PEC1 - Preproceso de datos

Gloria Manresa Santamaría

2022-11-03

1. Carga del archivo

Cargamos el archivo de datos y lo guardamos bajo el nombre my_data, mostramos las primeras y las últimas líneas para asegurarnos de que se han cargado correctamente todas las entradas. También comprobamos que el número de entradas en la base da datos original coincide con el número de líneas cargadas en R.

```
my_data <- read.csv('gpa_row.csv')
knitr::kable(head(my_data,3))
```

sat	tothrs	hsize	hsrank	hsperc	colgpa	athlete	female	white	black
920	43h	0.1	4	40.00000	2.04	TRUE	TRUE	FALSE	FALSE
1170	18h	9.3999996	191	20.31915	4.00	FALSE	FALSE	TRUE	FALSE
810	14h	1.1900001	42	35.29412	1.78	TRUE	FALSE	TRUE	FALSE

```
knitr::kable(tail(my_data,3))
```

	sat	tothrs	hsize	hsrank	hsperc	colgpa	athlete	female	white	black
4135	1340	62h	0.44999999	1	2.222222	4.00	FALSE	FALSE	TRUE	FALSE
4136	980	12h	0.34999999	23	65.714287	2.83	FALSE	TRUE	TRUE	FALSE
4137	1420	128h	3.1300001	38	12.140580	3.94	FALSE	FALSE	TRUE	FALSE

Confirmamos que las 4137 entradas coinciden con la líneas del documento excel original.

Comprobamos el tipo de datos con el que R ha interpretado cada variable:

```
knitr::kable(sapply(my_data, class),col.names="Tipo de datos")
```

	Tipo de datos
sat	integer
tothrs	character
hsize	character
hsrank	integer
hsperc	numeric
colgpa	numeric
athlete	character

	Tipo de datos
female	logical
white	character
black	character

Podemos oberservar que hay dos variables que R no ha interpretado como numéricas y deberían serlo, se trata de las variables "tothrs" y "hsize".

2. Normalización de las variables cualitativas

2.1 Athlete

R ha interpretado la variable Athlete como "character" cuando en realidad se trata de una variable categórica.

Observamos los valores que toma la variable "Athlete" antes de iniciar su transformación:

```
knitr::kable(table(my_data$athlete),col.names=c("Valores Athlete","Frecuencia absoluta"))
```

Valores Athlete	Frecuencia absoluta
false	11
FALSE	3932
true	1
TRUE	193

Vemos que toma 4 valores diferentes ya que en ocasiones se encuentra en minusculas. Antes de transformarla a categórica habrá que pasar toda la variable a mayúsculas para que solo tome dos atributos, TRUE y FALSE:

```
my_data$athlete <- as.factor(toupper(my_data$athlete))
```

Comprobamos que se ha transformado correctamente:

```
class(my_data$athlete)
```

```
## [1] "factor"
```

```
levels(my_data$athlete)
```

```
## [1] "FALSE" "TRUE"
```

2.2 Female

La variable Female se ha interpretado como una variable tipo "logical". Los criterios de verificación y de normalización del enunciado dice que las variables del tipo indicador deben codificarse como variables categóricas ("factor") con los valores TRUE y FALSE.

Transformamos la variable a "factor" y comprobamos que se ha realizado correctamente la transformación:

```
my_data$female <- as.factor(my_data$female)
class(my_data$female)
```

```
## [1] "factor"
```

```
levels(my_data$female)
```

```
## [1] "FALSE" "TRUE"
```

2.3. Black

R ha interpretado la variable black del tipo "character". En realidad esta variable debe ser "factor" por lo que iniciamos su transformación. En primer lugar comprobamos los diferentes valores que toma esta variable:

```
knitr::kable(table(my_data$black))
```

Var1	Freq
FALSE	3
false	10
FALSE	3890
FALSE	5
TRUE	228
TRUE	1

Observamos que toma 6 valores diferentes cuando deberían ser únicamente 2. Los valores extra se han creado por la existencia de espacios y por encontrarse en minúsculas.

Antes de pasar la variable a categórica borramos los espacios y pasamos a mayúsculas:

```
my_data$black <- as.factor((trimws(toupper(my_data$black))))
```

Comprobamos que la variable se ha transformado correctamente:

```
class(my_data$black)
```

```
## [1] "factor"
```

```
levels(my_data$black)
```

```
## [1] "FALSE" "TRUE"
```

2.4 White

Con la variable white pasa lo mismo que con la variable black. Así que procedemos de la misma manera:

```
my_data$white <- as.factor((trimws(toupper(my_data$white))))
```

Comprobamos que la variable se ha transdormado correctamente:

```
class(my_data$white)
```

```
## [1] "factor"
```

```
levels(my_data$white)
```

```
## [1] "FALSE" "TRUE"
```

3. Normalización de las variables cuantitativas

3.1 Nota de acceso

El formato de la variable "sat" es actualmente "integer". Mostramos algunos de los valores que toma esta variable al inicio y al final del dataframe.

```
head(my_data$sat,15)
```

```
##  [1]  920 1170  810  940 1180  980  880  980 1240 1230 1140 1150 1080  990 1000
```

```
tail(my_data$sat,15)
```

```
##  [1] 1010  850 1200 1260  950 1120 1000 1020 1180 1150  990  900 1340  980 1420
```

Por el tipo de dato de la variable "sat" y según comprobamos en la muestra de datos podemos confirmar que el formato de la variable "integer" es correcto.

Esta variable se trata de la nota de acceso medida en la escala de 400 a 1600 puntos por lo que habría que comprobar que no existe ningún dato fuera de este rango de valores.

```
summary(my_data$sat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     470     940    1030    1030    1120    1540
```

Confirmamos que se cumple este requisito siendo el valor mínimo 470 y el máximo 1540.

3.2 Horas totales cursadas al semestre

Revisamos el formato de la variable "tothrs" y mostramos algunos de los valores que toma.

```
class(my_data$tothrs)
```

```
## [1] "character"
```

```
head(my_data$tothrs,10)
```

```
##  [1] "43h"  "18h"  "14h"  "40h"  "18h"  "114h" "78h"  "55h"  "18h"  "17h"
```

Comprobamos que el formato de la variable es "character" y que los valores presentan la unidad de la variable. En primer lugar deberíamos eliminar la unidad de la variable, "h", para poder transformar la variable a numérica.

```
library(stringr)
my_data$tothrs <- str_extract(my_data$tothrs,'\\d+')
tail(my_data$tothrs,10)
```

```
##  [1] "16"  "18"  "75"  "47"  "18"  "49"  "50"  "62"  "12"  "128"
```

Una vez eliminada la unidad de la variable podemos pasarla a formato "numeric" tal y como solicita el enunciado.

```
my_data$tothrs <- as.numeric(my_data$tothrs)
```

Confirmamos que la transformación se ha realizado correctamente:

```
class(my_data$tothrs)
```

```
## [1] "numeric"
```

3.3 Nota media del estudiante al final del primer semestre

La variable "colgpa" es la nota media del estudiante al final del primer semestre, medida en escala de 0 a 4 puntos.

Comprobamos el formato de la variable y que los valores pertenecen al rango especificado, entre 0 y 4.

```
class(my_data$colgpa)
```

```
## [1] "numeric"
```

```
summary(my_data$colgpa)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   2.210   2.660   2.655   3.120   4.000      41
```

Podemos comprobar que el formato es "numeric", que el valor mínimo es 0.000 y el máximo 4.000 por lo que no es necesario hacer ninguna transformación por el momento.

