A1 - Preproceso de datos

Autor: Eduardo Mora González

octubre 2022

if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')  
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')

# Carga del archivo

Lo primero que hacemos es cargar el fichero:

library(readr)  
 fichero <- read\_csv("C:/Users/eduar/Dropbox/ESTUDIOS/Estadística avanzada/PEC1/gpa\_row.csv")

## Rows: 4137 Columns: 10

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (2): tothrs, hsize  
## dbl (4): sat, hsrank, hsperc, colgpa  
## lgl (4): athlete, female, white, black

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Una vez cargado el fichero, la salida nos da el numero de variables que hay de cada tipo (numéricas que hay 4, lógicas que hay 4 o texto que hay 2), pero ahora vamos a ver si los datos que tenemos están dentro de los parámetros correspondientes, según se especifica en el enunciado.

summary(fichero)

## sat tothrs hsize hsrank   
## Min. : 470 Length:4137 Length:4137 Min. : 1.00   
## 1st Qu.: 940 Class :character Class :character 1st Qu.: 11.00   
## Median :1030 Mode :character Mode :character Median : 30.00   
## Mean :1030 Mean : 52.83   
## 3rd Qu.:1120 3rd Qu.: 70.00   
## Max. :1540 Max. :634.00   
##   
## hsperc colgpa athlete female   
## Min. : 0.1667 Min. :0.000 Mode :logical Mode :logical   
## 1st Qu.: 6.4328 1st Qu.:2.210 FALSE:3943 FALSE:2277   
## Median :14.5963 Median :2.660 TRUE :194 TRUE :1860   
## Mean :19.2406 Mean :2.655   
## 3rd Qu.:27.7108 3rd Qu.:3.120   
## Max. :92.0000 Max. :4.000   
## NA's :41   
## white black   
## Mode :logical Mode :logical   
## FALSE:308 FALSE:3908   
## TRUE :3829 TRUE :229   
##   
##   
##   
##

Tras ver las estadísticas básicas de las variables, nos damos cuenta de lo siguiente:

**Variables numéricas** –> son: sat, hsrank, hsperc, colgpa. Las variables tothrs y hsize también son de tipo numérica pero R las ha interpretado como Strings.

**Variables lógicas** –> son: athlete, female, white y black. Estas si han sido interpretadas de manera correcta.

# Normalización de las variables cualitativas

## Athlete

Presentamos el contenido de la variable

head(fichero$athlete)

## [1] TRUE FALSE TRUE FALSE FALSE FALSE

Comprobamos el tipo de variable que es

class(fichero$athlete)

## [1] "logical"

Cambiamos a variable factor y lo comprobamos

fichero$athlete<- as.factor(fichero$athlete)  
head(fichero$athlete)

## [1] TRUE FALSE TRUE FALSE FALSE FALSE  
## Levels: FALSE TRUE

class(fichero$athlete)

## [1] "factor"

## Female

Presentamos el contenido de la variable

head(fichero$female)

## [1] TRUE FALSE FALSE FALSE FALSE TRUE

Comprobamos el tipo de variable que es

class(fichero$female)

## [1] "logical"

Cambiamos a variable factor y lo comprobamos

fichero$female<- as.factor(fichero$female)  
head(fichero$female)

## [1] TRUE FALSE FALSE FALSE FALSE TRUE   
## Levels: FALSE TRUE

class(fichero$female)

## [1] "factor"

## Black

Presentamos el contenido de la variable

head(fichero$black)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE

Comprobamos el tipo de variable que es

class(fichero$black)

## [1] "logical"

Cambiamos a variable factor y lo comprobamos

fichero$black<- as.factor(fichero$black)  
head(fichero$black)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE  
## Levels: FALSE TRUE

class(fichero$black)

## [1] "factor"

## White

Presentamos el contenido de la variable

head(fichero$white)

## [1] FALSE TRUE TRUE TRUE TRUE TRUE

Comprobamos el tipo de variable que es

class(fichero$white)

## [1] "logical"

Cambiamos a variable factor y lo comprobamos

fichero$white<- as.factor(fichero$white)  
head(fichero$white)

## [1] FALSE TRUE TRUE TRUE TRUE TRUE   
## Levels: FALSE TRUE

class(fichero$white)

## [1] "factor"

# Normalización de las variables cuantitativas

## Nota de acceso

Presentamos el contenido de la variable

head(fichero$sat)

## [1] 920 1170 810 940 1180 980

Comprobamos el tipo de variable que es

class(fichero$sat)

## [1] "numeric"

La variable se adapta a la normalización deseada.

