Preprocesado

Angel Caballero Domimguez

29/12/2021

Preprocesado

0. Paquetes de R

```
# Instalamos e iniamos el paquete tidyverse
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Instalamos e iniamos el paquete mice
#install.packages("mice")
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

Se utilizan los paquetes tidyverse y mice. Este último nos será de utilidad durante la limpieza de nuestros datos. Por lo tanto, el resto de funciones están o creadas para este caso o incluidas en R básico.

1. Integración

1.1. Lectura de datos

Extramos el data frame del archivo .csv:

```
df<-read.csv("./data/StudentsPerformance.csv",stringsAsFactors = TRUE
             , na.strings = "", header = TRUE)
```

1.2. Dimensiones y tipo de datos

Veamos las dimensiones del data frame:

```
dim(df)
```

```
## [1] 1000    8
```

Las dimensiones del data frame básico sería de 1000 filas y 8 columnas.

```
str(df)
```

```
## 'data.frame':    1000 obs. of  8 variables:
##  $ gender                     : Factor w/ 2 levels "female","male": 1 1 1 2 2 1 1 2 2 1 ...
##  $ race.ethnicity             : Factor w/ 5 levels "group A","group B",..: 2 3 2 1 3 2 2 2 4 2 ...
##  $ parental.level.of.education: Factor w/ 6 levels "associate's degree",..: 2 5 4 1 5 1 5 5 3 3 ...
##  $ lunch                      : Factor w/ 2 levels "free/reduced",..: 2 2 2 1 2 2 2 1 1 1 ...
##  $ test.preparation.course    : Factor w/ 2 levels "completed","none": 2 1 2 2 2 2 1 2 1 2 ...
##  $ math.score                 : int  72 69 90 47 76 71 88 40 64 38 ...
##  $ reading.score              : int  72 90 95 57 78 83 95 43 64 60 ...
##  $ writing.score              : int  74 88 93 44 75 78 92 39 67 50 ...
```

Como se puede observar, en 5 de las columnas los valores son factores y en 3 de ellas, son valores enteros.

1.3. Transformación de los nombres de las columnas

Debido a que los nombres de algunas columnas son demasiado largas, vamos a cambiarlos para que sean más accesibles los datos:

```
data <- df
colnames(data) <- c('Gender','Ethnicity','Parent_Education','Lunch','Preparation',
                    'Math','Reading','Writing')
```

1.4. Visión general de los datos

Vamos a ver cuáles son los niveles de cada uno de los factores:

```
dataLevels <- sapply(data[,1:5], levels)
dataLevels
```

```
## $Gender
## [1] "female" "male"
##
## $Ethnicity
## [1] "group A" "group B" "group C" "group D" "group E"
##
## $Parent_Education
## [1] "associate's degree" "bachelor's degree"  "high school"
## [4] "master's degree"    "some college"       "some high school"
##
## $Lunch
## [1] "free/reduced" "standard"
##
## $Preparation
## [1] "completed" "none"
```

A continuación, vamos visualizar los fragmentos inicial y final de la tabla:

```
knitr::kable(head(data,n=5))
```

Gender	Ethnicity	Parent_Education	Lunch	Preparation	Math	Reading	Writing
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75

```
knitr::kable(tail(data,n=5))
```

	Gender	Ethnicity	Parent_Education	Lunch	Preparation	Math	Reading	Writing
996	female	group E	master's degree	standard	completed	88	99	95
997	male	group C	high school	free/reduced	none	62	55	55
998	female	group C	high school	free/reduced	completed	59	71	65
999	female	group D	some college	standard	completed	68	78	77
1000	female	group D	some college	free/reduced	none	77	86	86

Por último, vamos a ver un resumen del dataframe:

```
summary(data)
```

```
##     Gender      Ethnicity             Parent_Education          Lunch
##  female:518   group A: 89   associate's degree:222     free/reduced:355
##  male  :482   group B:190   bachelor's degree :118     standard    :645
##               group C:319   high school       :196
##               group D:262   master's degree   : 59
##               group E:140   some college      :226
##                             some high school  :179
##     Preparation       Math           Reading          Writing
##  completed:358   Min.   :  0.00   Min.   : 17.00   Min.   : 10.00
```

```
##  none     :642   1st Qu.: 57.00   1st Qu.: 59.00   1st Qu.: 57.75
##                  Median : 66.00   Median : 70.00   Median : 69.00
##                  Mean   : 66.09   Mean   : 69.17   Mean   : 68.05
##                  3rd Qu.: 77.00   3rd Qu.: 79.00   3rd Qu.: 79.00
##                  Max.   :100.00   Max.   :100.00   Max.   :100.00
```

Como se puede comprobar de la información recibida, en principio no habría ninguna fila a la que le falte un valor en alguna columna de factor.

Antes de terminar con este apartado, vamos a guardar nuestro data frame y los niveles de los factores en archivos RData:

```
save(data, file = "data.RData")
save(dataLevels, file = "dataLevels.RData")
```

2. Visualización

2.1. Factores

Vamos a comenzar la visualización con los factores.

```
# Porcentaje por género
pct_gender <- data %>%
  summarise(
    pct_female = 100*(sum(data$Gender==dataLevels$Gender[1])/dim(data)[1]),
    pct_male = 100*(sum(data$Gender==dataLevels$Gender[2])/dim(data)[1]),
    .groups = "drop"
  )

rownames(pct_gender)<-c("Porcentajes")
colnames(pct_gender)<-c("Mujeres", "Hombres")

pct_gender <- as.data.frame(pct_gender)

knitr::kable(pct_gender)
```

	Mujeres	Hombres
Porcentajes	51.8	48.2

```
# Porcentaje por etnia
pct_ethnicity <- data %>%
  summarise(
    pct_a = 100*(sum(data$Ethnicity == dataLevels$Ethnicity[1])/dim(data)[1]),
    pct_b = 100*(sum(data$Ethnicity == dataLevels$Ethnicity[2])/dim(data)[1]),
    pct_c = 100*(sum(data$Ethnicity == dataLevels$Ethnicity[3])/dim(data)[1]),
    pct_d = 100*(sum(data$Ethnicity == dataLevels$Ethnicity[4])/dim(data)[1]),
    pct_e = 100*(sum(data$Ethnicity == dataLevels$Ethnicity[5])/dim(data)[1]),
    .groups = "drop"
  )

rownames(pct_ethnicity)<-c("Porcentajes")
```

```r
colnames(pct_ethnicity)<-c("Group A","Group B","Group C","Group D","Group E")

pct_ethnicity <- as.data.frame(pct_ethnicity)

knitr::kable(pct_ethnicity)
```

	Group A	Group B	Group C	Group D	Group E
Porcentajes	8.9	19	31.9	26.2	14

```r
# Porcentaje por educación de los padres
pct_ed <- data %>%
  summarise(
    pct_a = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[1])
                 /dim(data)[1]),
    pct_b = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[2])
                 /dim(data)[1]),
    pct_c = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[3])
                 /dim(data)[1]),
    pct_d = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[4])
                 /dim(data)[1]),
    pct_e = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[5])
                 /dim(data)[1]),
    pct_f = 100*(sum(data$Parent_Education == dataLevels$Parent_Education[6])
                 /dim(data)[1]),
    .groups = "drop"
  )

rownames(pct_ed)<-c("Porcentajes")
colnames(pct_ed)<-c("Associate's degree","Bachelor's degree","High school"
                    ,"Master's degree","Some college","Some high school")

pct_ed <- as.data.frame(pct_ed)

knitr::kable(pct_ed)
```

	Associate's degree	Bachelor's degree	High school	Master's degree	Some college	Some high school
Porcentajes	22.2	11.8	19.6	5.9	22.6	17.9

```r
# Porcentaje por almuerzo
pct_lunch <- data %>%
  summarise(
    pct_free = 100*(sum(data$Lunch == dataLevels$Lunch[1])/dim(data)[1]),
    pct_standard = 100*(sum(data$Lunch == dataLevels$Lunch[2])/dim(data)[1]),
    .groups = "drop"
  )

rownames(pct_lunch)<-c("Porcentajes")
colnames(pct_lunch)<-c("Free/Reduced","Standard")
```

```r
pct_lunch <- as.data.frame(pct_lunch)

knitr::kable(pct_lunch)
```

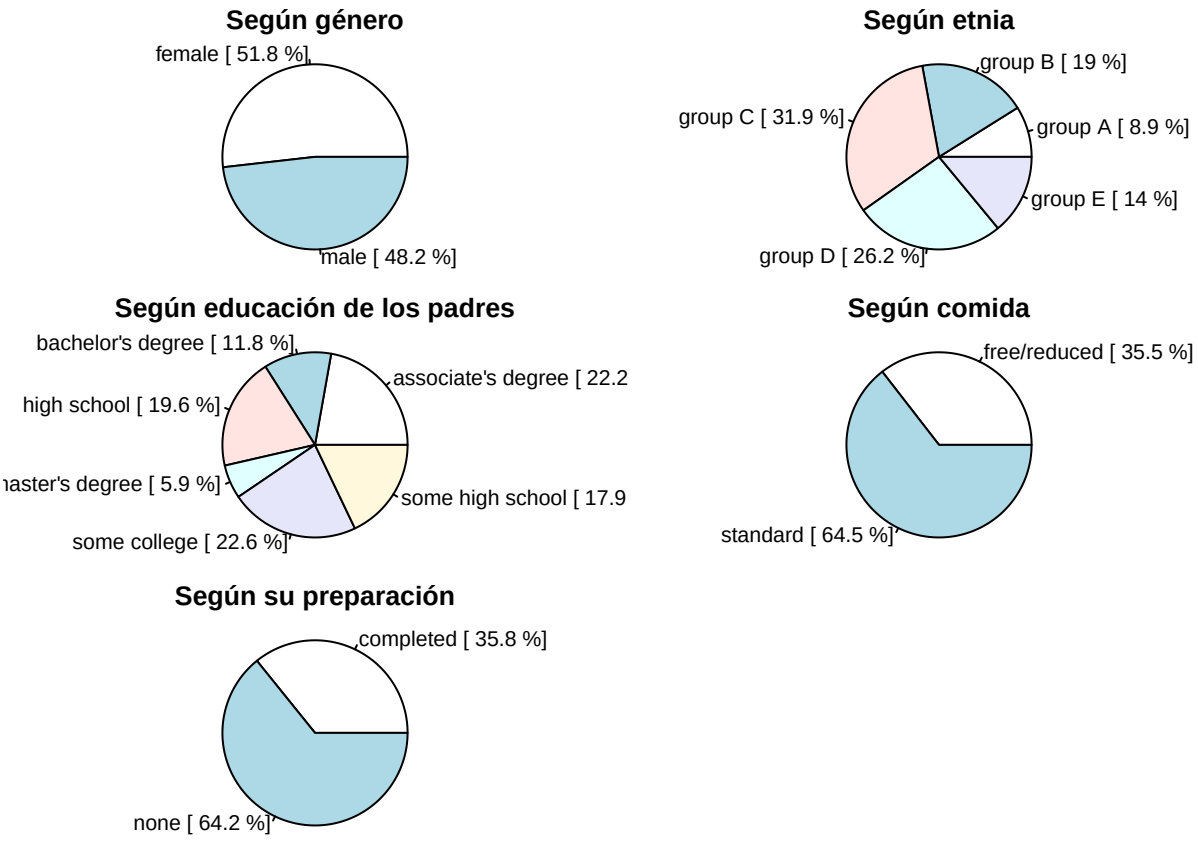
	Free/Reduced	Standard
Porcentajes	35.5	64.5