3.4 Número total de estudiantes en la cohorte de graduados del bachillerato

Comprobamos el formato de la variable y mostramos los primeros valores que toma la variable.

```
class(my_data$hsize)
```

```
## [1] "character"
```

```
head(my_data$hsize,6)
```

```
## [1] "0.1"       "9.3999996" "1.1900001" "5.71"      "2.1400001" "2.6800001"
```

Podemos observar que los primeros 6 valores utilizan el punto como separador decimal tal y como se solicita en el enunciado. Aun así no podemos estar seguros de que esto se cumple en todos los valores que toma la variable. De hecho con el siguiente código confirmamos que existen valores que utilizan la coma como delimitador en lugar del punto.

```
str_view(my_data$hsize, ",",match=TRUE)
```

Tomamos un valor como referencia para poder comprobar la transformación del punto a la coma:

```
my_data$hsize[53]
```

```
## [1] "2,54"
```

Realizamos el cambio de la coma a punto en todos los valores de "hsize" y confirmamos con el valor de referencia que se ha realizado correctamente la transformación:

```
my_data$hsize <- gsub(",",".",my_data$hsize)
my_data$hsize[53]
```

```
## [1] "2.54"
```

Anteriormente también hemos podido observar que R ha interpretado la variable como "character" cuando debería ser numérica. Transformamos la variable a numérica y comprobamos el resultado:

```
my_data$hsize <- as.numeric(my_data$hsize)
class(my_data$hsize)
```

```
## [1] "numeric"
```

```
head(my_data$hsize,10)
```

```
##  [1] 0.10 9.40 1.19 5.71 2.14 2.68 3.11 2.68 3.67 0.10
```

3.5 Ranking relativo del estudiante

Comprobamos el formato de la variable y observamos los primeros valores.

```
class(my_data$hsperc)
```

```
## [1] "numeric"
```

```
head(my_data$hsperc,7)
```

```
## [1] 40.00000 20.31915 35.29412 44.13310 40.18692 15.29851 51.76849
```

La variable se trata de una variable numérica, que es correcto. Ahora procedemos a confirmar que los valores que toman esta variable coinciden con los valores que según el enunciado deben tomar, que es "hsrank/hsize", para la comparación se utilizaran los primeros 3 decimales únicamente.

```
table(round(my_data$hsperc,3) == round(my_data$hsrank/my_data$hsize,3))
```

```
##
## FALSE  TRUE
##    12  4125
```

Con el resultado del código anterior podemos confirmar que existen 12 entradas en las que no coinciden los resultados.

Comprobamos los 12 resultados para decidir como proceder:

```
# Valores de la variable "hsperc"
round(my_data$hsperc[((round(my_data$hsperc,3) ==
                         round(my_data$hsrank/my_data$hsize,3)))== FALSE],3)
```

```
##  [1] 56.554 18.418 37.345 41.097 24.479 54.223  8.944 21.484 15.174 17.368
## [11] 46.098 20.852
```

```
# Valores "hsrank/hsize"
round((my_data$hsrank/my_data$hsize)[((round(my_data$hsperc,3) ==
                        round(my_data$hsrank/my_data$hsize,3)))== FALSE],3)
```

```
##  [1] 55.280 16.620 37.073 39.401 23.437 53.587  7.813 20.313 14.063 15.426
## [11] 44.720 19.687
```

Las diferencias observadas se tratan en algunos casos del orden de unidades. Modificamos los valores de las variables para que en todos los casos se cumpla "hsrank/hsize":

```
my_data$hsperc<- my_data$hsrank/my_data$hsize
```

Comprobamos que se ha modificado correctamente:

```
table(round(my_data$hsperc,3) == round(my_data$hsrank/my_data$hsize,3))
```

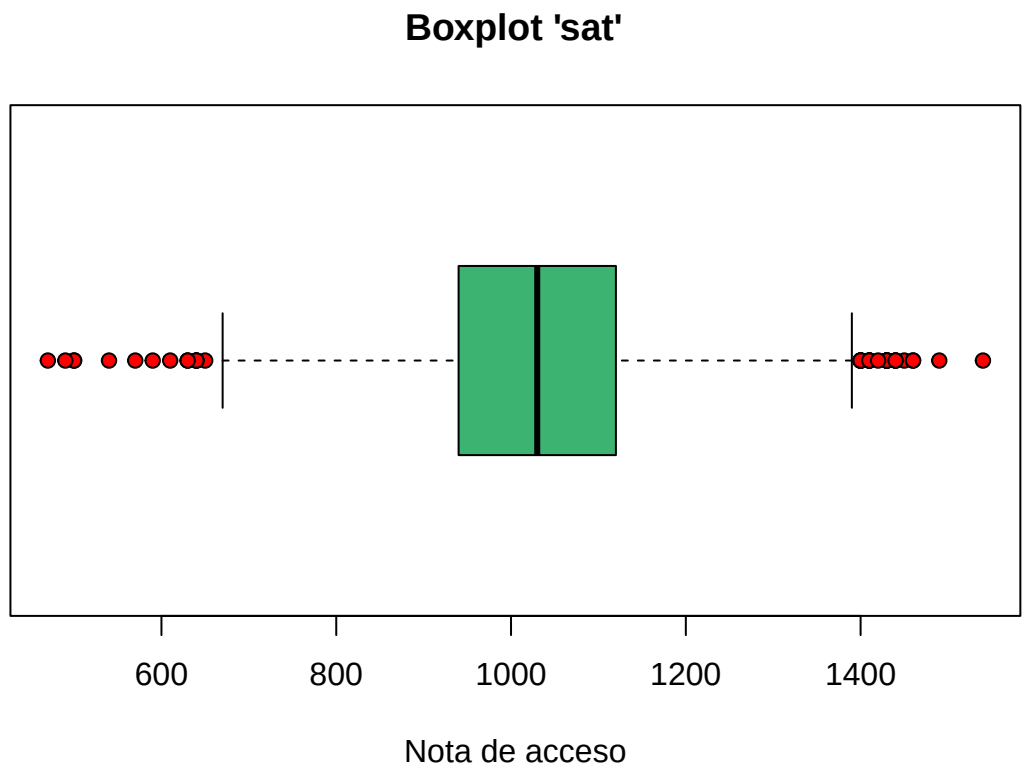
```
##
## TRUE
## 4137
```

4. Valores atípicos

- **Nota de acceso, "sat".**

Comprobaremos si existen valores atípicos en la variable "sat" realizando un diagrama de caja:

```
boxplot(my_data$sat,
        main = "Boxplot 'sat'",
        xlab = "Nota de acceso",
        outpch = 21,
        outbg = "red",
        horizontal=T,
        col="mediumseagreen"
        )
```



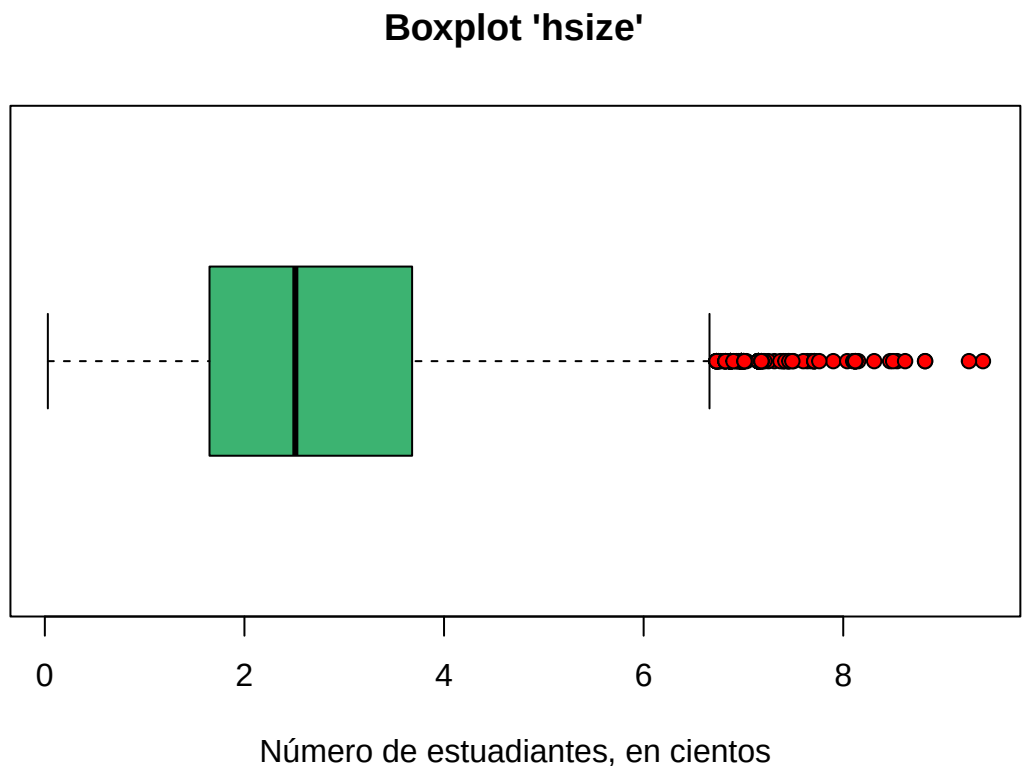
Observamos que sí que existen valores atípicos en la variable "sat" en los dos extremos de la muestra. Los valores atípicos, en rojo, no los consideraremos anómalos ya que se encuentran dentro del rango de la variable.