## Horas totales cursadas al semestre

Presentamos el contenido de la variable

head(fichero$tothrs)

## [1] "43h" "18h" "14h" "40h" "18h" "114h"

Comprobamos el tipo de variable que es

class(fichero$tothrs)

## [1] "character"

Debemos quitar la “h” final y convertir la variable en numérica

library(stringr)  
  
fichero$tothrs <- str\_replace (fichero$tothrs, "h", "")  
head(fichero$tothrs)

## [1] "43" "18" "14" "40" "18" "114"

fichero$tothrs <- as.numeric(fichero$tothrs)

Comprobamos el tipo de variable que es

class(fichero$tothrs)

## [1] "numeric"

Ahora la variable se adapta a la normalización deseada.

## Nota media del estudiante al final del primer semestre

Presentamos el contenido de la variable

head(fichero$colgpa)

## [1] 2.04 4.00 1.78 2.42 2.61 3.03

Comprobamos el tipo de variable que es

class(fichero$colgpa)

## [1] "numeric"

La variable se adapta a la normalización deseada.

## Número total de estudiants en la cohorte de graduados del bachillerato

Presentamos el contenido de la variable

head(fichero$hsize)

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

Comprobamos el tipo de variable que es

class(fichero$hsize)

## [1] "character"

Debemos convertir la variable en numérica

library(stringr)  
  
fichero$hsize <- str\_replace (fichero$hsize, ",", ".")  
head(fichero$hsize)

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

fichero$hsize <- as.numeric(fichero$hsize)

Comprobamos el tipo de variable que es

class(fichero$hsize)

## [1] "numeric"

Ahora la variable se adapta a la normalización deseada.

## Ranking relativo del estudiante

Presentamos el contenido de la variable

head(fichero$hsperc)

## [1] 40.00000 20.31915 35.29412 44.13310 40.18692 15.29851

Comprobamos el tipo de variable que es

class(fichero$hsperc)

## [1] "numeric"

Truncamos los decimales que tienen a solo 3:

fichero$hsperc <- signif(fichero$hsperc, digits =3)

La variable se adapta a la normalización deseada.

# Valores atípicos

Vemos una vez normalizadas las variables, en que rango se encuentran y si están dentro de lo deseado.

summary(fichero)

## sat tothrs hsize hsrank   
## Min. : 470 Min. : 6.00 Min. :0.03 Min. : 1.00   
## 1st Qu.: 940 1st Qu.: 17.00 1st Qu.:1.65 1st Qu.: 11.00   
## Median :1030 Median : 47.00 Median :2.51 Median : 30.00   
## Mean :1030 Mean : 52.83 Mean :2.80 Mean : 52.83   
## 3rd Qu.:1120 3rd Qu.: 80.00 3rd Qu.:3.68 3rd Qu.: 70.00   
## Max. :1540 Max. :137.00 Max. :9.40 Max. :634.00   
##   
## hsperc colgpa athlete female white   
## Min. : 0.167 Min. :0.000 FALSE:3943 FALSE:2277 FALSE: 308   
## 1st Qu.: 6.430 1st Qu.:2.210 TRUE : 194 TRUE :1860 TRUE :3829   
## Median :14.600 Median :2.660   
## Mean :19.241 Mean :2.655   
## 3rd Qu.:27.700 3rd Qu.:3.120   
## Max. :92.000 Max. :4.000   
## NA's :41   
## black   
## FALSE:3908   
## TRUE : 229   
##   
##   
##   
##   
##

La variable **tothrs**` y **hsrank** está dentro de los parámetros normales.

La variable **colgpa** está en escala de 0 a 4 puntos, por lo que está de forma correcta. Además esta variable tiene 41 campos nulos.

