogramas que relacionan peso y estado civil

Antes de proceder a analizar la figura, se debe tener en cuenta que las muestras tomadas no son equiparables, puesto que, frente a un, por ejemplo, 58% de personas casadas, se tiene sólo un 3% de personas viviendo en pareja, por lo que las conclusiones extraídas de este histograma no son fieles a la realidad. Aparte de este inconveniente, se puede observar que aquellas personas que superan los 110 Kg, pertenecen al grupo de personas nunca casadas y personas actualmente casadas. Por el otro lado, se puede observar que la frecuencia de peso por excelencia se concentra en los pesos entre los 60 y los 80 Kg, por lo que podría decirse que la población extraída para muestrear se mantiene en la media de peso nacional (84 Kg). Podría concluirse, que, como se suele decir, con el matrimonio las personas se vuelven más dejadas, mientras que aquellos viviendo en pareja y divorciados se mantienen más sanos físicamente.

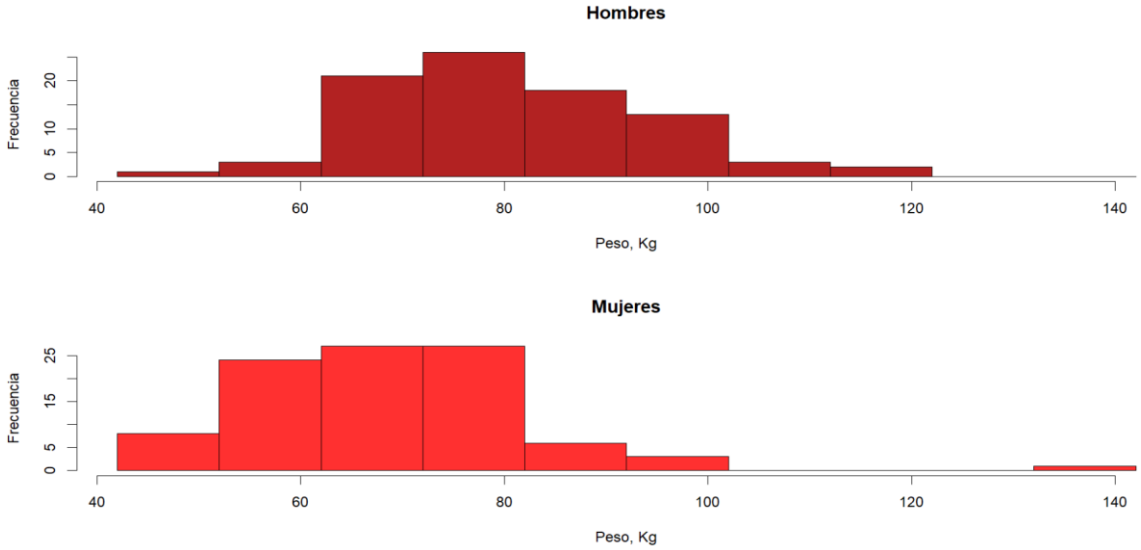

Figura 65: Histogramas que relacionan peso y sexo

De este par de histogramas se puede concluir que, como es de esperar fisiológicamente, los hombres tienen por lo general mayor peso que las mujeres (a pesar de que el poco porcentaje que ronda los 140Kg corresponde a una mujer). La distribución del peso de los hombres sigue prácticamente una distribución normal.