- **Número total de estudiantes, "hsize".**

La variable "hsize" presenta el siguiente diagrama de caja:

```
boxplot(my_data$hsize,
        main = "Boxplot 'hsize'",
```

```
        xlab = "Número de estuadiantes, en cientos",
        outpch = 21,
        outbg = "red",
        horizontal=T,
        col="mediumseagreen"
        )
```



Comprobamos que también existen valores atípicos, en este caso se encuentran en el límite superior.

Estos valores atípicos también se encuentran dentro del rango de la variable por lo que no se consideran anómalos.

5. Imputación de valores

En primer lugar vamos a buscar los valores perdidos, si los hay, de las diferentes variables cuantitativas.

- **Nota de acceso "sat"**

```
table(is.na(my_data$sat))
```

```
##
## FALSE
##  4137
```

No existen valores perdidos en la variable "sat".

- **Horas totales cursadas en el semestre "tothrs"**

```
table(is.na(my_data$tothrs))
```

```
##
## FALSE
##  4137
```

No existen valores perdidos en la variable "tothrs".

- **Número total de estudiantes "hsize"**

```
table(is.na(my_data$hsize))
```

```
##
## FALSE
##  4137
```

No existen valores perdidos en la variable "hsize".

- **Ranking del estudiante "hsrank"**

```
table(is.na(my_data$hsrank))
```

```
##
## FALSE
##  4137
```

No existen valores perdidos en la variable "hsrank".

- **Ranking relativo del estudiante "hsperc"**

```
table(is.na(my_data$hsperc))
```

```
##
## FALSE
##  4137
```

No existen valores perdidos en la variable "hsperc".

- **Nota media del estudiante al final del semestre "colgpa"**

```
table(is.na(my_data$colgpa))
```

```
##
## FALSE  TRUE
##  4096    41
```

Hay 41 valores perdidos en la variable "colgpa", que recordamos hace referencia a la nota media del estudiante al final del primer semestre, medido entre 0 y 4 puntos.

Vamos a estudiar por separado los valores femeninos y los masculinos.

Valores femeninos Creamos un nuevo dataframe únicamente con las variables cuantitativas que hacen referencia a mujeres.

```
fem1 <- subset(my_data,my_data$female==TRUE)
borrar <- c("athlete","female","white","black")
fem <- fem1[ , !(names(fem1) %in% borrar)]
knitr::kable(head(fem,3))
```

	sat	tothrs	hsize	hsrank	hsperc	colgpa
1	920	43	0.10	4	40.00000	2.04
6	980	114	2.68	41	15.29851	3.03
11	1140	78	3.34	95	28.44311	2.98

Buscamos cuantos de los valores perdidos de la variable "colgpa" corresponden a mujeres:

```
table(is.na(fem$colgpa))
```

```
##
## FALSE  TRUE
##  1843    17
```

Guardamos los índices de los valores perdidos para las mujeres para más adelante comprobar que se han imputado correctamente los nuevos valores.

```
index_fem <- which((my_data$female==TRUE)&(is.na(my_data$colgpa)==TRUE))
```

Con la función kNN() del paquete VIM calculamos los valores perdidos con el modelo basado en los k vecinos más próximos, en este caso se utilizan 11.

```
library(VIM)
new.fem <- kNN(fem,k=11)
```

Imputamos los nuevos valores calculados en el dataframe original:

```
my_data$colgpa[is.na(my_data$colgpa)==TRUE&my_data$female==TRUE] <-
  new.fem$colgpa[new.fem$colgpa_imp==TRUE]
```

Con los índices guardados anteriormente de los valores perdidos comprobamos los nuevos valores imputados en el dataframe global. También comprobamos que no existan valores perdidos en el caso de las mujeres.

```
table(is.na(my_data$colgpa[my_data$female==TRUE]))
```

```
##
## FALSE
##  1860
```

```
my_data$colgpa[index_fem]
```

```
##  [1] 3.31 2.60 2.82 2.81 3.15 2.55 2.59 2.78 2.37 2.19 2.72 2.74 2.78 2.67 2.60
## [16] 2.41 2.46
```

Valores masculinos Seguimos los mismos pasos en el caso de los hombres. En primer lugar comprobamos cuantos valores perdidos existen y creamos un nuevo dataframe de las variables cuantitavas en caso de los hombres.

```
masc1 <- subset(my_data,my_data$female==FALSE)
masc <- masc1[ , !(names(masc1) %in% borrar)]
table(is.na(masc$colgpa))
```

```
##
## FALSE  TRUE
##  2253    24
```

Podemos observar que existen 24 valores perdidos de hombres, guardamos sus índices para poder comprobar más adelante los nuevos valores en el dataframe global.

```
index_masc <- which((my_data$female==FALSE)&(is.na(my_data$colgpa)==TRUE))
```

Calculamos los valores perdido con el modelo de k (11 en este caso) vecinos más próximos.

```
new.masc <- kNN(masc,k=11)
```

Imputamos los nuevos valores obtenidos al dataframe global:

```
my_data$colgpa[is.na(my_data$colgpa)==TRUE&my_data$female==FALSE] <-
  new.masc$colgpa[new.masc$colgpa_imp==TRUE]
```

Comprobamos que no existen valores perdidos de la variable "colgpa" en hombres y con los índices tomados anteriormente de los valores perdidos mostramos los valores que toman ahora esta variable:

```
table(is.na(my_data$colgpa[my_data$female==FALSE]))
```

```
##
## FALSE
##  2277
```

```
my_data$colgpa[index_masc]
```

```
##  [1] 2.35 2.65 2.48 2.18 2.15 2.26 2.72 2.26 2.43 2.81 2.60 3.46 2.69 2.50 2.50
## [16] 2.68 2.35 2.76 2.42 2.26 3.23 2.74 3.41 1.68
```

Realizamos una última comprobación para asegurarnos que todos los valores han sido sustituidos tanto para hombres como para mujeres:

```
table(is.na(my_data$colgpa))
```

```
##
## FALSE
##  4137
```

6. Creación de una nueva variable

Creamos una nueva variable con el nombre "gpaletter" y que tome los siguientes valores en función de los valores de "gpa": - A: de 3.5 a 4.00 - B: de 2.5 a 3.49 - C: de 1.5 a 2.49 - D: de 0 a 1.49

```
library(tidyverse)
my_data <- my_data %>%
mutate(gpaletter = case_when(my_data$colgpa<=4&my_data$colgpa>=3.5 ~ 'A'
                             ,my_data$colgpa>=2.5 & my_data$colgpa<=3.49 ~ 'B'
                             ,my_data$colgpa>=1.5 & my_data$colgpa<=2.49 ~ 'C'
                             ,my_data$colgpa>=0 & my_data$colgpa<=1.49 ~ 'D'

                             )
  )
```