```r
# Porcentaje por preparación
pct_prep <- data %>%
  summarise(
    pct_ninguna = 100*(sum(data$Preparation == dataLevels$Preparation[1])
                       /dim(data)[1]),
    pct_completa = 100*(sum(data$Preparation == dataLevels$Preparation[2])
                        /dim(data)[1]),
    .groups = "drop"
  )

rownames(pct_prep)<-c("Porcentajes")
colnames(pct_prep)<-c("Completed","None")

pct_prep <- as.data.frame(pct_prep)

knitr::kable(pct_prep)
```

	Completed	None
Porcentajes	35.8	64.2

```r
# Modificación de los parámetros gráficos
par(mfrow = c(3,2),
    mar = c(0.5,0,1,0))

# Gráficos de los factores según porcentaje
pie(c(pct_gender[1,1], pct_gender[1,2]), labels = paste(dataLevels$Gender,
                                                        sep = " ", "[", pct_gender,
                                                        "%]"), main = "Según género")

pie(c(pct_ethnicity[1,1], pct_ethnicity[1,2],pct_ethnicity[1,3], pct_ethnicity[1,4],
      pct_ethnicity[1,5]), labels = paste(dataLevels$Ethnicity, sep = " ", "[",
                                          pct_ethnicity,
                                          "%]"), main = "Según etnia")

pie(c(pct_ed[1,1], pct_ed[1,2],pct_ed[1,3], pct_ed[1,4],pct_ed[1,5], pct_ed[1,6]),
    labels = paste(dataLevels$Parent_Education, sep = " ", "[", pct_ed, "%]"),
    main = "Según educación de los padres")

pie(c(pct_lunch[1,1], pct_lunch[1,2]), labels = paste(dataLevels$Lunch, sep = " ",
                                                      "[", pct_lunch, "%]"),
    main = "Según comida")
```

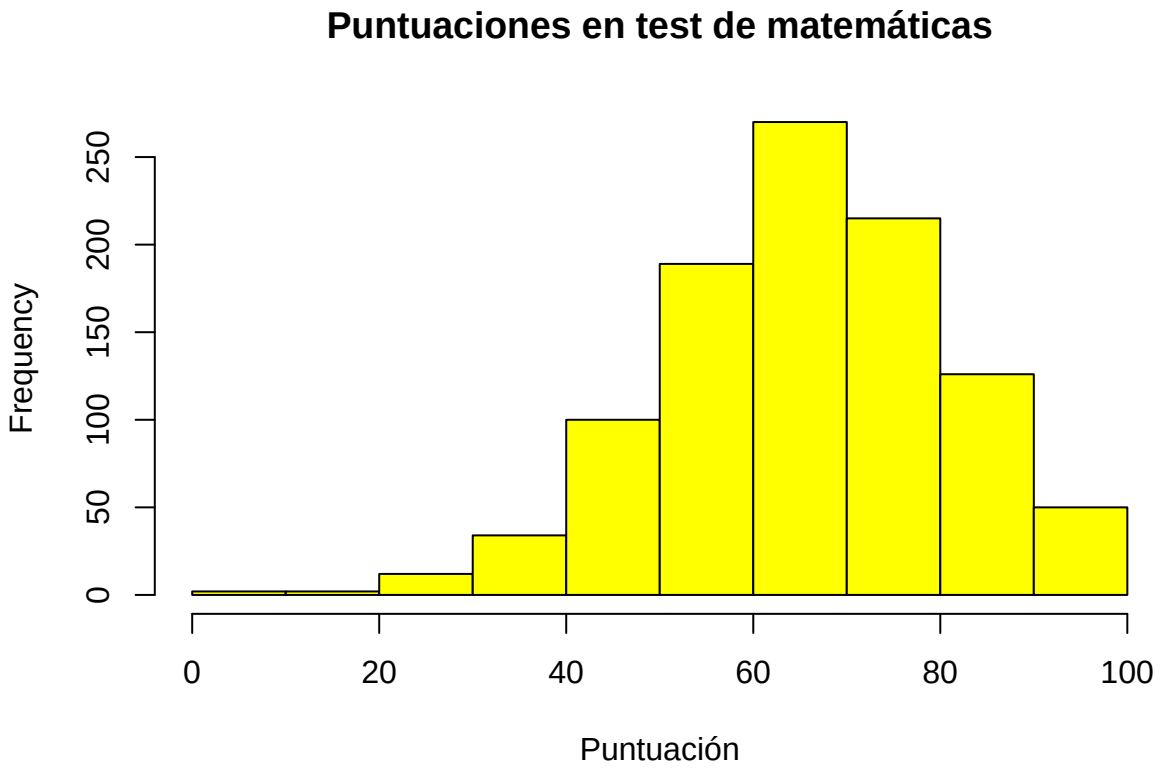
```
pie(c(pct_prep[1,1], pct_prep[1,2]), labels = paste(dataLevels$Preparation, sep = " ",
                                                    "[", pct_prep, "%]"),
    main = "Según su preparación")
```



2.2. Valores numéricos

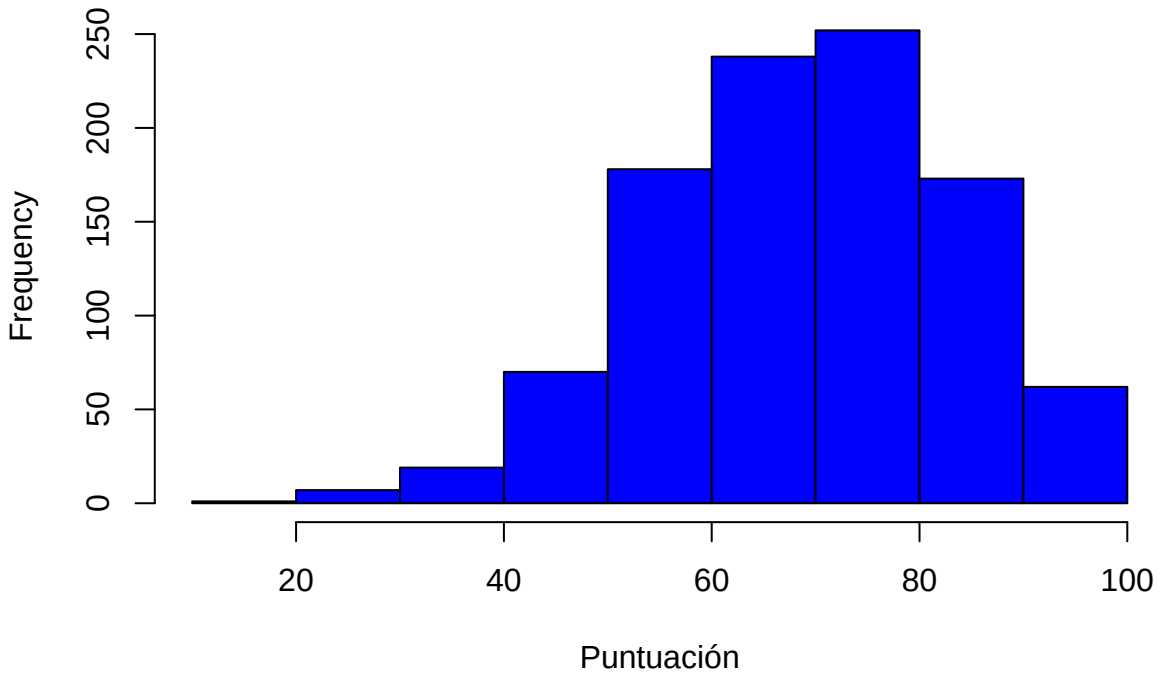
Vamos a continuar con la visualización de los valores numéricos, comenzando con un histograma de cada uno de las puntuaciones de cada uno de los tests.

```
hist(data$Math, main = "Puntuaciones en test de matemáticas", xlab = "Puntuación",
     col = "yellow")
```



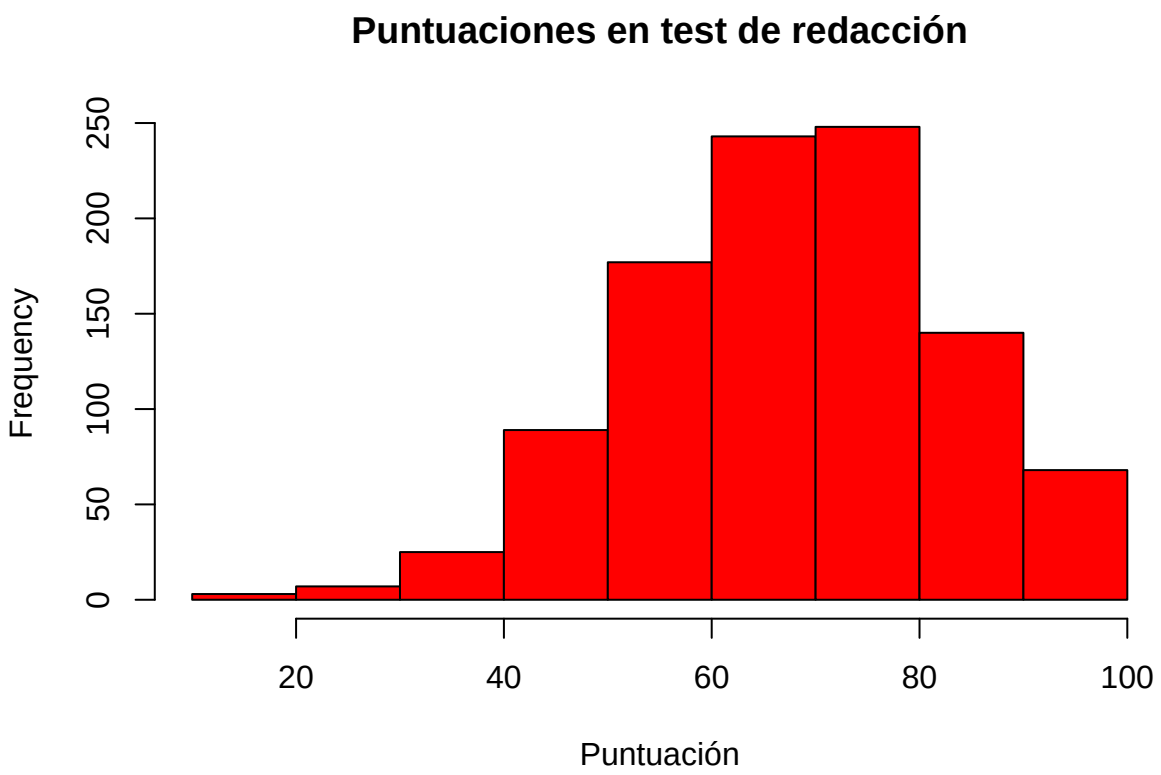
Como se puede observar en el histograma, el rango de puntuación más repetido en el test de matemáticas se encuentra entre los 60 y 70 puntos.