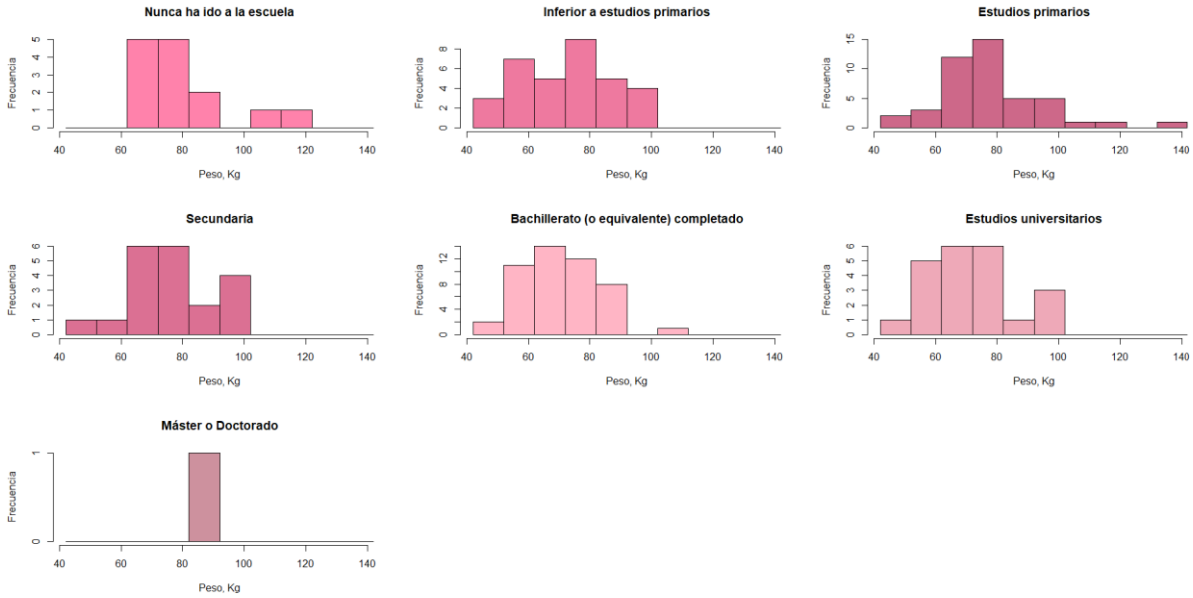

Figura 66: Histogramas que relacionan peso y nivel de educación

Es realmente interesante observar que aquellas personas con los estudios más altos, se mantienen en intervalos inferiores a los 110 Kg, y esta distribución de peso va creciendo cuanto más bajo es el nivel educativo (el mayor rango de peso se encuentra entre gente que no ha ido a la escuela, o que se quedaron en los estudios primarios). Es muy probable que la falta de conocimiento general sea el responsable de que haya una fracción de personas con sobrepeso.

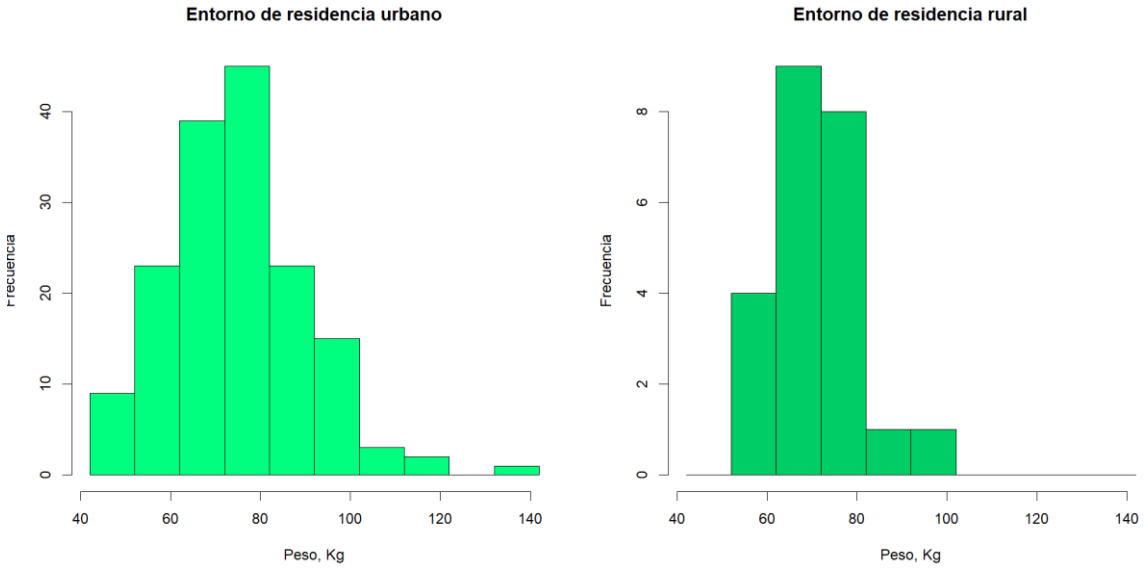

Figura 66: Histogramas que relacionan peso y nivel de educación

Por último, como podría intuirse, aquellas personas que viven en un entorno rural muestran una menor tendencia a sobrepasar los 110 Kg, mientras que en el caso del entorno residencial urbano recoge a personas con mayor sobrepeso (se ha de aclarar que el porcentaje de muestra que vive en dicho entorno es mayor). Se podría interpretar según estos datos, que es más sano vivir en un entorno rural, tanto por el deporte que se hace, como por la alimentación que engloba.