Comprobamos que se ha creado correctamente la nueva variable:

```
head(my_data,5)
```

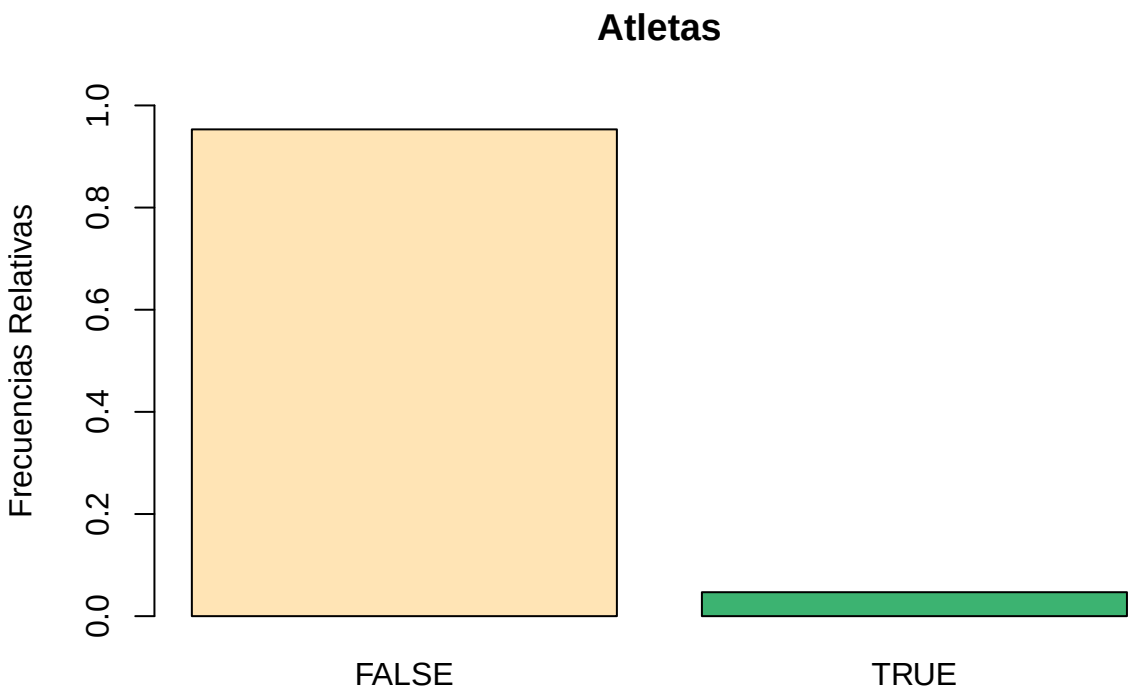
```
##    sat tothrs hsize hsrank   hsperc colgpa athlete female white black gpaletter
## 1  920     43  0.10      4 40.00000   2.04    TRUE   TRUE FALSE FALSE         C
## 2 1170     18  9.40    191 20.31915   4.00   FALSE  FALSE  TRUE FALSE         A
## 3  810     14  1.19     42 35.29411   1.78    TRUE  FALSE  TRUE FALSE         C
## 4  940     40  5.71    252 44.13310   2.42   FALSE  FALSE  TRUE FALSE         C
## 5 1180     18  2.14     86 40.18691   2.61   FALSE  FALSE  TRUE FALSE         B
```

7. Estudio descriptivo

7.1 Estudio descriptivo de las variables cualitativas

Realizamos un gráfico que muestre la variable "athlete" en porcentajes, es decir, la frecuencia relativa de ésta:

```
barplot(prop.table(table(my_data$athlete)),
        col=c("moccasin","mediumseagreen"),
        ylim=c(0,1),
        main="Atletas",
        ylab ="Frecuencias Relativas")
```



Podemos observar que existe una gran diferencia entre los valores que toma esta variable. La mayoría de personas de la muestra no son atletas y únicamente un pequeño porcentaje sí que lo son.

Queremos comprobar cual es la distribución en caso de separar los hombre y las mujeres. Para ello se han realizado dos gráficos diferentes uno con todos los hombres y otro con todas las mujeres, de manera que podemos comparar los resultados ya que se ha realizado con las frecuencias relativas, en porcentaje.

```
par(mfrow=c(1,2))
barplot(prop.table(table(my_data$athlete[my_data$female==FALSE])),
        main = "Hombres",
        ylab="Frecuencia relativa",
        xlab = "Atletas",
        ylim = c(0, 1),
        col=c("mediumseagreen"),
        beside=TRUE)
barplot(prop.table(table(my_data$athlete[my_data$female==TRUE])),
        main = "Mujeres",
        xlab = "Atletas",
        ylim = c(0, 1),
        col=c("moccasin"),
        beside=TRUE
        )
```

Del gráfico leemos que el porcentaje de hombres atletas es mayor al porcentaje de mujeres atletas.

7.2 Estudio descriptivo de las variables cuantitativas

En primer lugar calcularemos diferentes estimadores para más adelante analizarlos.

```
sat.meanh <- round(mean(my_data$sat[my_data$female==FALSE]),2)
sat.medianh <- round(median(my_data$sat[my_data$female==FALSE]),2)
sat.meanth <- round(mean(my_data$sat[my_data$female==FALSE],trim=0.05),2)
sat.sdh <- round(sd(my_data$sat[my_data$female==FALSE]),2)
sat.rich <- round(IQR(my_data$sat[my_data$female==FALSE]),2)
sat.damh <- round(mad(my_data$sat[my_data$female==FALSE]),2)

sat.meanm <- round(mean(my_data$sat[my_data$female==TRUE]),2)
sat.medianm <- round(median(my_data$sat[my_data$female==TRUE]),2)
sat.meantm <- round(mean(my_data$sat[my_data$female==TRUE],trim=0.05),2)
sat.sdm <- round(sd(my_data$sat[my_data$female==TRUE]),2)
sat.ricm <- round(IQR(my_data$sat[my_data$female==TRUE]),2)
sat.damm <- round(mad(my_data$sat[my_data$female==TRUE]),2)

tothrs.mean <- round(mean(my_data$tothrs),2)
tothrs.median <- round(median(my_data$tothrs),2)
tothrs.meant <- round(mean(my_data$tothrs,trim=0.05),2)
tothrs.sd <- round(sd(my_data$tothrs),2)
tothrs.ric <- round(IQR(my_data$tothrs),2)
tothrs.dam <- round(mad(my_data$tothrs),2)
```

```
hsize.mean <- round(mean(my_data$hsize),2)
hsize.median <- round(median(my_data$hsize),2)
hsize.meant <- round(mean(my_data$hsize,trim=0.05),2)
hsize.sd <- round(sd(my_data$hsize),2)
hsize.ric <- round(IQR(my_data$hsize),2)
hsize.dam <- round(mad(my_data$hsize),2)

hsrank.mean <- round(mean(my_data$hsrank),2)
hsrank.median <- round(median(my_data$hsrank),2)
hsrank.meant <- round(mean(my_data$hsrank,trim=0.05),2)
hsrank.sd <- round(sd(my_data$hsrank),2)
hsrank.ric <- round(IQR(my_data$hsrank),2)
hsrank.dam <- round(mad(my_data$hsrank),2)

hsperc.mean <- round(mean(my_data$hsperc),2)
hsperc.median <- round(median(my_data$hsperc),2)
hsperc.meant <- round(mean(my_data$hsperc,trim=0.05),2)
hsperc.sd <- round(sd(my_data$hsperc),2)
hsperc.ric <- round(IQR(my_data$hsperc),2)
hsperc.dam <- round(mad(my_data$hsperc),2)
```