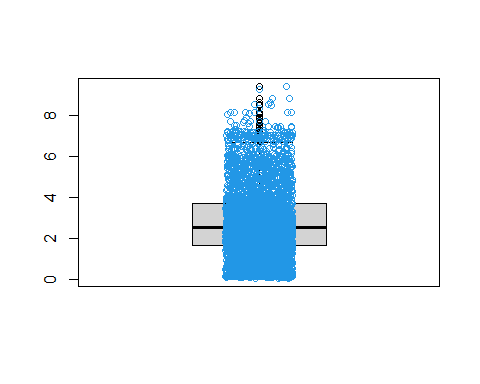
Para la variable **hsperc** se va a comprobar si el cálculo se ha hecho de manera correcta, para ello se comprueba si son iguales los valores, si no se sustituye, y se cuenta los distintos. Finalmente se muestra el número de errores que ha tenido los datos.

distintos = 0  
  
for(i in 1: length(fichero$hsperc)){  
   
 calculo <- signif((fichero$hsrank[i] / fichero$hsize[i]), digits =3)  
   
 iguales <- identical(calculo, fichero$hsperc[i])  
   
   
 if(iguales==FALSE){  
 fichero$hsperc[i]<-calculo  
 distintos = distintos + 1  
 }  
   
}  
  
distintos

## [1] 12

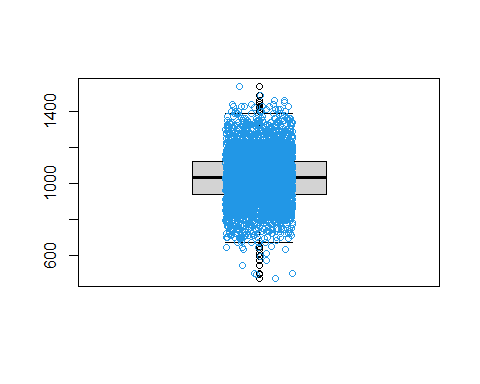
Para la variable **hsize** nos damos cuenta de que esta dentro de los rangos, pero tiene valores atípicos, como se puede comprobar en el boxplot siguiente:

boxplot(fichero$hsize)  
  
# Points  
stripchart(fichero$hsize, # Data  
 method = "jitter", # Random noise  
 pch = 1, # Pch symbols  
 col = 4, # Color of the symbol  
 vertical = TRUE, # Vertical mode  
 add = TRUE) # Add it over



La variable **SAT** está en escala de 400 a 1600 puntos, pero tiene valores atípicos, como se puede comprobar en el boxplot siguiente:

boxplot(fichero$sat)  
  
# Points  
stripchart(fichero$sat, # Data  
 method = "jitter", # Random noise  
 pch = 1, # Pch symbols  
 col = 4, # Color of the symbol  
 vertical = TRUE, # Vertical mode  
 add = TRUE) # Add it over



# Imputación de valores

Lo primero que vamos a hacer es dividir los registros entre hombre y mujer

fichero\_mujer <- fichero  
fichero\_mujer <- fichero\_mujer [fichero\_mujer$female == TRUE ,]  
fichero\_hombre <- fichero  
fichero\_hombre <- fichero\_hombre[fichero\_hombre$female == FALSE ,]  
  
summary(fichero\_hombre$colgpa)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 2.12 2.57 2.59 3.06 4.00 24

summary(fichero\_mujer$colgpa)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.210 2.325 2.740 2.734 3.170 4.000 17

Comprobamos que para los hombres hay 24 nulos y para las mujeres 17.

Hacemos la imputación para las mujeres, lo mostramos y los añadimos a la lista de mujeres

library(VIM)

## Warning: package 'VIM' was built under R version 4.1.3

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':  
##   
## sleep

imputacion\_mujer <- kNN(fichero\_mujer, k=11)  
  
imputacion\_mujer$colgpa[imputacion\_mujer$colgpa\_imp=='TRUE']

## [1] 3.26 2.60 2.82 2.81 2.89 2.94 2.59 2.78 2.36 2.19 2.72 2.75 2.38 2.73 2.11  
## [16] 2.41 2.46

fichero\_mujer$colgpa <- imputacion\_mujer$colgpa  
  
summary(fichero\_mujer$colgpa)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.210 2.330 2.740 2.733 3.170 4.000