```
hist(data$Reading, main = "Puntuaciones en test de lectura", xlab = "Puntuación",
     col = "blue")
```



Según este histograma, el rango de puntuación más repetido en el test de lectura se encuentra entre los 70 y 80 puntos, seguido de cerca por el rango entre los 60 y 70.

```
hist(data$Writing, main = "Puntuaciones en test de redacción", xlab = "Puntuación",
     col = "red")
```



Según el histograma del test de redacción, el rango de puntuación más repetido entre los 70 y 80 puntos, seguido de cerca por el rango entre los 60 y 70.

Como se puede observar de los tres histogramas, nos encontramos con que los valores anteriores al 30 parecen ser outliers. Por este motivo, durante la limpieza se buscarán estos valores.

Vamos a continuar con la relación de aprobados y suspensos. Para obtener estos datos por asignatura se ha puesto como nota mínima para aprobar un 50 sobre 100.

```
pass <- 50
```

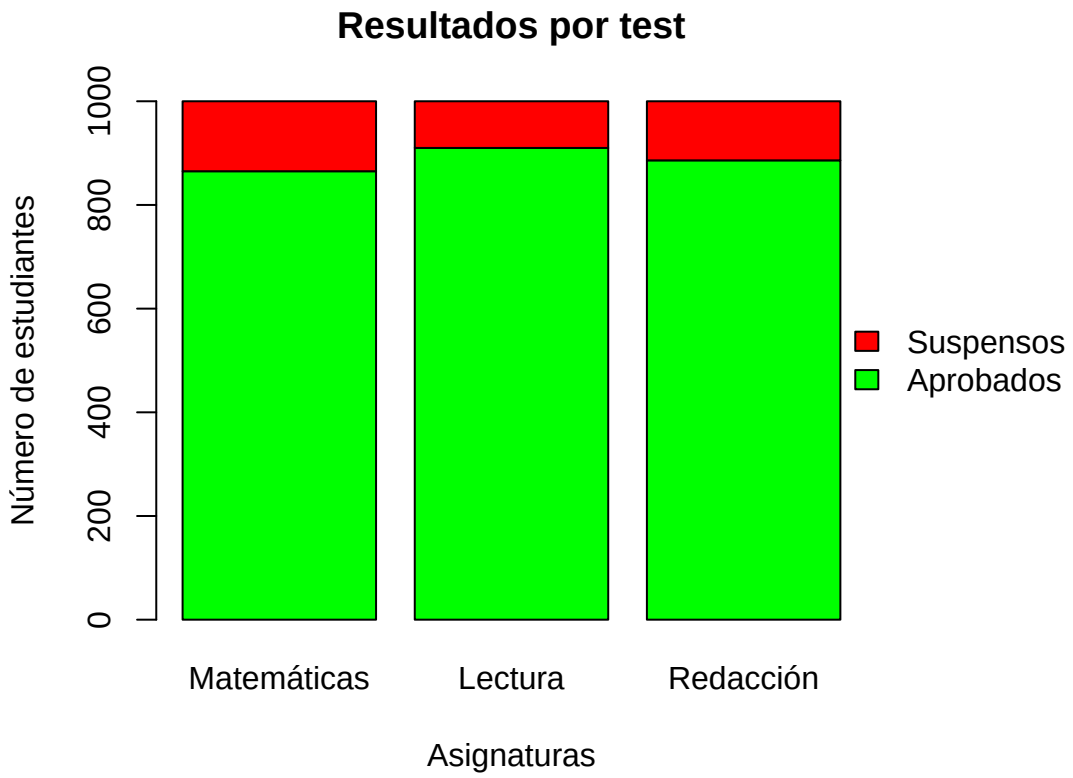
```
test_pass <- data %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )

test_fail <- data %>%
  summarise(
    count_math = sum(Math<pass),
    count_reading = sum(Reading<pass),
    count_writing = sum(Writing<pass),
```

```
    .groups = "drop"
  )

tests_stats <- rbind(test_pass,test_fail)

rownames(tests_stats)<-c("Aprobado","Suspenso")
colnames(tests_stats)<-c("Matematicas", "Lectura","Redaccion")

knitr::kable(tests_stats)
```

	Matematicas	Lectura	Redaccion
Aprobado	865	910	886
Suspenso	135	90	114

Como se puede observar, han aprobado en cada test más de un 85% del total, siendo además cantidades similares de aprobados. Además, la asignatura con el mayor número de aprobados es lectura, con 910 estudiantes aprobados.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(tests_stats[1,]),as.numeric(tests_stats[2,])) ~
          c("Matemáticas","Lectura","Redacción"),
        xlab = "Asignaturas",
        ylab = "Número de estudiantes",
        ylim = c(0, 1000),
        col = c("green", "red") ,
        legend.text = c("Aprobados", "Suspensos"),
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Resultados por test")
```

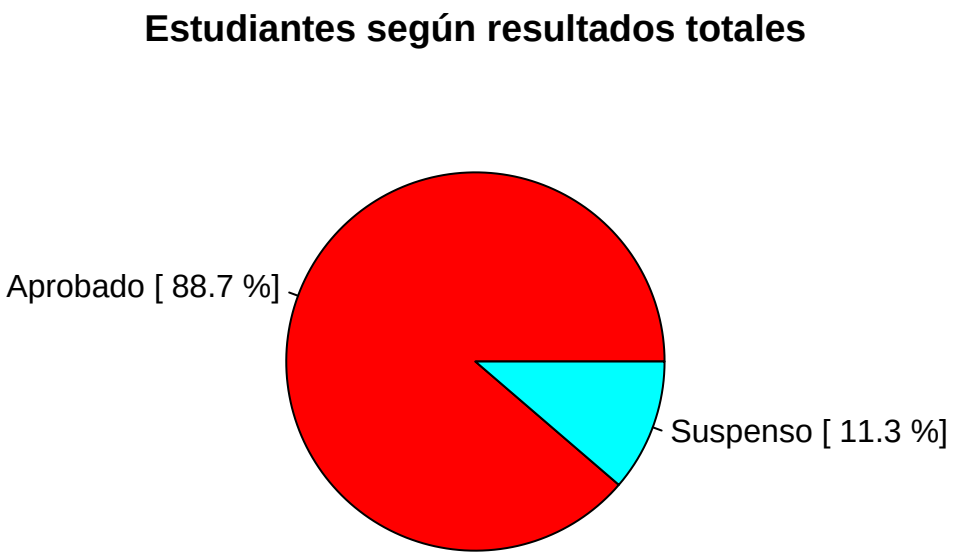



```
pct_tests <- NULL

for (k in 1:dim(tests_stats)[2]) {
  pct <- round(100*c(as.numeric(tests_stats[,k])))/sum(c(as.numeric(tests_stats[,k])))
  if (is.null(pct_tests)) {
    pct_tests <- pct
  } else {
    pct_tests <- cbind(pct_tests,pct)
  }
}

rownames(pct_tests)<-c("Aprobado","Suspenso")
colnames(pct_tests)<-c("Matematicas", "Lectura","Redaccion")

pct_tests <- as.data.frame(pct_tests)

knitr::kable(pct_tests)
```

	Matematicas	Lectura	Redaccion
Aprobado	86.5	91	88.6
Suspenso	13.5	9	11.4

Al realizar los porcentajes de los resultados, podemos confirmar que el mínimo porcentaje de aprobados es en el test de matemáticas con un 86,5% y el máximo es del 91% en el caso del test de lectura.

```
pct_pass <- mean(as.numeric(pct_tests[1,]))

pct_fail <- mean(as.numeric(pct_tests[2,]))

pie(c(pct_pass, pct_fail), col = rainbow(dim(tests_stats)[1]),
    labels = paste(rownames(tests_stats), sep = " ", "[", c(c(pct_pass, pct_fail)),
                   "%]"), main = "Estudiantes según resultados totales")
```



2.2.1. Correlación entre puntuaciones

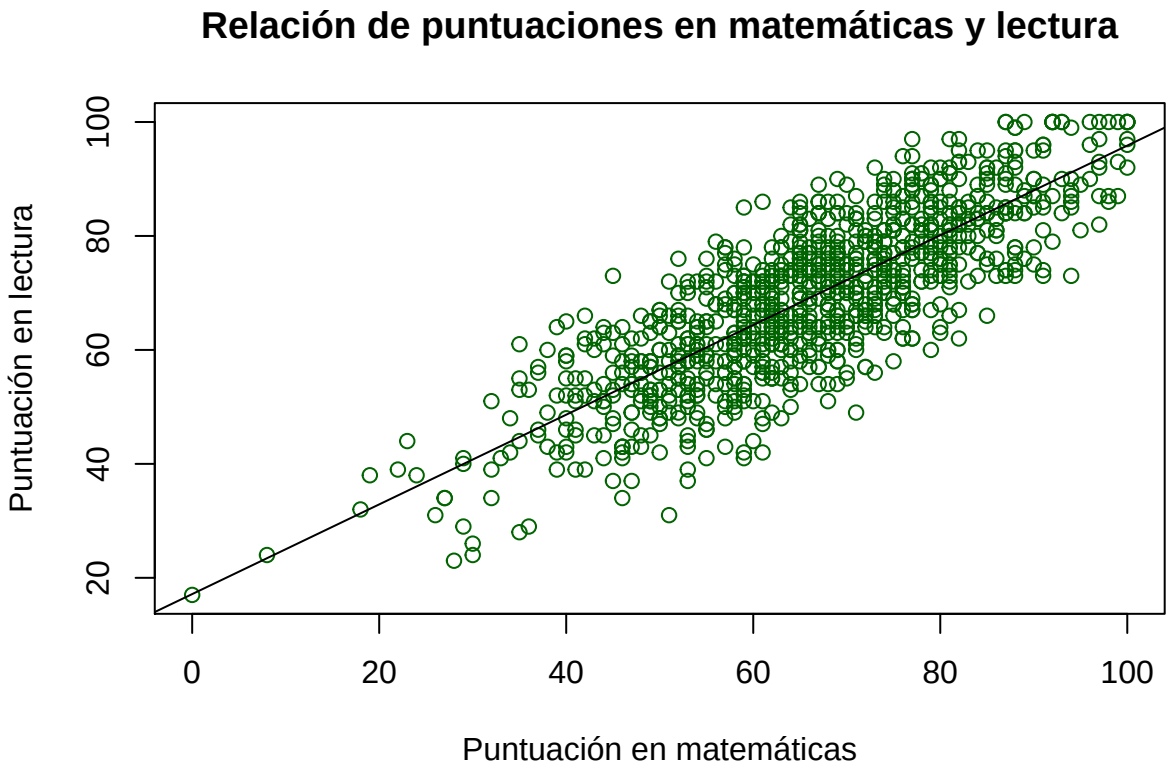
Vamos a ver la correlación entre las puntuaciones de los tests:

```
knitr::kable(cor(data[,6:8]))
```