- **Nota de acceso "sat"**

Estadísticos de la variable "sat":

```
df.7.sat <- data.frame("SAT_HOMBRES" = c(sat.meanh,sat.medianh,sat.meanth,sat.sdh,sat.rich,sat.damh),
                       "SAT_MUJERES" = c(sat.meanm,sat.medianm,sat.meantm,sat.sdm,sat.ricm,sat.damm),
                     row.names= c("Media aritmética","Mediana","Media recortada","Desviación estándar","Ra
knitr::kable(df.7.sat)
```

	SAT_HOMBRES	SAT_MUJERES
Media aritmética	1049.70	1006.62
Mediana	1050.00	1000.00
Media recortada	1050.15	1005.26
Desviación estándar	144.98	128.37
Rango intercuartílico	180.00	170.00
Desviación absoluta respecto de la mediana	133.43	133.43

Summary de la variable "sat":

```
summary(my_data$sat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     470     940    1030    1030    1120    1540
```

Histograma:

```
hist(my_data$sat,
     main = "Histograma notas de acceso",
     col="moccasin",
     xlab = "Nota de acceso",
     ylab = "Frecuancia absoluta"
     )
```

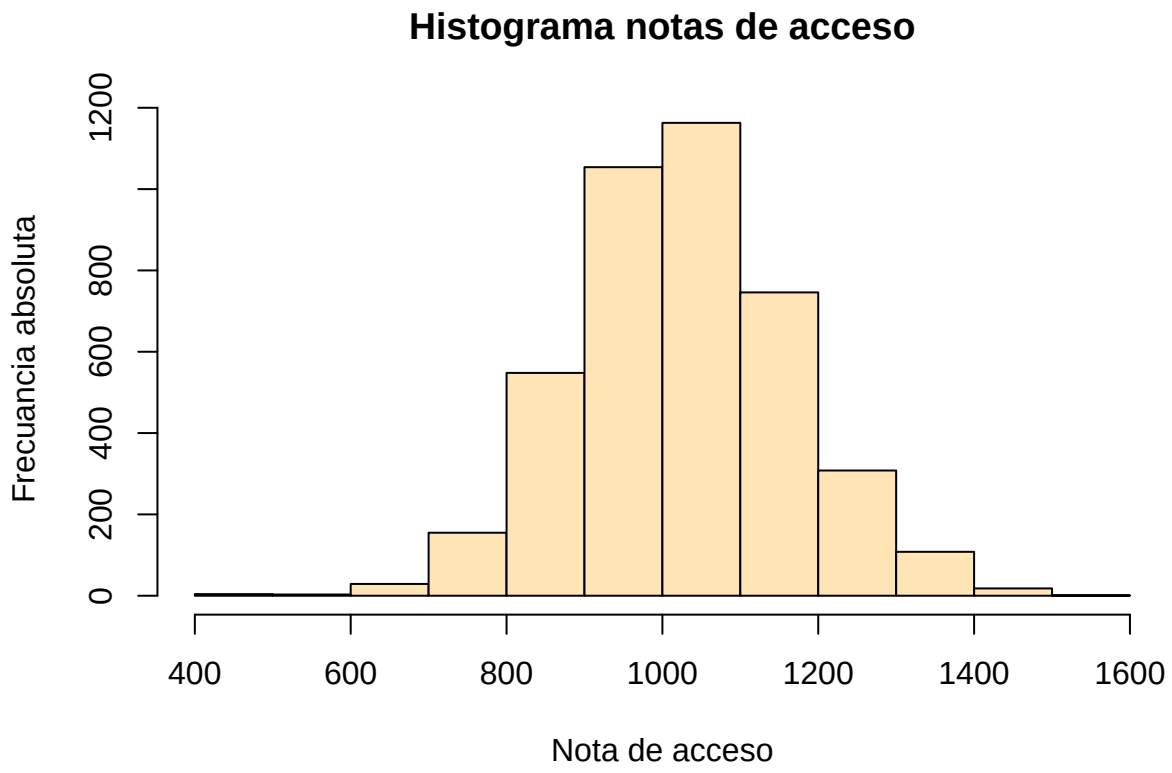
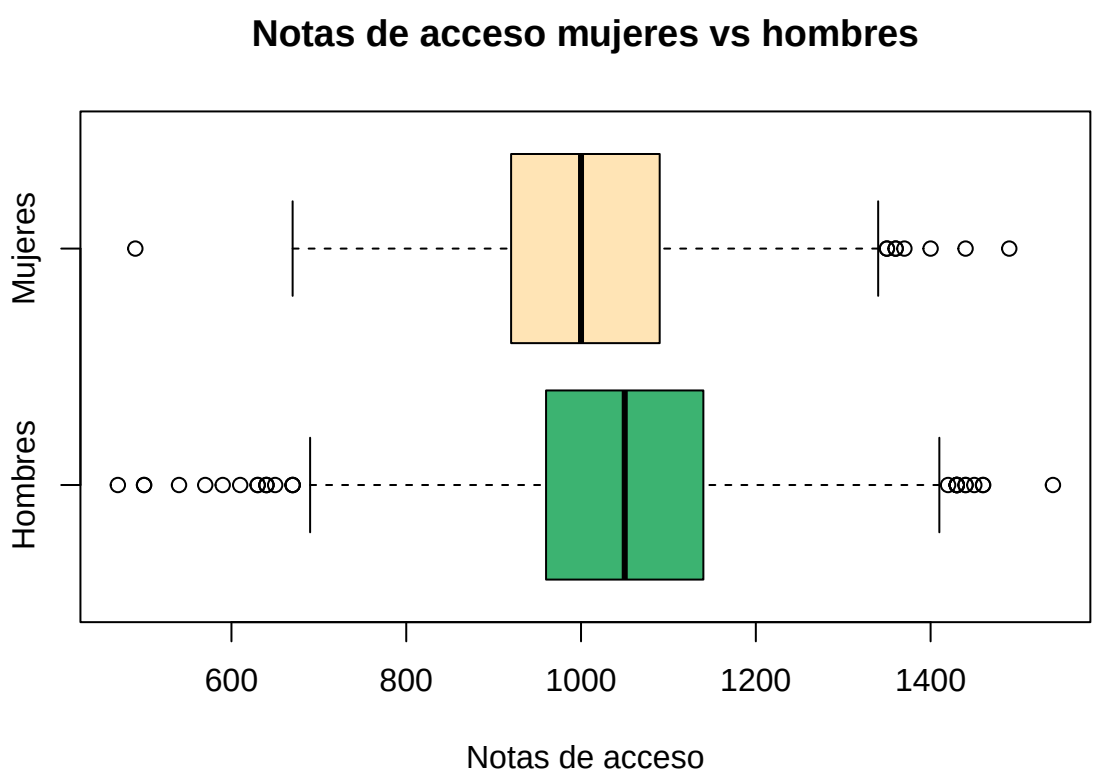


Diagrama de caja:

```
boxplot(my_data$sat~my_data$female,
        main = "Notas de acceso mujeres vs hombres",
        xlab = "Notas de acceso",
        ylab = NULL,
        col=c("mediumseagreen","moccasin"),
        horizontal = T,
        names = c("Hombres","Mujeres") )
```



La variable de las notas de acceso cumple con una distribución normal, tanto en el caso de los hombres como en el de las mujeres.

En el diagrama de cajas observamos que la mediana de las mujeres es ligeramente inferior a la de los hombres. En lo que se refiere a la dispersión de los datos en el caso de los hombres encontramos los datos más dispersos ya que el rango intercuartílico es ligeramente mayor y los bigotes también son más largos. Por el contrario, en el caso de las mujeres los datos se encuentran menos dispersos en torno a una mediana menor. También cabe destacar que los hombres presentan más valores atípicos, sobre todo en el límite inferior.