Hacemos la imputación para los hombres, lo mostramos y los añadimos a la lista de hombres

library(VIM)  
imputacion\_hombre <- kNN(fichero\_hombre, k=11)  
  
imputacion\_hombre$colgpa[imputacion\_hombre$colgpa\_imp=='TRUE']

## [1] 2.47 2.65 2.37 2.18 2.15 2.26 2.72 2.26 2.43 2.81 2.69 3.46 2.70 2.20 2.50  
## [16] 2.68 2.35 3.00 2.42 2.26 3.23 2.25 3.41 1.68

fichero\_hombre$colgpa <- imputacion\_hombre$colgpa  
  
summary(fichero\_hombre$colgpa)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.120 2.570 2.589 3.060 4.000

Unimos las dos listas con los datos ya imputados.

fichero\_2 <- rbind(fichero\_hombre, fichero\_mujer)  
summary(fichero\_2)

## sat tothrs hsize hsrank   
## Min. : 470 Min. : 6.00 Min. :0.03 Min. : 1.00   
## 1st Qu.: 940 1st Qu.: 17.00 1st Qu.:1.65 1st Qu.: 11.00   
## Median :1030 Median : 47.00 Median :2.51 Median : 30.00   
## Mean :1030 Mean : 52.83 Mean :2.80 Mean : 52.83   
## 3rd Qu.:1120 3rd Qu.: 80.00 3rd Qu.:3.68 3rd Qu.: 70.00   
## Max. :1540 Max. :137.00 Max. :9.40 Max. :634.00   
## hsperc colgpa athlete female white   
## Min. : 0.167 Min. :0.000 FALSE:3943 FALSE:2277 FALSE: 308   
## 1st Qu.: 6.430 1st Qu.:2.210 TRUE : 194 TRUE :1860 TRUE :3829   
## Median :14.600 Median :2.660   
## Mean :19.238 Mean :2.654   
## 3rd Qu.:27.700 3rd Qu.:3.120   
## Max. :92.000 Max. :4.000   
## black   
## FALSE:3908   
## TRUE : 229   
##   
##   
##   
##

# Creación de una nueva variable

Con la calificación dada, se asigna el valor de la variable categórica. Se ha establecido los intervalos para cada letra según el enunciado.

fichero\_2["gpaletter"] <- cut(fichero\_2$colgpa, breaks = c(-0.01,1.49,2.49,3.49,4), labels = c("D", "C", "B", "A"))  
  
head(fichero\_2)

## # A tibble: 6 x 11  
## sat tothrs hsize hsrank hsperc colgpa athlete female white black gpaletter  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <fct> <fct>   
## 1 1170 18 9.40 191 20.3 4 FALSE FALSE TRUE FALSE A   
## 2 810 14 1.19 42 35.3 1.78 TRUE FALSE TRUE FALSE C   
## 3 940 40 5.71 252 44.1 2.42 FALSE FALSE TRUE FALSE C   
## 4 1180 18 2.14 86 40.2 2.61 FALSE FALSE TRUE FALSE B   
## 5 880 78 3.11 161 51.8 1.84 FALSE FALSE FALSE FALSE C   
## 6 980 55 2.68 101 37.7 3.05 FALSE FALSE TRUE FALSE B

tail(fichero\_2)

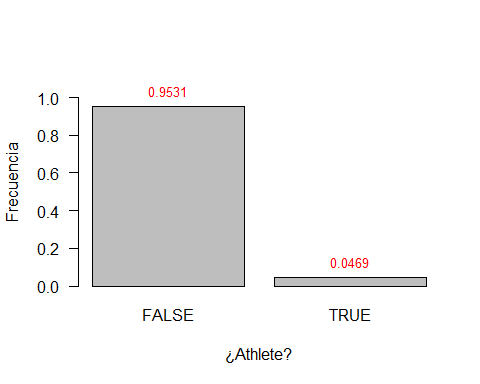
## # A tibble: 6 x 11  
## sat tothrs hsize hsrank hsperc colgpa athlete female white black gpaletter  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <fct> <fct>   
## 1 1000 18 1.41 25 17.7 2.33 FALSE TRUE TRUE FALSE C   
## 2 1020 75 1 41 41 1.97 FALSE TRUE TRUE FALSE C   
## 3 1180 47 2.3 20 8.7 3.36 FALSE TRUE TRUE FALSE B   
## 4 990 49 2.33 89 38.2 2.24 FALSE TRUE TRUE FALSE C   
## 5 900 50 0.1 2 20 2.46 FALSE TRUE TRUE FALSE C   
## 6 980 12 0.350 23 65.7 2.83 FALSE TRUE TRUE FALSE B