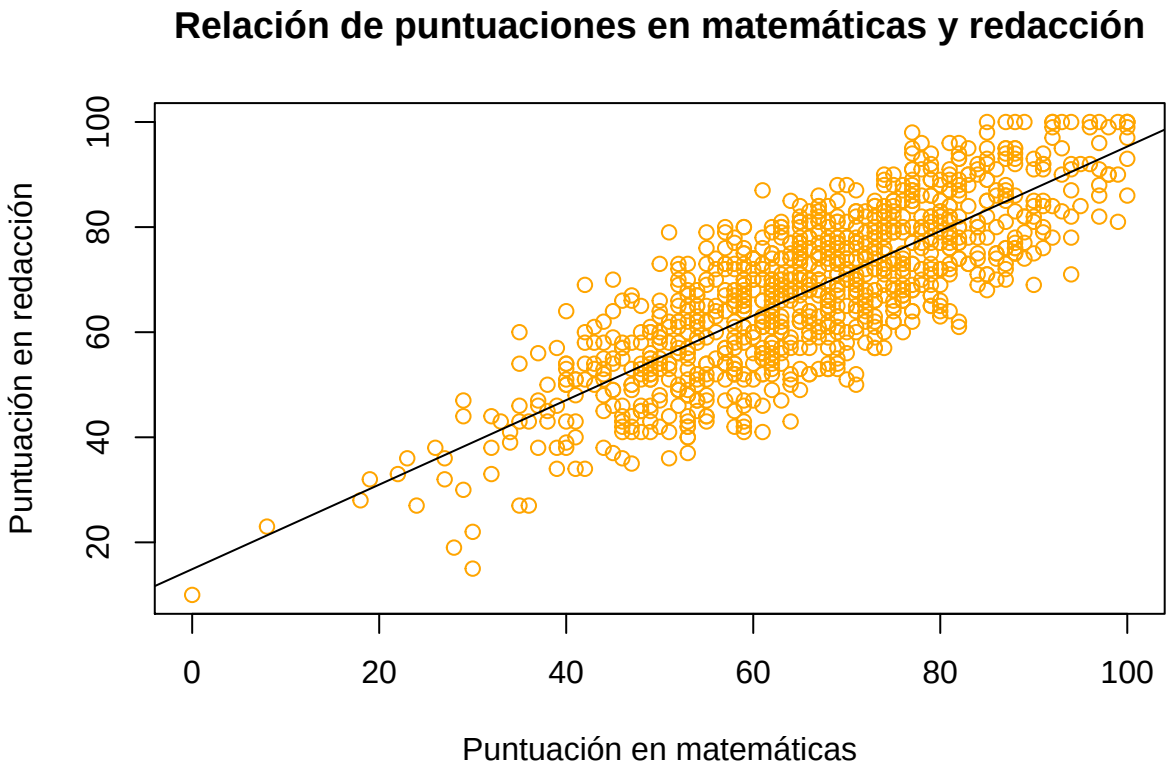
	Math	Reading	Writing
Math	1.0000000	0.8175797	0.8026420
Reading	0.8175797	1.0000000	0.9545981
Writing	0.8026420	0.9545981	1.0000000

Como se puede obsevar en la tabla, los tests con mayor correlación son el de lectura y el de redacción, aunque existe una gran correlación entre todos los tests. Para poder visualizar esta correlación de una manera gráfica se han creado los diagramas de puntos que aparecen a continuación:

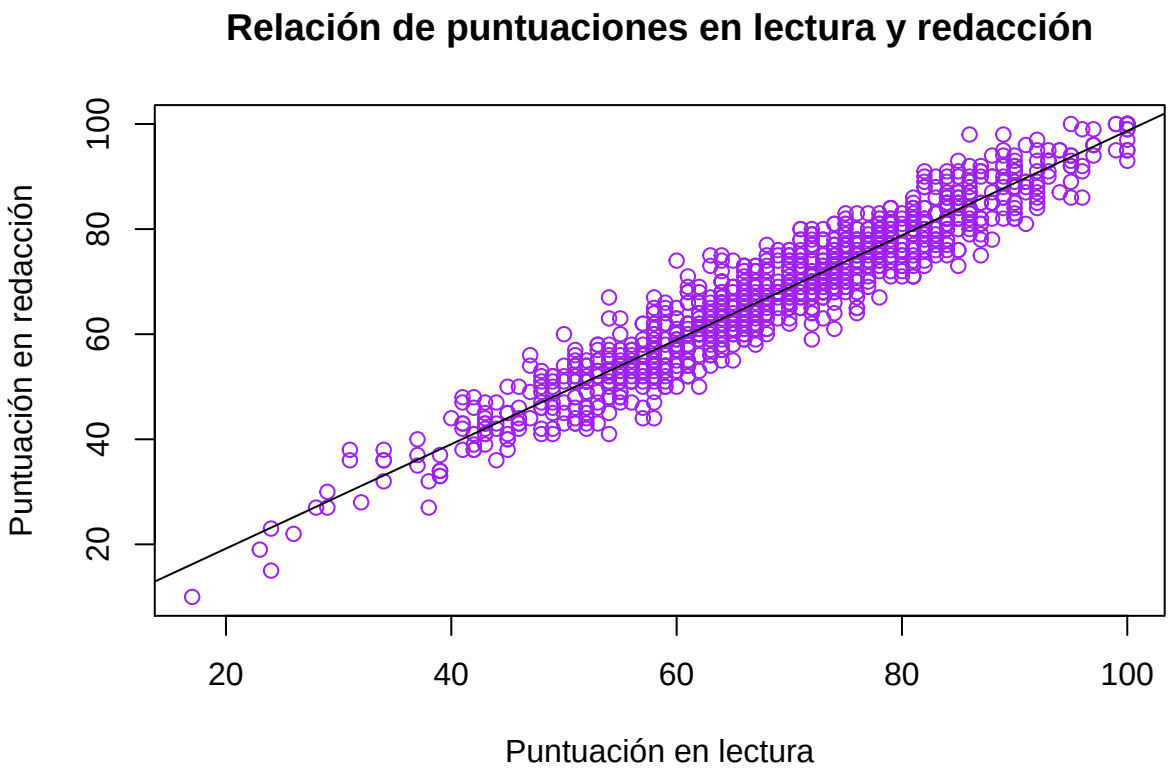
```
plot(data$Math, data$Reading,
     main = "Relación de puntuaciones en matemáticas y lectura",
     col = "dark green",
     xlab = "Puntuación en matemáticas",
     ylab = "Puntuación en lectura")

abline(lm(data$Reading ~ data$Math))
```




```
plot(data$Math, data$Writing,
     main = "Relación de puntuaciones en matemáticas y redacción",
     col = "orange",
     xlab = "Puntuación en matemáticas",
     ylab = "Puntuación en redacción")

abline(lm(data$Writing ~ data$Math))
```




```
plot(data$Reading, data$Writing,
     main = "Relación de puntuaciones en lectura y redacción",
     col = "purple",
     xlab = "Puntuación en lectura",
     ylab = "Puntuación en redacción")

abline(lm(data$Writing ~ data$Reading))
```



2.3. General (Aprobados)

Vamos a comenzar a visualizar la relación entre las personas aprobadas y los distintos factores.

2.3.1. Resultados por género

En primer lugar, vamos a comprobar si existe alguna diferencia significativa entres los hombres y las mujeres en los resultados de los tests.

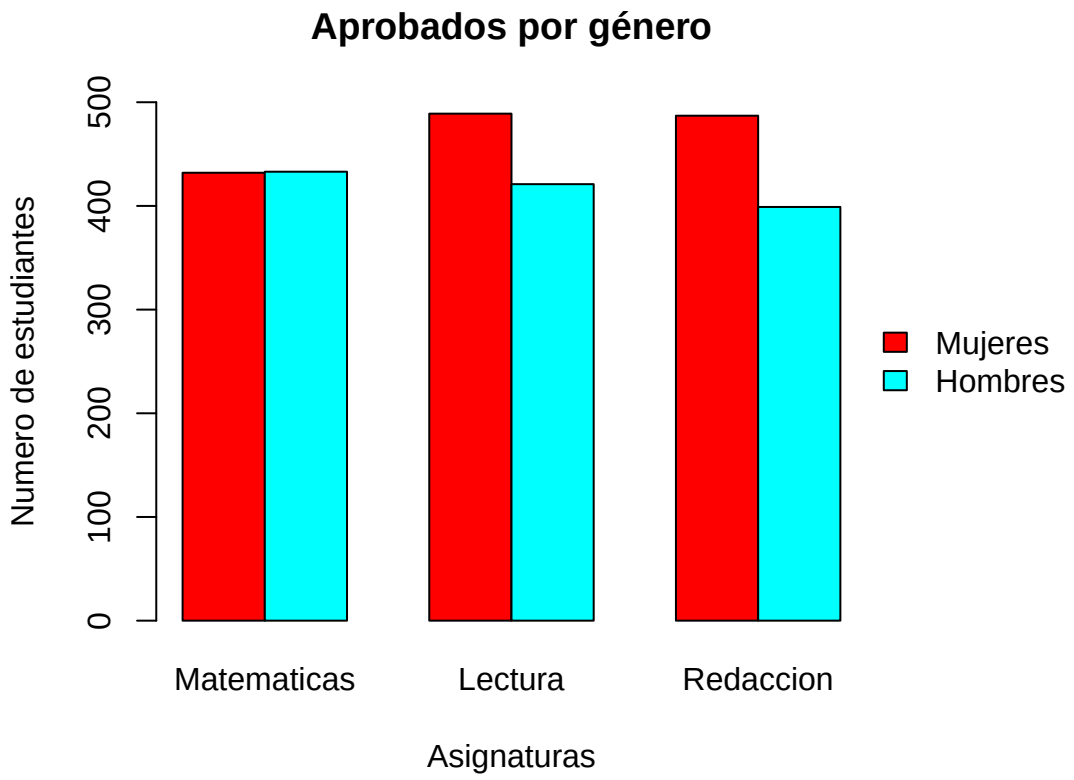
```
gender_tests <- NULL

for (k in 1:length(dataLevels$Gender)) {
  aux_gender <- data[data$Gender==dataLevels$Gender[k],] %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )
  if (is.null(gender_tests)) {
    gender_tests <- aux_gender
  } else {
    gender_tests <- rbind(gender_tests,aux_gender)
  }
}

rownames(gender_tests)<-c("Mujer","Hombre")
colnames(gender_tests)<-c("Matematicas", "Lectura","Redaccion")
```

```
gender_tests <- as.data.frame(gender_tests)

knitr::kable(gender_tests)
```