- **Horas totales cursadas en el semestre, "tothrs"**

Estadísticos de la variable "tothrs":

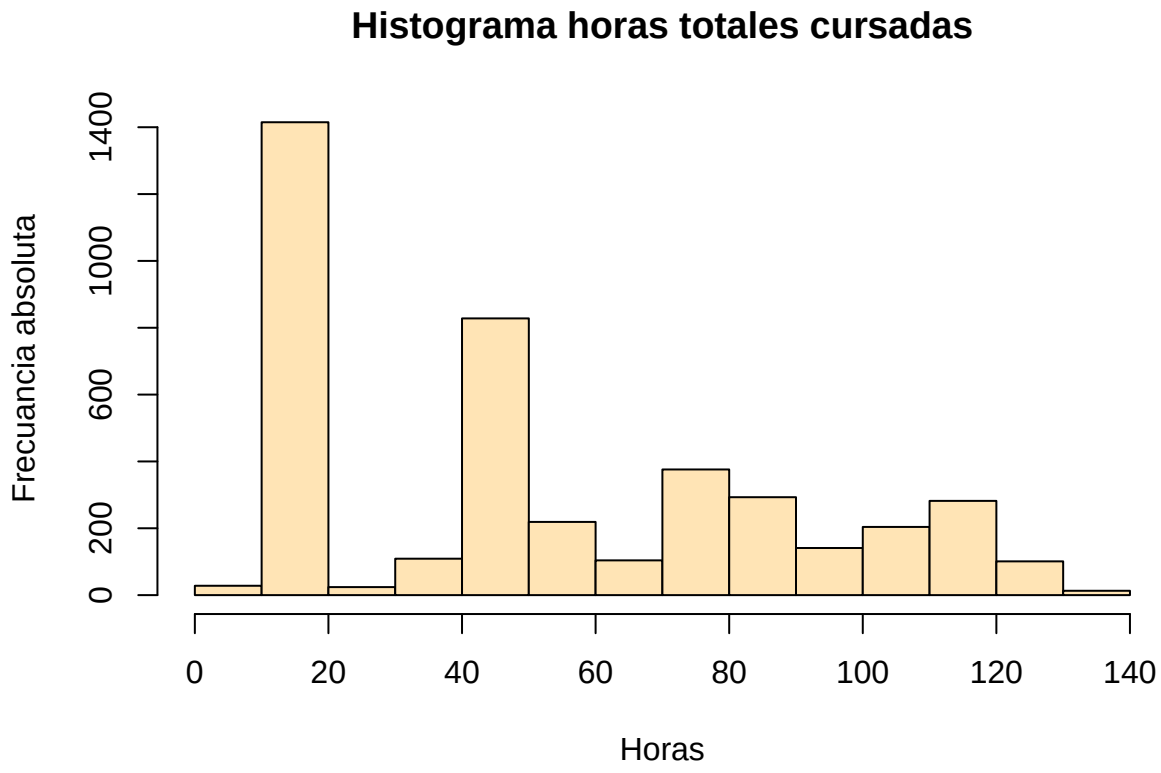
```
df.7.tothrs <- data.frame("TOTHRS" = c(tothrs.mean,tothrs.median,
                                       tothrs.meant,tothrs.sd,tothrs.ric,tothrs.dam),
                  row.names= c("Media aritmética","Mediana","Media recortada",
                               "Desviación estándar","Rango intercuartílico",
                               "Desviación absoluta respecto de la mediana"))

knitr::kable(df.7.tothrs)
```

	TOTHRS
Media aritmética	52.83
Mediana	47.00
Media recortada	51.27

	TOTHRS
Desviación estándar	35.33
Rango intercuartílico	63.00
Desviación absoluta respecto de la mediana	45.96

Histograma:

```
hist(my_data$tothrs,
     main = "Histograma horas totales cursadas",
     col="moccasin",
     xlab = "Horas",
     ylab = "Frecuancia absoluta"
     )
```



Podemos observar que no existe simetría en los datos de la variable. La moda está entre 10 y 20 horas. Encontramos una desviación típica muy alta, que se traduce en que los datos se encuentran muy dispersos alrededor de la media.

- **Número total de estudiantes, "hsize"**

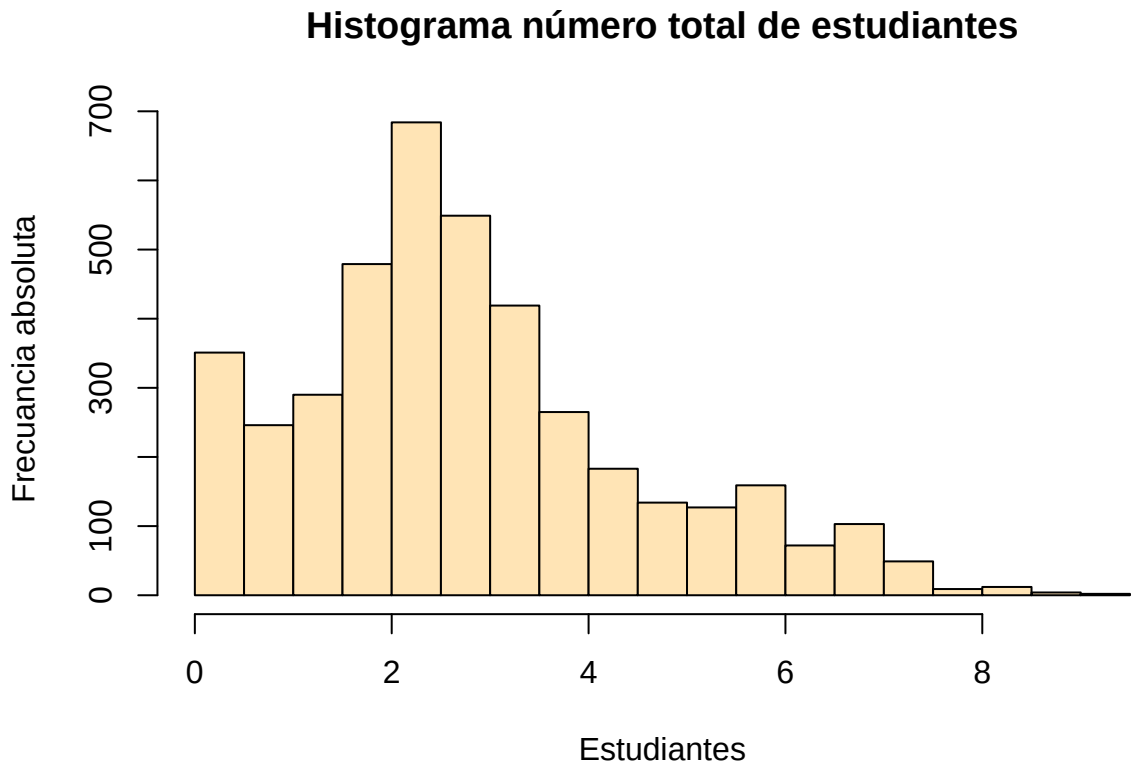
Estadísticos de la variable "hsize":

```
df.7.hsize <- data.frame("HSIZE" = c(hsize.mean,hsize.median,hsize.meant,hsize.sd,
                                     hsize.ric,hsize.dam),
                         row.names= c("Media aritmética","Mediana","Media recortada",
                                      "Desviación estándar","Rango intercuartílico",
                                      "Desviación absoluta respecto de la mediana"))
knitr::kable(df.7.hsize)
```