# Estudio descriptivo

## Estudio descriptivo de las variables cualitativas

### Athlete en porcentaje de atletas

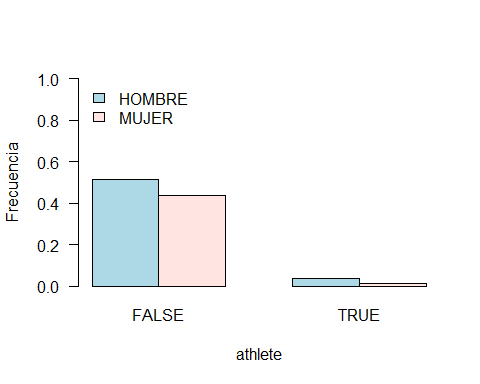
tabla1 <- table(fichero\_2$athlete)  
tabla1 <- prop.table(tabla1)  
  
xx <- barplot(tabla1, ylim=c(0,1.1), xlab='¿Athlete?', ylab='Frecuencia', las=1)  
text(x=xx, y=tabla1, pos=3, cex=0.8, col="red", label=round(tabla1, 4))



Como se puede ver, hay un mayor número de personas que no practican deporte respecto a las que si las practican.

### Athlete en porcentaje de atletas en función del sexo

tabla2 <- table(fichero\_2$female, fichero\_2$athlete)  
tabla2 <- prop.table(tabla2)  
valores <- c("HOMBRE", "MUJER")  
  
barplot(tabla2, beside = TRUE, las=1,   
 xlab='athlete', ylab='Frecuencia',  
 col = c("lightblue", "mistyrose"),  
 ylim = c(0,1))  
legend('topleft', legend=valores, bty='n',  
 fill=c("lightblue", "mistyrose"))

 A nivel visual se puede ver, que los hombres hacen mas deporte que las mujeres.

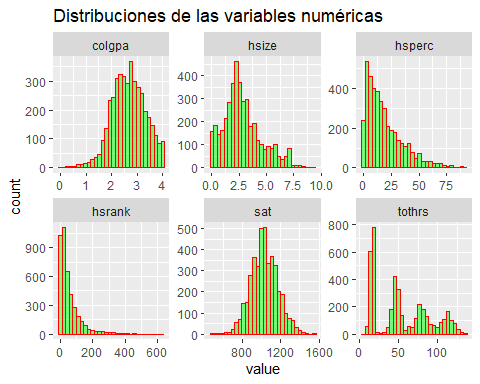
## Estudio descriptivo de las variables cuantitativas

### Estudio descriptivo de las variables cuantitativas “sat”, “tothrs”, “hsize”, “hsrank”.

Se va a generar histogramas para verificar la distribución de las variables numéricas

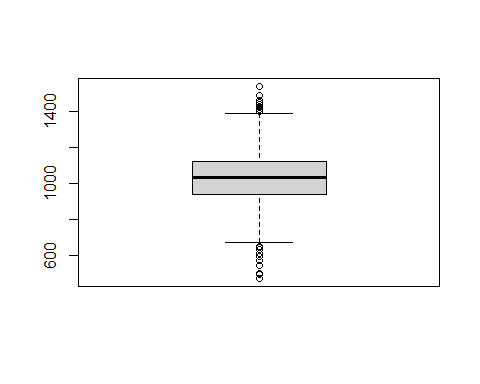
library(purrr)  
library(tidyr)  
library(ggplot2)  
  
fichero\_2 %>%  
 keep(is.numeric) %>%   
 gather() %>%   
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram(col="red",  
 fill="green",  
 alpha = 0.5,) +  
 ggtitle("Distribuciones de las variables numéricas")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