	Matematicas	Lectura	Redaccion
Mujer	432	489	487
Hombre	433	421	399

Como se puede observar, las mujeres se muestran superiores en los campos de lectura y redacción, con una diferencia aproximada de 60 personas en cada uno de estos. Por otro lado, en el campo de las matemáticas nos encontramos unos resultados más similares, con una sola persona más por parte de los hombres.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(gender_tests[1,]),as.numeric(gender_tests[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Numero de estudiantes",
        ylim = c(0, 500),
        col = rainbow(dim(gender_tests)[1]),
        legend.text = c("Mujeres", "Hombres"),
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Aprobados por género")
```



2.3.2. Resultados según la etnia

En segundo lugar, vamos a comprobar si existe alguna relación entre los miembros de una etnia y sus resultados de los tests.

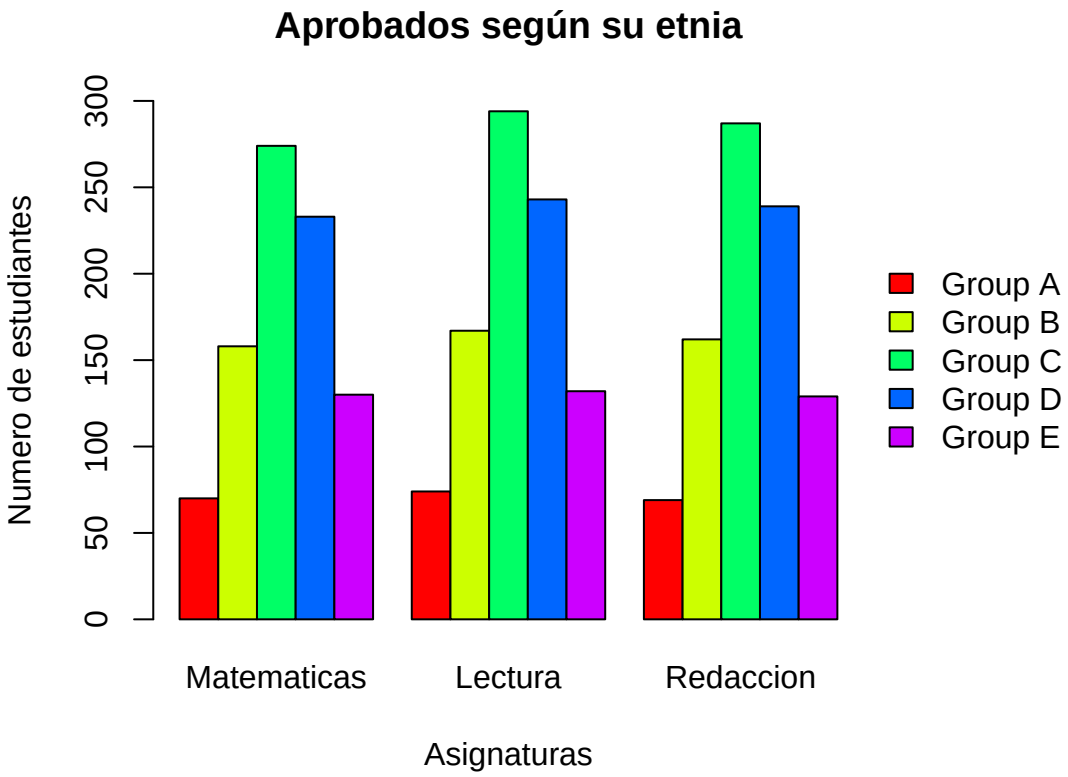
```
ethnicity_tests <- NULL

for (k in 1:length(dataLevels$Ethnicity)) {
  aux_ethnicity <- data[data$Ethnicity==dataLevels$Ethnicity[k],] %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )
  if (is.null(ethnicity_tests)) {
    ethnicity_tests <- aux_ethnicity
  } else {
    ethnicity_tests <- rbind(ethnicity_tests,aux_ethnicity)
  }
}

rownames(ethnicity_tests)<-c("Group A","Group B","Group C","Group D","Group E")
colnames(ethnicity_tests)<-c("Matematicas", "Lectura","Redaccion")

ethnicity_tests <- as.data.frame(ethnicity_tests)

knitr::kable(ethnicity_tests)
```

	Matematicas	Lectura	Redaccion
Group A	70	74	69
Group B	158	167	162
Group C	274	294	287
Group D	233	243	239
Group E	130	132	129

Como se puede observar, el grupo que ha obtenido el mejor conjunto de resultados es el grupo c, seguido por el d, continuando con el b, teniendo como siguiente el e y, por último, el a. De la misma manera, se puede observar que las personas de cada grupo que aprueban un test, suele superar el resto también.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(ethnicity_tests[1,]),as.numeric(ethnicity_tests[2,]),
              as.numeric(ethnicity_tests[3,]),as.numeric(ethnicity_tests[4,]),
              as.numeric(ethnicity_tests[5,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Numero de estudiantes",
        ylim = c(0, 300),
        col = rainbow(dim(ethnicity_tests)[1]),
        legend.text = c("Group A","Group B","Group C","Group D","Group E"),
```

```
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Aprobados según su etnia")
```



2.3.3. Resultados según los estudios de los padres

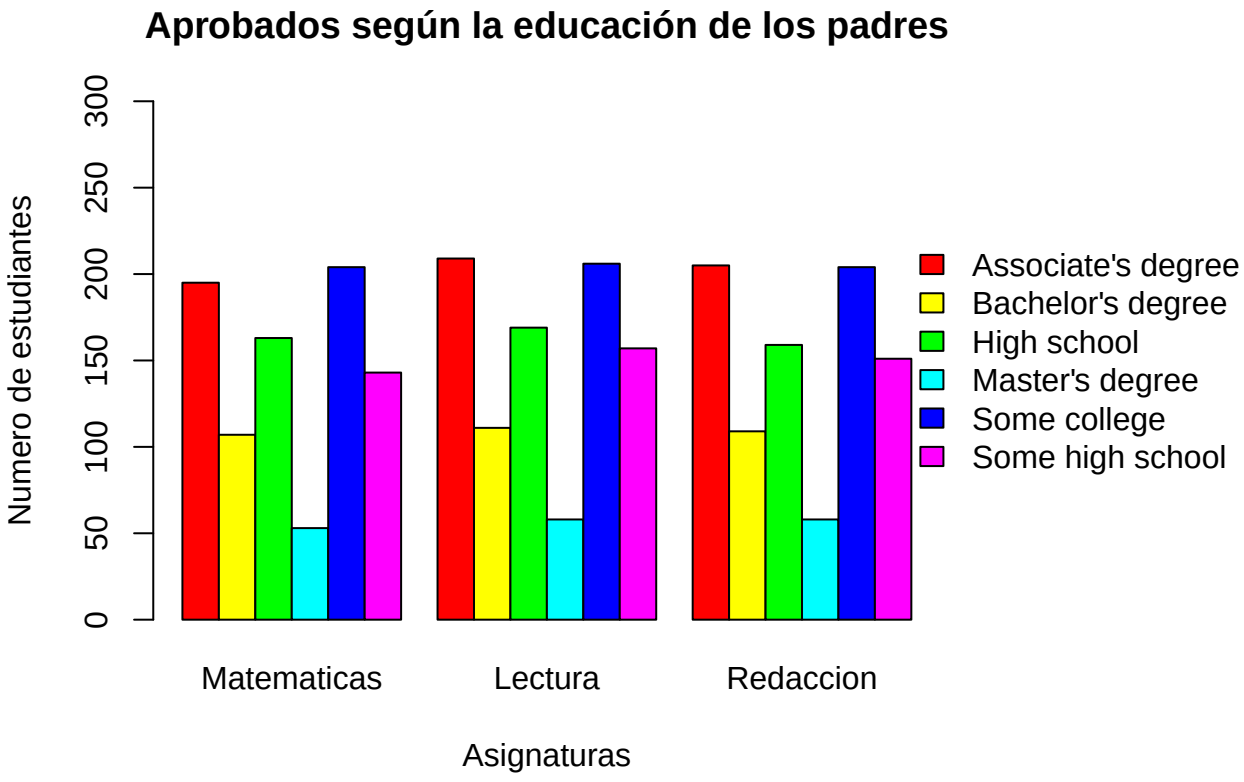
```
parent_tests <- NULL

for (k in 1:length(dataLevels$Parent_Education)) {
  aux_parent <- data[data$Parent_Education==dataLevels$Parent_Education[k],] %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )
  if (is.null(parent_tests)) {
    parent_tests <- aux_parent
  } else {
    parent_tests <- rbind(parent_tests,aux_parent)
  }
}

rownames(parent_tests)<-c("Associate's degree","Bachelor's degree","High school",
                          "Master's degree","Some college","Some high school")
colnames(parent_tests)<-c("Matematicas", "Lectura","Redaccion")
```

```r
parent_tests <- as.data.frame(parent_tests)

knitr::kable(parent_tests)
```

	Matematicas	Lectura	Redaccion
Associate's degree	195	209	205
Bachelor's degree	107	111	109
High school	163	169	159
Master's degree	53	58	58
Some college	204	206	204
Some high school	143	157	151

Como se puede observar, los valores obtenidos son proporcionales a la cantidad de personas total que forma cada grupo. De la misma manera, hay una cantidad similar de perrsonas que han aprobado en cada uno de los tests por cada uno de los niveles de educación.

```r
par(mar = c(5,4,4,8))

barplot(cbind(as.numeric(parent_tests[1,]),as.numeric(parent_tests[2,]),
              as.numeric(parent_tests[3,]),as.numeric(parent_tests[4,]),
              as.numeric(parent_tests[5,]),as.numeric(parent_tests[6,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Numero de estudiantes",
        ylim = c(0, 300),
        col = rainbow(dim(parent_tests)[1]),
        legend.text = c("Associate's degree","Bachelor's degree","High school",
                        "Master's degree","Some college","Some high school"),
        args.legend = list(x = "right", bty="n", inset=c(-0.40,0), xpd = TRUE),
        main = "Aprobados según la educación de los padres")
```



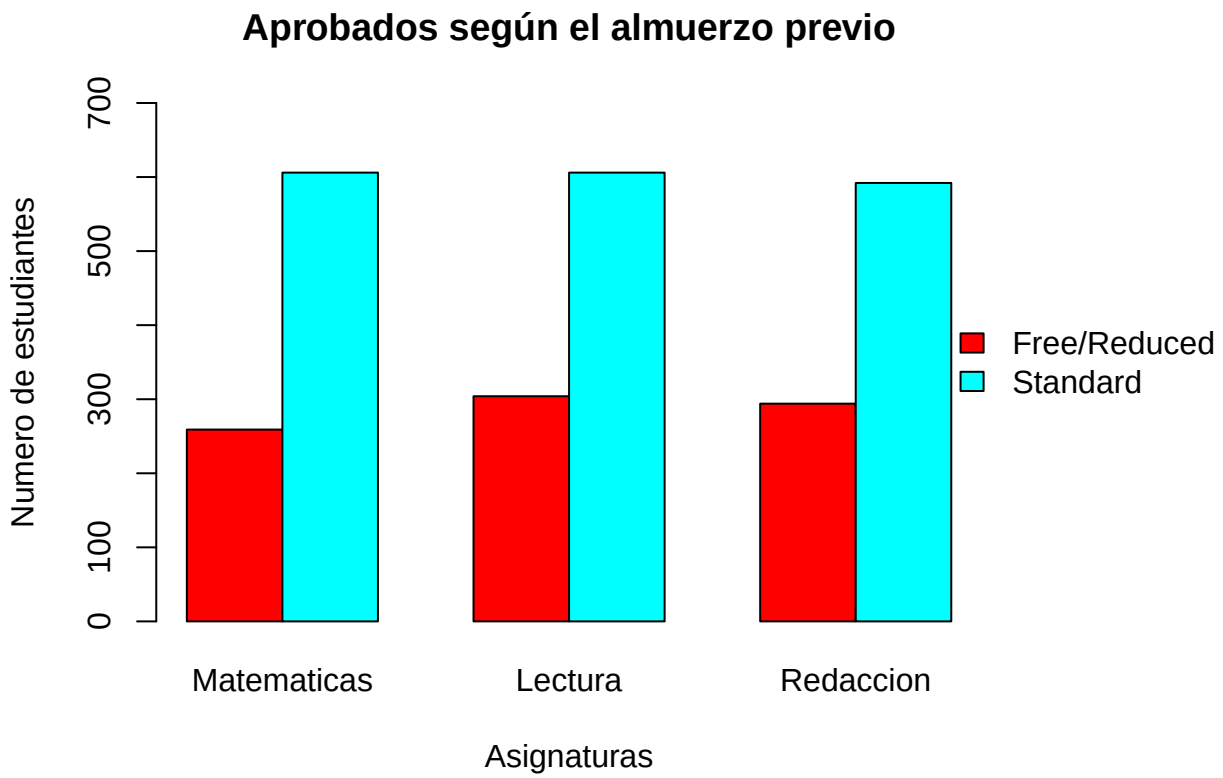
2.3.4. Resultados según el almuerzo previo a los tests

```
lunch_tests <- NULL

for (k in 1:length(dataLevels$Lunch)) {
  aux_lunch <- data[data$Lunch==dataLevels$Lunch[k],] %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )
  if (is.null(lunch_tests)) {
    lunch_tests <- aux_lunch
  } else {
    lunch_tests <- rbind(lunch_tests,aux_lunch)
  }
}

rownames(lunch_tests)<-c("Free/Reduced","Standard")
colnames(lunch_tests)<-c("Matematicas", "Lectura","Redaccion")

lunch_tests <- as.data.frame(lunch_tests)

knitr::kable(lunch_tests)
```