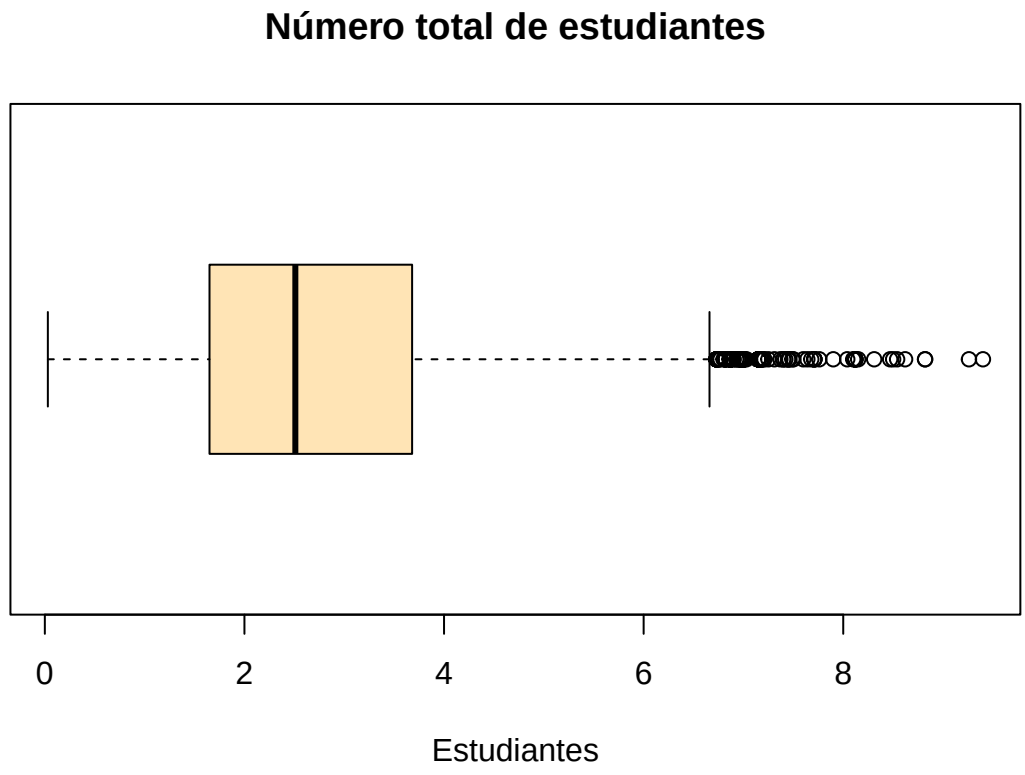
	HSIZE
Media aritmética	2.80
Mediana	2.51
Media recortada	2.71
Desviación estándar	1.74
Rango intercuartílico	2.03
Desviación absoluta respecto de la mediana	1.42

Histograma:

```
hist(my_data$hsize,
     main = "Histograma número total de estudiantes",
     col="moccasin",
     xlab = "Estudiantes",
     ylab = "Frecuancia absoluta"
     )
```



```
boxplot(my_data$hsize,
        main = "Número total de estudiantes",
        xlab = "Estudiantes",
        ylab = NULL,
        col=c("moccasin"),
        horizontal = T)
```



Nos encontramos con bastantes valores atípicos en el límite superior de la variable.

- **Ranking del estudiante, "hsrank"**

Estadísticos de la variable "hsrank":

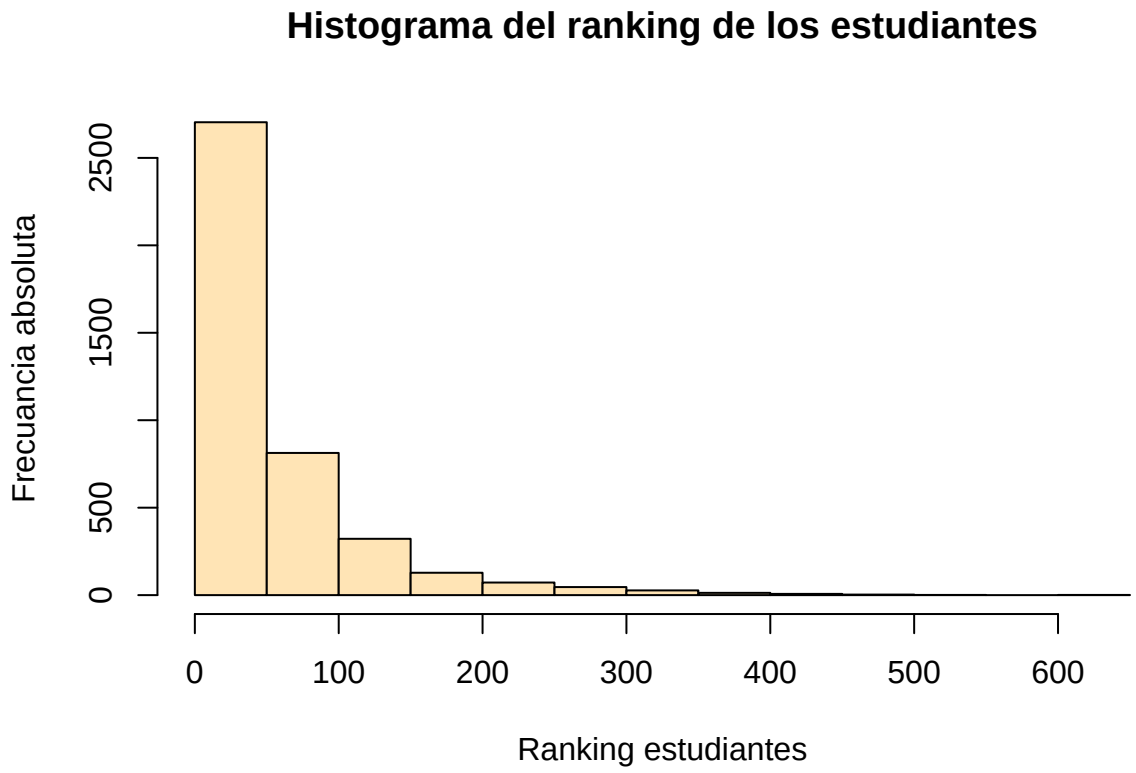
```
df.7.hsrank <- data.frame("HSRANK" = c(hsrank.mean,hsrank.median,hsrank.meant,
                                       hsrank.sd,hsrank.ric,hsrank.dam),
                  row.names= c("Media aritmética","Mediana","Media recortada",
                               "Desviación estándar","Rango intercuartílico",
                               "Desviación absoluta respecto de la mediana"))
knitr::kable(df.7.hsrank)
```

	HSRANK
Media aritmética	52.83
Mediana	30.00
Media recortada	43.99

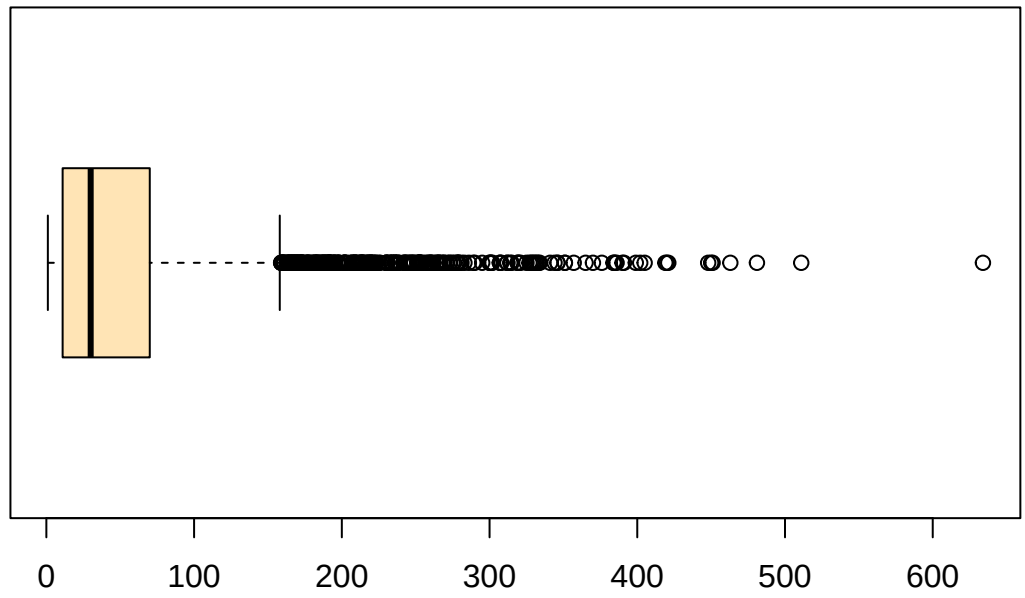
	HSRANK
Desviación estándar	64.68
Rango intercuartílico	59.00
Desviación absoluta respecto de la mediana	35.58