3.2. Ejercicio 2

Este segundo y último ejercicio propuesto, ofrece la siguiente tabla de valores:

Intervalos	15-25	25-35	35-40	40-45	45-55	55-65
Frecuencias	10	20	15	20	20	15

Figura 67: Tabla propuesta de intervalos y frecuencias

Utilizando esos datos, se pide dibujar el histograma y el polígono de distribución a partir de dichos datos. Se especifica que la superficie de cada rectángulo debe ser proporcional a la frecuencia de dicho intervalo, por lo que se da a entender que se debe crear un histograma cuyo eje y indique la densidad de las frecuencias.

Pero antes de mostrar el histograma de densidad, se ha querido mostrar un histograma junto a su polígono de distribución de frecuencias, que permite plasmar la tabla propuesta de intervalos y frecuencias, tal cual se ofrece.

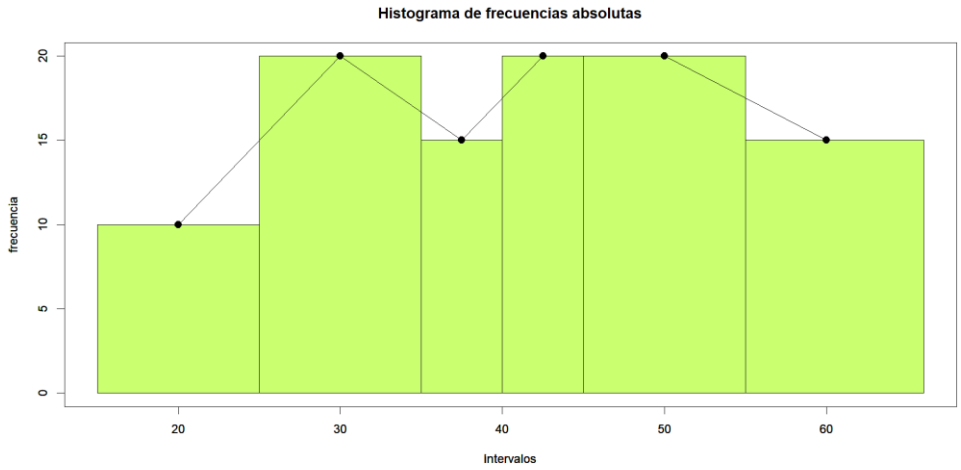

Figura 67: Tabla propuesta de intervalos y frecuencias

Ahora sí, la figura 68 representa el histograma de densidad, de frecuencias absolutas:

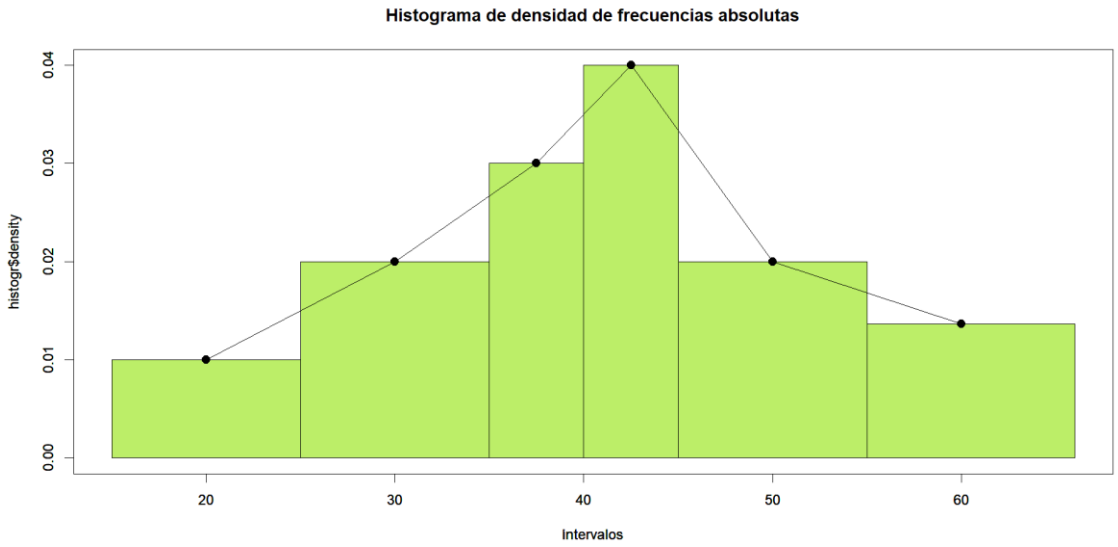

Figura 68: Histograma y polígono de distribución

Exactamente como se esperaba, aquellos intervalos cuyas frecuencias absolutas son menores, representan una menor superficie en cada rectángulo.