	Matematicas	Lectura	Redaccion
Free/Reduced	259	304	294
Standard	606	606	592

Como se puede comprobar al ver los datos, la mayoría de las personas que han aprobado en cada uno de los tests ha tenido un almuerzo normal antes del examen, dando a entender de que se trata de la opción preferible sobre tener un almuerzo reducido.

```
par(mar = c(5,4,4,7))

barplot(cbind(as.numeric(lunch_tests[1,]),as.numeric(lunch_tests[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Numero de estudiantes",
        ylim = c(0, 700),
        col = rainbow(dim(lunch_tests)[1]),
        legend.text = c("Free/Reduced","Standard"),
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Aprobados según el almuerzo previo")
```



2.3.5. Resultados según la preparación para los tests

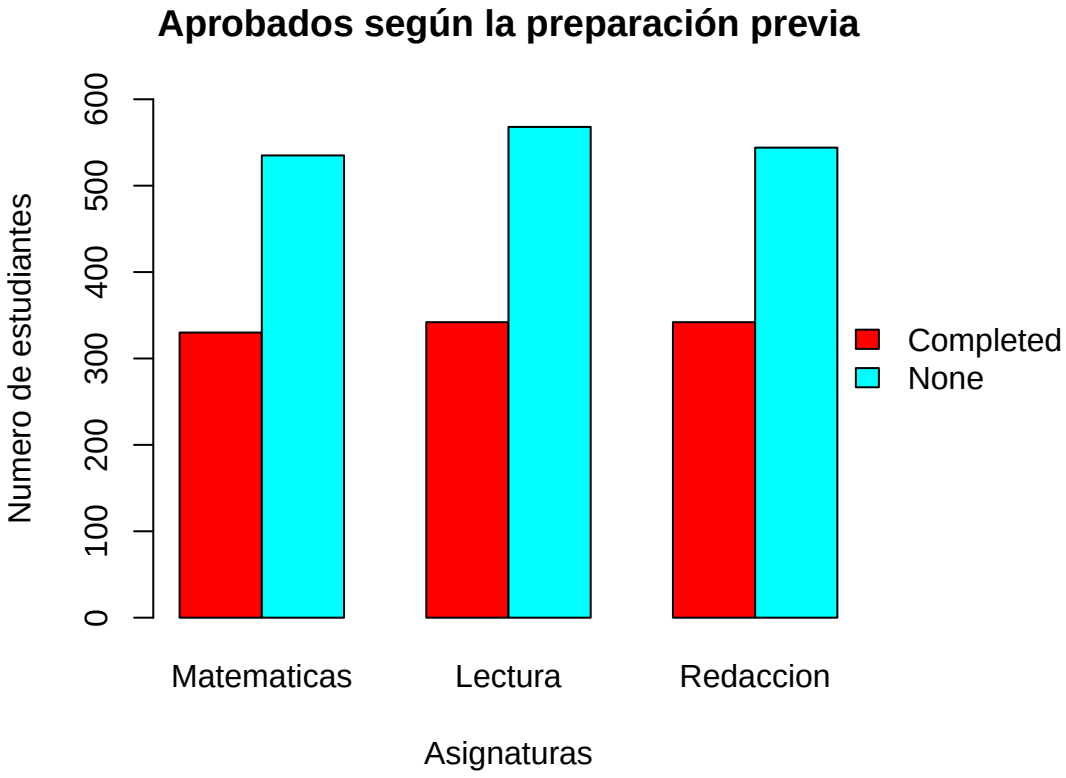
```
preparation_tests <- NULL
```

```
for (k in 1:length(dataLevels$Preparation)) {
  aux_preparation <- data[data$Preparation==dataLevels$Preparation[k],] %>%
  summarise(
    count_math = sum(Math>=pass),
    count_reading = sum(Reading>=pass),
    count_writing = sum(Writing>=pass),
    .groups = "drop"
  )
  if (is.null(preparation_tests)) {
    preparation_tests <- aux_preparation
  } else {
    preparation_tests <- rbind(preparation_tests,aux_preparation)
  }
}

rownames(preparation_tests)<-c("Completed","None")
colnames(preparation_tests)<-c("Matematicas", "Lectura","Redaccion")

preparation_tests <- as.data.frame(preparation_tests)

knitr::kable(preparation_tests)
```

	Matematicas	Lectura	Redaccion
Completed	330	342	342
None	535	568	544

Como se puede observar, la mayor parte de los estudiantes que han superado los tests se tratan de personas que nos se los han preparado, con una diferencia aproximada de unos 200 estudiantes en cada uno de los tests.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(preparation_tests[1,]),as.numeric(preparation_tests[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Numero de estudiantes",
        ylim = c(0, 600),
        col = rainbow(dim(preparation_tests)[1]),
        legend.text = c("Completed","None"),
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Aprobados según la preparación previa")
```



2.4. General (Puntuaciones medias)

Vamos a comenzar a visualizar la relación entre la puntuación media por test y los distintos factores.

2.4.1. Resultados por género

En primer lugar, vamos a comprobar si existe alguna diferencia significativa entres los hombres y las mujeres en los resultados de los tests.

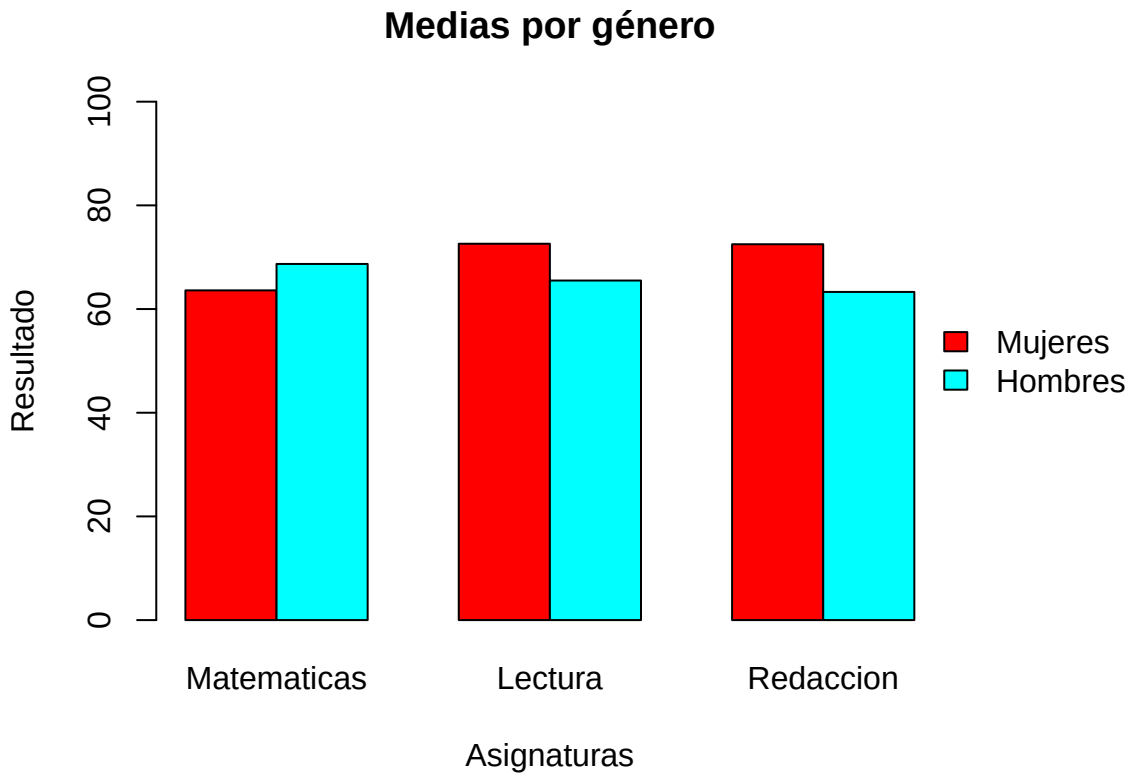
```
gender_avg_score <- NULL

for (k in 1:length(dataLevels$Gender)) {
  aux_gender <- data[data$Gender==dataLevels$Gender[k],] %>%
  summarise(
    avg_score_math = round(mean(Math),1),
    avg_score_reading = round(mean(Reading),1),
    avg_score_writing = round(mean(Writing),1),
    .groups = "drop"
  )
  if (is.null(gender_avg_score)) {
    gender_avg_score <- aux_gender
  } else {
    gender_avg_score <- rbind(gender_avg_score,aux_gender)
  }
}

rownames(gender_avg_score)<-c("Mujer","Hombre")
```

```
colnames(gender_avg_score)<-c("Matematicas Media", "Lectura Media","Redaccion Media")

gender_avg_score <- as.data.frame(gender_avg_score)

knitr::kable(gender_avg_score)
```

	Matematicas Media	Lectura Media	Redaccion Media
Mujer	63.6	72.6	72.5
Hombre	68.7	65.5	63.3

Los resultados obtenidos son semejantes al del número de aprobados por género, la media de las mujeres en lectura y redacción es superior, pero en matemáticas los hombres tienen una media superior.