Histograma:

```
hist(my_data$hsrank,
     main = "Histograma del ranking de los estudiantes",
     col="moccasin",
     xlab = "Ranking estudiantes",
     ylab = "Frecuancia absoluta"
     )
```




```
boxplot(my_data$hsrank,
        main = "Ranking de los estudiantes",
        xlab = NULL,
        ylab = NULL,
        col=c("moccasin"),
        horizontal = T)
```



El histograma del ranking del estudiante presenta asimetría hacia la derecha y el diagrama de cajas muestra que los datos atípicos se encuentran en un único extremo.

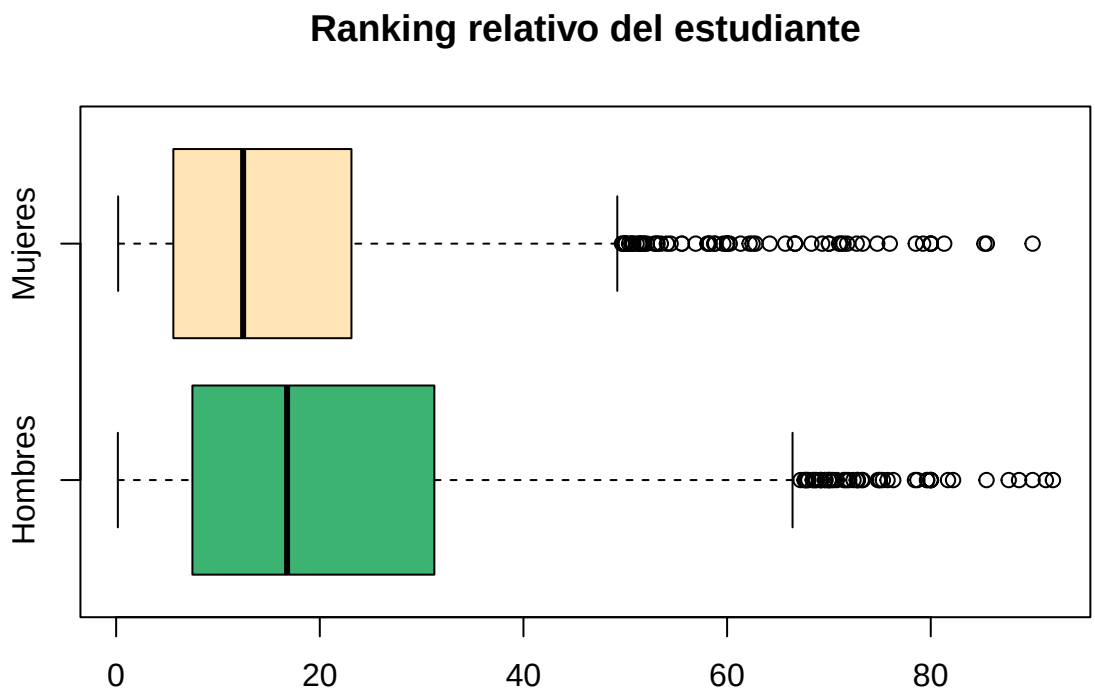
- **Ranking relativo del estudiante, "hsperc"**

Estadísticos de la variable "hsperc":

```
df.7.hsperc <- data.frame("HSPERC" = c(hsperc.mean,hsperc.median,hsperc.meant,
                                       hsperc.sd,hsperc.ric,hsperc.dam),
                  row.names= c("Media aritmética","Mediana","Media recortada",
                               "Desviación estándar","Rango intercuartílico",
                               "Desviación absoluta respecto de la mediana"))
knitr::kable(df.7.hsperc)
```

	HSPERC
Media aritmética	19.24
Mediana	14.58
Media recortada	17.73
Desviación estándar	16.57
Rango intercuartílico	21.28
Desviación absoluta respecto de la mediana	14.05

```
boxplot(my_data$hsperc~my_data$female,
        main = "Ranking relativo del estudiante",
        xlab = NULL,
        ylab = NULL,
        horizontal = T,
        col=c("mediumseagreen","moccasin"),
        names = c("Hombres","Mujeres"))
```



Los hombres presentan unos datos más dispersos que las mujeres y una mediana superior. Ambos presentan valores atípicos en el extremo superior.

8. Archivo final

```
#readr::write_csv(my_data, 'gpa_clean.csv')
```

9. Informe ejecutivo

9.1 Tabla resumen del preprocesamiento

Fases	Observaciones y acciones realizadas
Inicio	o Observaciones: 4137 o Número de variables cuantitativas: 4 o Número de variables cualitativas: 6 o Total de variables: 10
Normalizar variable "athlete"	o Convertir todas las entradas a mayúsculas o Transformar la variable a "factor"
Normalizar variable "female"	o Se ha transformado la variable a "factor"
Normalizar variable "black"	o Convertir todas las entradas a mayúsculas o Borrar los espacios o Transformar la variable a "factor"
Normalizar variable "white"	o Convertir todas las entradas a mayúsculas o Borrar los espacios o Transformar la variable a "factor"
Normalizar variable "tothrs"	o Borrar la unidad, "h", de todas las entradas o Transformar la variable a "numeric"
Normalizar variable "hsize"	o Reemplazar la coma decimal al punto decimal o Transformar la variable a "numeric"
Normalizar variable "hsperc"	o Reemplazar 12 entradas por "hsrank/hsize"
Valores perdidos variable "colgpa"	o En los 41 valores perdidos insertar el resultado dado por la función knn, donde k=11 y teniendo en cuenta que se han estudiado hombres y mujeres por separado
Nueva variable "gpaletter"	o Crear una nueva variable que toma los valores A, B C o D en función de los valores de "colgpa"

Fases	Observaciones y acciones realizadas
Final	o Observaciones: 4137 o Número de variables cuantitativas: 6 o Número de variables cualitativas: 5 o Total de variables: 11

9.2 Resumen estadístico

- **sat: nota de acceso (medida en escala de 400 a 1600 puntos)**

En lo que hace referencia a las notas de acceso los hombres tienen una mediana superior a las mujeres. También cabe destacar que los hombres presentan notas de acceso más dispersas, existen en el caso de los hombres más valores atípicos en el límite inferior.

- **tothrs: horas totales cursadas en el semestre**

No existe ningún tipo de simetría en esta variable. Existen dos grandes rangos de horas en las que se encuentran gran parte de los datos.

- **hsize: número total de estudiantes en la cohorte de graduados del bachillerato (en cientos)**

Existen bastantes valores atípicos en el límite superior de la variable. Los valores presentan cierta simetria con una cola a la derecha.

- **hsrank: ranking del estudiante**

La mayoría de los estudiantes se encuentran en un pequeño rango de valores por lo que existen muchos valores atípicos.

- **hsperc: ranking relativo del estudiante**

Los hombres presentan unos datos más dispersos que las mujeres y una mediana superior. Ambos presentan valores atípicos en el extremo superior.

- **athlete: indicador de si el estudiante practica algún deporte en la universidad**

La gran mayoría de estudiantes no practica ningún deporte y de los pocos que sí que practican hay más hombres que mujeres.