Ahora bien, estas frecuencias ofrecidas se podrían haber tomado como frecuencias acumuladas, por lo que el último intervalo tendrá una incidencia de hasta 100. Esto se ha querido representar en el histograma de la figura 69:

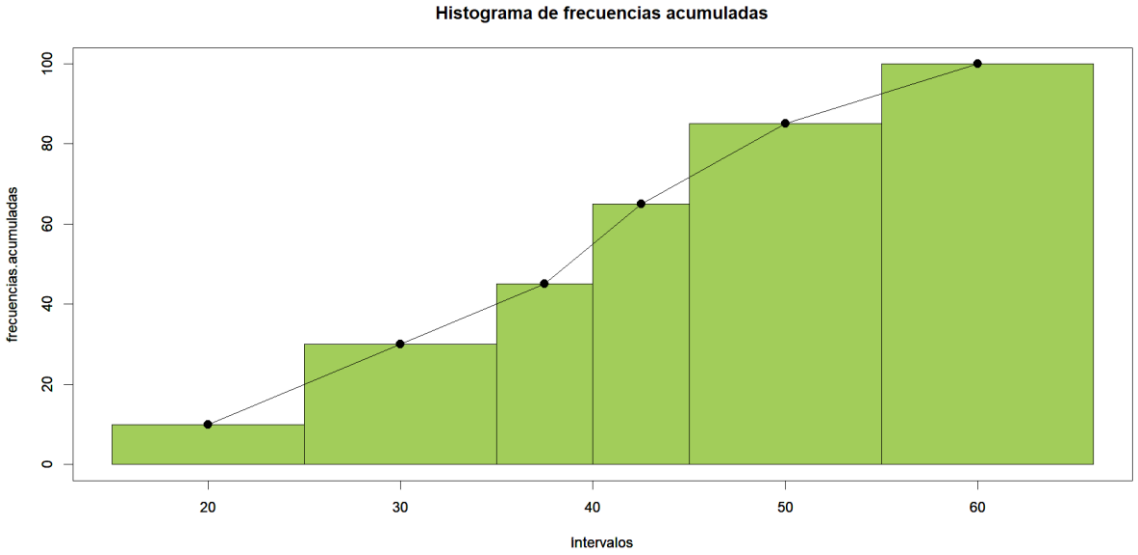

Figura 69: Histograma y polígono de distribución de los intervalos con frecuencias acumuladas

En este gráfico podría observarse a ojo los cuantiles.

Recordemos que teóricamente, un cuantil es un punto que divide la función de distribución de una variable aleatoria en intervalos regulares. En nuestro histograma podemos tomar el eje y como centiles, y el eje x en aquellos intervalos irregulares referidos en la definición teórica.

Por ejemplo, observando la gráfica, se estima que el centil 30% quedará sobre el intervalo 40, mientras que el percentil 70%, alcanza el intervalo 50.

Dichos valores se han comprobado con el código, y efectivamente:

```
> quantile(intervalos.frecuencia.abs, 0.3)
 30%
42.5
> quantile(intervalos.frecuencia.abs, 0.7)
70%
 50
```

Figura 70: Cálculo de los centiles 30 y 70%

Por último, por completar el orden de histogramas ofrecidos en este ejercicio, se muestra el histograma de frecuencias acumuladas, en el que se muestra en el eje y la densidad.