```
par(mar = c(5,4,4,8))

barplot(cbind(as.numeric(gender_avg_score[1,]),as.numeric(gender_avg_score[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Resultado",
        ylim = c(0, 100),
        col = rainbow(dim(gender_avg_score)[1]),
        args.legend = list(x = "right", bty="n", inset=c(-0.25,0), xpd = TRUE),
        legend.text = c("Mujeres", "Hombres"),
        main = "Medias por género")
```



2.4.2. Resultados según la etnia

En segundo lugar, vamos a comprobar si existe alguna relación entre los miembros de una etnia y sus resultados de los tests.

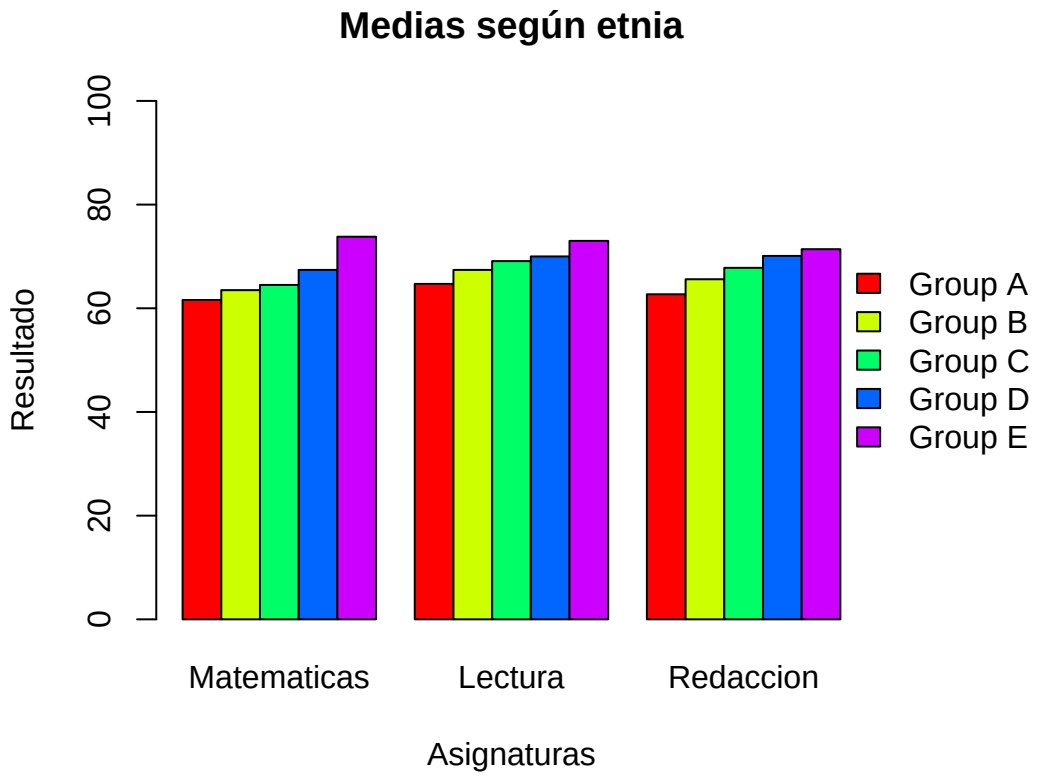
```
ethnicity_avg_score <- NULL

for (k in 1:length(dataLevels$Ethnicity)) {
  aux_ethnicity <- data[data$Ethnicity==dataLevels$Ethnicity[k],] %>%
  summarise(
    avg_score_math = round(mean(Math),1),
    avg_score_reading = round(mean(Reading),1),
    avg_score_writing = round(mean(Writing),1),
    .groups = "drop"
  )
  if (is.null(ethnicity_avg_score)) {
    ethnicity_avg_score <- aux_ethnicity
  } else {
    ethnicity_avg_score <- rbind(ethnicity_avg_score,aux_ethnicity)
  }
}

rownames(ethnicity_avg_score)<-c("Group A","Group B","Group C","Group D","Group E")
colnames(ethnicity_avg_score)<-c("Matematicas Media", "Lectura Media","Redaccion Media")

ethnicity_avg_score <- as.data.frame(ethnicity_avg_score)

knitr::kable(ethnicity_avg_score)
```

	Matematicas Media	Lectura Media	Redaccion Media
Group A	61.6	64.7	62.7
Group B	63.5	67.4	65.6
Group C	64.5	69.1	67.8
Group D	67.4	70.0	70.1
Group E	73.8	73.0	71.4

Como se puede observar, el grupo que ha obtenido la mejor media en todas los tests es el grupo e, seguido por d, c, b y a, en ese orden. Cabe destacar también que cada grupo ha obtenido una media similar en cada uno de los tests.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(ethnicity_avg_score[1,]),as.numeric(ethnicity_avg_score[2,]),
              as.numeric(ethnicity_avg_score[3,]),as.numeric(ethnicity_avg_score[4,]),
              as.numeric(ethnicity_avg_score[5,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Resultado",
        ylim = c(0, 100),
        col = rainbow(dim(ethnicity_avg_score)[1]),
        legend.text = c("Group A","Group B","Group C","Group D","Group E"),
```

```
        args.legend = list(x = "right", bty="n", inset=c(-0.25,0), xpd = TRUE),
        main = "Medias según etnia")
```



2.4.3. Resultados según los estudios de los padres

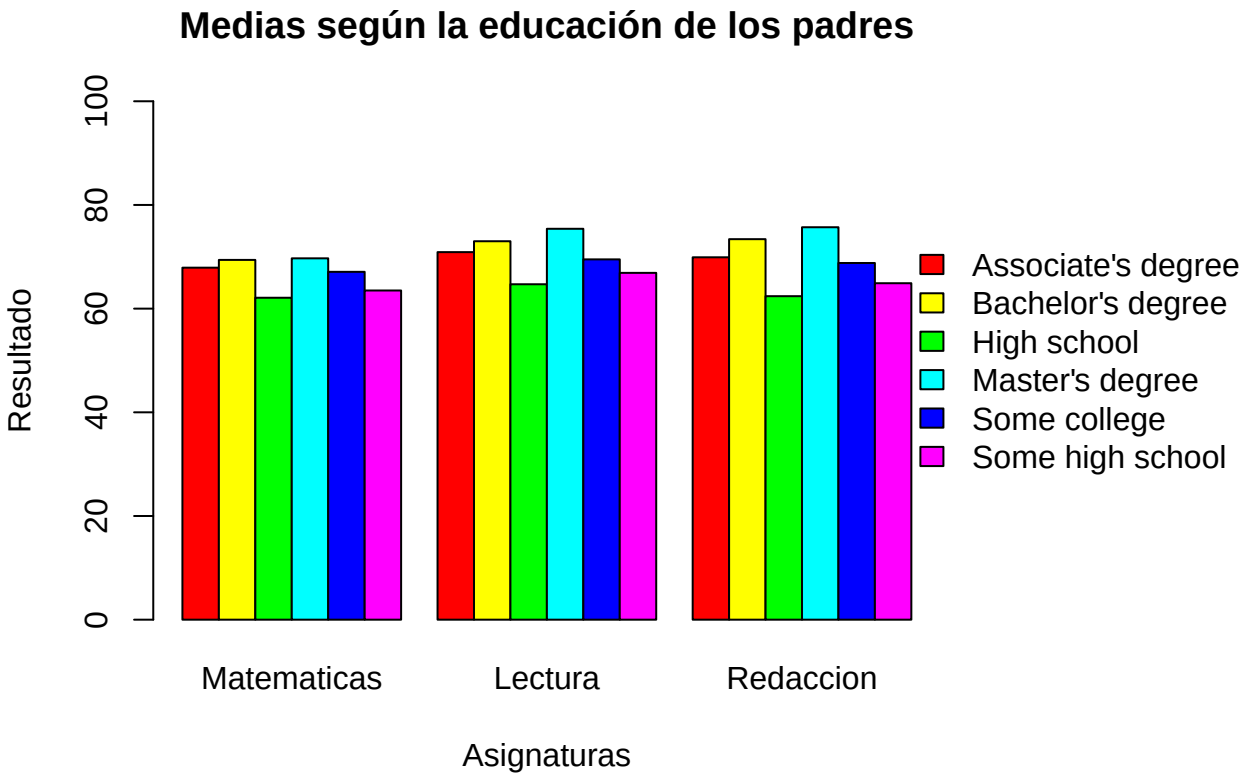
```
parent_avg_score <- NULL

for (k in 1:length(dataLevels$Parent_Education)) {
  aux_parent <- data[data$Parent_Education==dataLevels$Parent_Education[k],] %>%
  summarise(
    avg_score_math = round(mean(Math),1),
    avg_score_reading = round(mean(Reading),1),
    avg_score_writing = round(mean(Writing),1),
    .groups = "drop"
  )
  if (is.null(parent_avg_score)) {
    parent_avg_score <- aux_parent
  } else {
    parent_avg_score <- rbind(parent_avg_score,aux_parent)
  }
}

rownames(parent_avg_score)<-c("Associate's degree","Bachelor's degree","High school",
                              "Master's degree","Some college","Some high school")
colnames(parent_avg_score)<-c("Matematicas Media", "Lectura Media","Redaccion Media")
```

```r
parent_avg_score <- as.data.frame(parent_avg_score)

knitr::kable(parent_avg_score)
```

	Matematicas Media	Lectura Media	Redaccion Media
Associate's degree	67.9	70.9	69.9
Bachelor's degree	69.4	73.0	73.4
High school	62.1	64.7	62.4
Master's degree	69.7	75.4	75.7
Some college	67.1	69.5	68.8
Some high school	63.5	66.9	64.9

Como se puede observar, los valores obtenidos son proporcionales a la cantidad de personas total que forma cada grupo. De la misma manera, hay una cantidad similar de perrsonas que han aprobado en cada uno de los tests por cada uno de los niveles de educación.

```r
par(mar = c(5,4,4,8))

barplot(cbind(as.numeric(parent_avg_score[1,]),as.numeric(parent_avg_score[2,]),
              as.numeric(parent_avg_score[3,]),as.numeric(parent_avg_score[4,]),
              as.numeric(parent_avg_score[5,]),as.numeric(parent_avg_score[6,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Resultado",
        ylim = c(0, 100),
        col = rainbow(dim(parent_avg_score)[1]),
        legend.text = c("Associate's degree","Bachelor's degree","High school",
                        "Master's degree","Some college","Some high school"),
        args.legend = list(x = "right", bty="n", inset=c(-0.40,0), xpd = TRUE),
        main = "Medias según la educación de los padres")
```