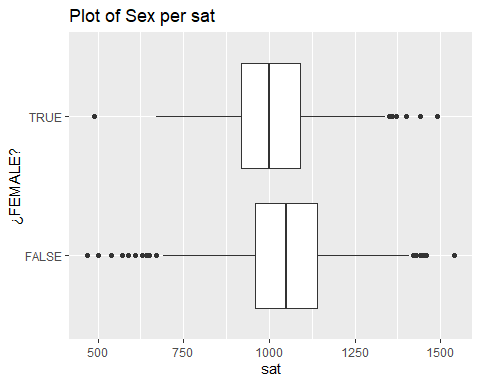


### Distribución de los valores de “sat”

boxplot(fichero\_2$sat)

 ### Distribución de los valores de “sat” por sexo

ggplot(fichero\_2, aes(x=sat, y=female, fill=sat)) +   
 geom\_boxplot()+  
 labs(title="Plot of Sex per sat",x="sat", y = "¿FEMALE?")



### Tabla con diversas medidas de tendencia central y dispersión, robustas y no robustas.

Cabecera <- c("Estimadores","sat", "tothrs", "hsize", "hsrank" )  
sat <- c(mean(fichero\_2$sat),median(fichero\_2$sat),mean(fichero\_2$sat,trim=0.05),sd(fichero\_2$sat),IQR(fichero\_2$sat), mad(fichero\_2$sat) )  
tothrs <- c(mean(fichero\_2$tothrs),median(fichero\_2$tothrs),mean(fichero\_2$tothrs,trim=0.05),sd(fichero\_2$tothrs),IQR(fichero\_2$tothrs), mad(fichero\_2$tothrs) )  
hsize<- c(mean(fichero\_2$hsize),median(fichero\_2$hsize),mean(fichero\_2$hsize,trim=0.05),sd(fichero\_2$hsize),IQR(fichero\_2$hsize), mad(fichero\_2$hsize) )  
hsrank<- c(mean(fichero\_2$hsrank),median(fichero\_2$hsrank),mean(fichero\_2$hsrank,trim=0.05),sd(fichero\_2$hsrank),IQR(fichero\_2$hsrank), mad(fichero\_2$hsrank) )  
Estimadores <- c("Media", "Mediana","Media recortada","Desviación estándar","RIC","DAM")  
  
tabla\_medidas <- data.frame(  
 "Estimadores" = Estimadores,   
 "sat" = sat,  
 "tothrs" = tothrs,  
 "hsize" = hsize,  
 "hsrank" = hsrank  
)  
  
tabla\_medidas

## Estimadores sat tothrs hsize hsrank  
## 1 Media 1030.3312 52.83225 2.799727 52.83007  
## 2 Mediana 1030.0000 47.00000 2.510000 30.00000  
## 3 Media recortada 1029.4792 51.27060 2.711791 43.99383  
## 4 Desviación estándar 139.4014 35.32959 1.736579 64.68358  
## 5 RIC 180.0000 63.00000 2.030000 59.00000  
## 6 DAM 133.4340 45.96060 1.423296 35.58240

# Archivo final

head(fichero\_2)

## # A tibble: 6 x 11  
## sat tothrs hsize hsrank hsperc colgpa athlete female white black gpaletter  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <fct> <fct>   
## 1 1170 18 9.40 191 20.3 4 FALSE FALSE TRUE FALSE A   
## 2 810 14 1.19 42 35.3 1.78 TRUE FALSE TRUE FALSE C   
## 3 940 40 5.71 252 44.1 2.42 FALSE FALSE TRUE FALSE C   
## 4 1180 18 2.14 86 40.2 2.61 FALSE FALSE TRUE FALSE B   
## 5 880 78 3.11 161 51.8 1.84 FALSE FALSE FALSE FALSE C   
## 6 980 55 2.68 101 37.7 3.05 FALSE FALSE TRUE FALSE B

write.csv(fichero\_2, file = "gpa\_clean.csv")

# Informe ejecutivo

## Tabla resumen del preprocesamiento

| Línea de Código | Observaciones |
| --- | --- |
|  | n. Observaciones = |
| 2 | 3 |

## Resumen estadístico