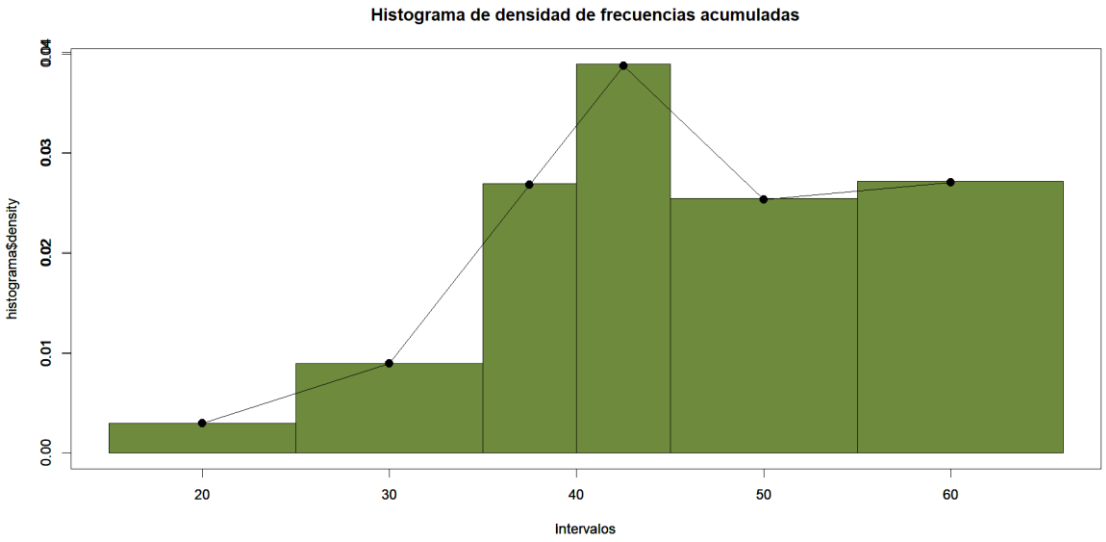

Figura 71: Histograma de densidad y polígono de distribución de frecuencias acumuladas

Se ofrece ahora el código comentado:

```
setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-1")

#frecuencias absolutas

intervalos<-c(20,30,37.5,42.5,50,60) #eje x

frecuencia <- c(10, 20, 15, 20, 20, 15) #eje x

intervalos.frecuencia <- rep(intervalos,frecuencia) #repeticiones para poder crear el histograma

brk=c(15,25,35,40,45,55,66) #los breaks del histograma


#frecuencias acumuladas

frecuencias.acumuladas <- c(10, 30, 45, 65, 85, 100) #frecuencias acumuladas

intervalos.frecuencia.abs <- rep(intervalos,frecuencias.acumuladas)


#histograma de frecuencias absolutas

hist(intervalos.frecuencia,breaks=brk, freq=TRUE, xlab = "Intervalos",ylab = "",main = "Histograma de frecuencias absolutas", col="darkolivegreen1")

par(new=TRUE) #permite superponer gráficos

plot(intervalos,frecuencia, plot(frecuencia~intervalos,type="l",xlim=c(15,66), ylim=c(0,20)),cex=2,pch=20,bg="black",col="black") #este plot muestra el polígono de distribución (el tipo l indica puntos conectados por líneas)
```

```
#histograma de densidad

histogr<- hist(intervalos.frecuencia, breaks=brk, xlab = "Intervalos",ylab =
"",main = "Histograma de densidad de frecuencias absolutas",
col="darkolivegreen2")

par(new=TRUE)

plot(intervalos, histogr$density, plot(histogr$density~intervalos,xlim=c(15,66),
ylim=c(0,0.04),type="l"),cex=2,pch=20,bg="black",col="black")


#histograma de frecuencias acumuladas

hist(intervalos.frecuencia.abs, breaks=brk, freq=TRUE, xlab = "Intervalos",ylab
= "",main = "Histograma de frecuencias acumuladas", col="darkolivegreen3")

par(new=TRUE)

plot(intervalos,frecuencias.acumuladas,
plot(frecuencias.acumuladas~intervalos,type="l",xlim=c(15,66),
ylim=c(0,100)),cex=2,pch=20,bg="black",col="black")


#histograma de frecuencias acumuladas por densidades

histograma<- hist(intervalos.frecuencia.abs, breaks=brk, xlab =
"Intervalos",ylab = "",main = "Histograma de densidad de frecuencias
acumuladas", col="darkolivegreen4")

par(new=TRUE)

plot(intervalos, histograma$density,
plot(histograma$density~intervalos,xlim=c(15,66),
ylim=c(0,0.03899),type="l"),cex=2,pch=20,bg="black",col="black")


quantile(intervalos.frecuencia.abs, 0.3) #para calcular los cuantiles

quantile(intervalos.frecuencia.abs, 0.7)
```

4. CONCLUSIONES

Este primer contacto con programación en R ha permitido al alumno familiarizarse con los conceptos más básicos de la bioestadística vistos en clase. El alumno ha descubierto la facilidad y la rapidez con la que se pueden trabajar grandes cantidades de datos, utilizando RStudio.

A pesar de la complicación que resulta siempre enfrentarse a un nuevo lenguaje de programación, R es intuitivo y en los diversos tutoriales que ofrece Internet, se puede extraer toda la información necesaria para crear los códigos.