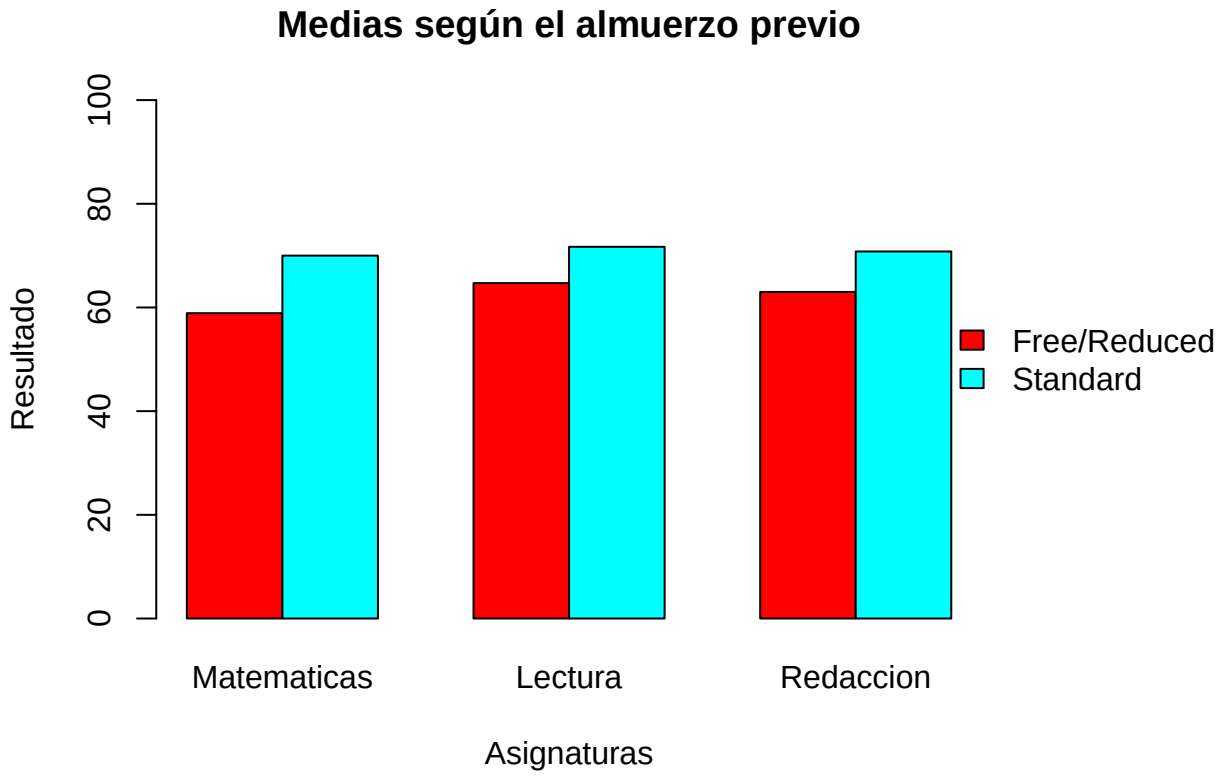
2.4.4. Resultados según el almuerzo previo a los tests

```
lunch_avg_score <- NULL

for (k in 1:length(dataLevels$Lunch)) {
  aux_lunch <- data[data$Lunch==dataLevels$Lunch[k],] %>%
  summarise(
    avg_score_math = round(mean(Math),1),
    avg_score_reading = round(mean(Reading),1),
    avg_score_writing = round(mean(Writing),1),
    .groups = "drop"
  )
  if (is.null(lunch_avg_score)) {
    lunch_avg_score <- aux_lunch
  } else {
    lunch_avg_score <- rbind(lunch_avg_score,aux_lunch)
  }
}

rownames(lunch_avg_score)<-c("Free/Reduced","Standard")
colnames(lunch_avg_score)<-c("Matematicas Media", "Lectura Media","Redaccion Media")

lunch_avg_score <- as.data.frame(lunch_avg_score)

knitr::kable(lunch_avg_score)
```

	Matematicas Media	Lectura Media	Redaccion Media
Free/Reduced	58.9	64.7	63.0
Standard	70.0	71.7	70.8

Como se puede comprobar al ver los datos, la media de los resultados de los estudiantes que han realizado un almuerzo estándar es superior, además de ser similar en los tres tests.

```
par(mar = c(5,4,4,7))

barplot(cbind(as.numeric(lunch_avg_score[1,]),as.numeric(lunch_avg_score[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Resultado",
        ylim = c(0, 100),
        col = rainbow(dim(lunch_tests)[1]),
        legend.text = c("Free/Reduced","Standard"),
        args.legend = list(x = "right", bty="n", inset=c(-0.30,0), xpd = TRUE),
        main = "Medias según el almuerzo previo")
```



2.4.5. Resultados según la preparación para los tests

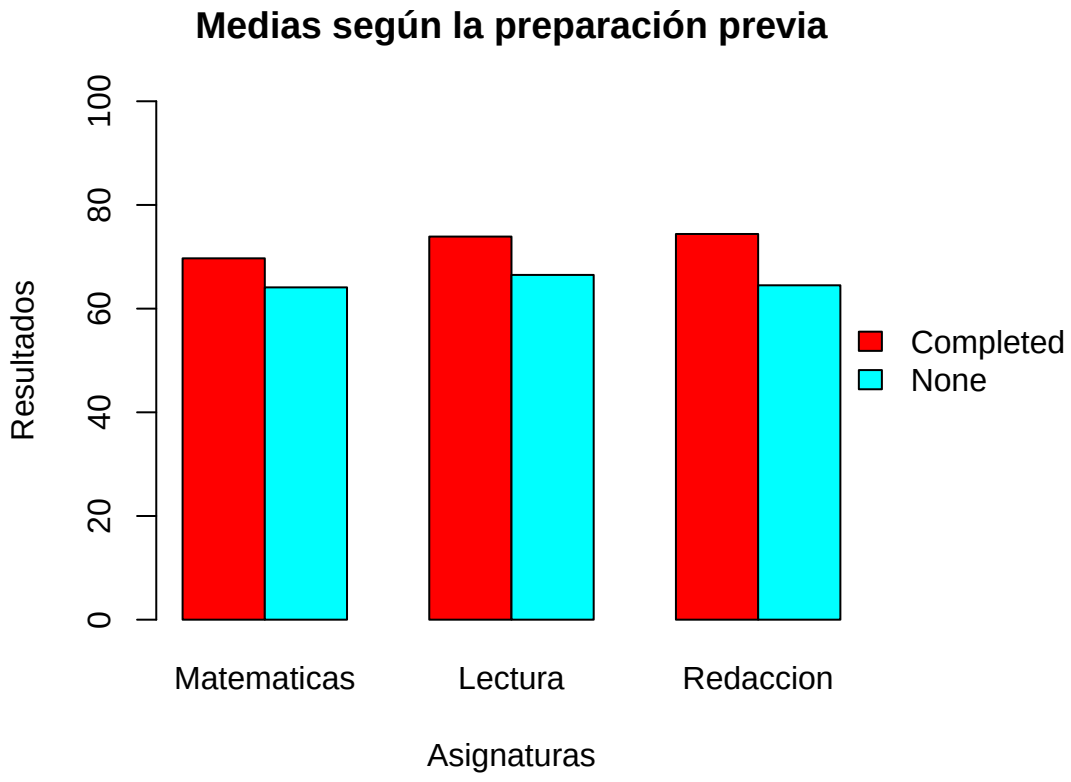
```
preparation_avg_score <- NULL

for (k in 1:length(dataLevels$Preparation)) {
```

```
  aux_preparation <- data[data$Preparation==dataLevels$Preparation[k],] %>%
  summarise(
    avg_score_math = round(mean(Math),1),
    avg_score_reading = round(mean(Reading),1),
    avg_score_writing = round(mean(Writing),1),
    .groups = "drop"
  )
  if (is.null(preparation_avg_score)) {
    preparation_avg_score <- aux_preparation
  } else {
    preparation_avg_score <- rbind(preparation_avg_score,aux_preparation)
  }
}

rownames(preparation_avg_score)<-c("Completed","None")
colnames(preparation_avg_score)<-c("Matematicas Media", "Lectura Media",
                                   "Redaccion Media")

preparation_avg_score <- as.data.frame(preparation_avg_score)

knitr::kable(preparation_avg_score)
```

	Matematicas Media	Lectura Media	Redaccion Media
Completed	69.7	73.9	74.4
None	64.1	66.5	64.5

Como se puede observar, los estudiantes con preparación previa han obtenido de media mejores resultados en todos los tests, aunque hay una mayor diferencia con matemáticas respecto al resto.

```
par(mar = c(5,4,4,10))

barplot(cbind(as.numeric(preparation_avg_score[1,]),
              as.numeric(preparation_avg_score[2,])) ~
          c("Matematicas","Lectura","Redaccion"),
        beside = TRUE,
        xlab = "Asignaturas",
        ylab = "Resultados",
        ylim = c(0, 100),
        col = rainbow(dim(preparation_tests)[1]),
        legend.text = c("Completed","None"),
        args.legend = list(x = "right", bty="n", inset=c(-0.3,0), xpd = TRUE),
        main = "Medias según la preparación previa")
```



3. Limpieza

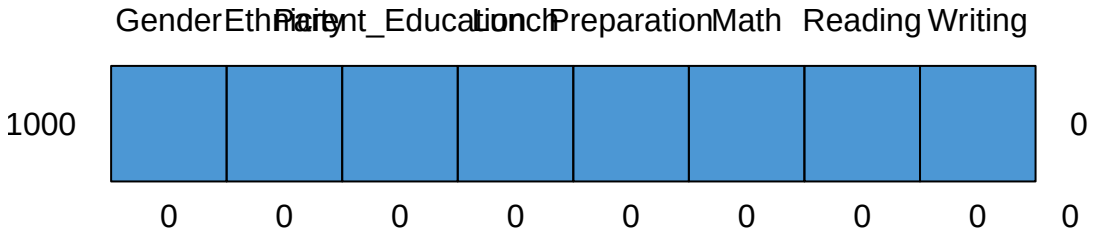
3.1. Valores ausentes

Vamos comenzar la limpieza buscando los valores ausentes, y para ello vamos a utilizar el paquete "mice".

La función en específico que vamos a utilizar es "md.pattern", que revisa las columnas de nuestro data frame y nos indica cuáles tienen valores perdidos.

```
md.pattern(data)
```

```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##      Gender Ethnicity Parent_Education Lunch Preparation Math Reading Writing
## 1000      1         1                1     1           1    1       1       1 0
##           0         0                0     0           0    0       0       0 0
```

Así mismo, vamos a utilizar la función "is.na" para buscar columnas que contengan el valor "NA".

```
find_na = function(data){

  k_row <- 1
  k_col <- 1

  mres <- NULL

  row_size <- dim(data)[1]
  col_size <- dim(data)[1]

  na_search <- is.na(data)

  for (row in na_search) {
    if(row){
      vres <- c(k_row,k_col)
      if (is.null(mres)) {
        mres <- vres
      } else {
        mres <- rbind(mres,vres)
      }
    }
    k_row <- k_row + 1

    if(k_row > row_size){
      k_row <- k_row - row_size
      k_col <- k_col + 1
    }

  }

  if(is.null(mres)) {
    print("No se han encontrado valores na")
    return(0)
  } else {
    print("Se han encontrado valores na")
    colnames(mres) = c("row","column")
    rownames(mres) = NULL

    dfres <- as.data.frame(mres)

    return(dfres)
  }

}
```

```
find_na(data)
```

```
## [1] "No se han encontrado valores na"
```

```
## [1] 0
```

Como se puede obsevar de los resultados de ambas funciones, en nuestro data frame no se encuntra ningún valor ausente.

3.2. Valores duplicados

Vamos a continuar con la búsqueda de filas duplicadas utilizando la función "duplicated". Como esta función nos devueve una lista de booleanos, para poder revisarla automáticamente se ha creado la siguiente función:

```
find_duplicated = function(data){

  k <- 1

  mres <- NULL

  dup_search <- duplicated(data)

  for (row in dup_search) {
    if(row){
      vres <- k
      if (is.null(mres)) {
        mres <- vres
      } else {
        mres <- cbind(mres,vres)
      }
    }
    k <- k + 1
  }

  if(is.null(mres)) {
    print("No se han encontrado filas con valores duplicados")
    return(0)
  } else {
    print("Se han encontrado filas con valores duplicados")
    colnames(mres) = NULL
    rownames(mres) = NULL

    dfres <- list(mres)

    return(dfres)
  }

}
```

```
find_duplicated(data)
```

```
## [1] "No se han encontrado filas con valores duplicados"
```

```
## [1] 0
```

3.3. Valores outliers

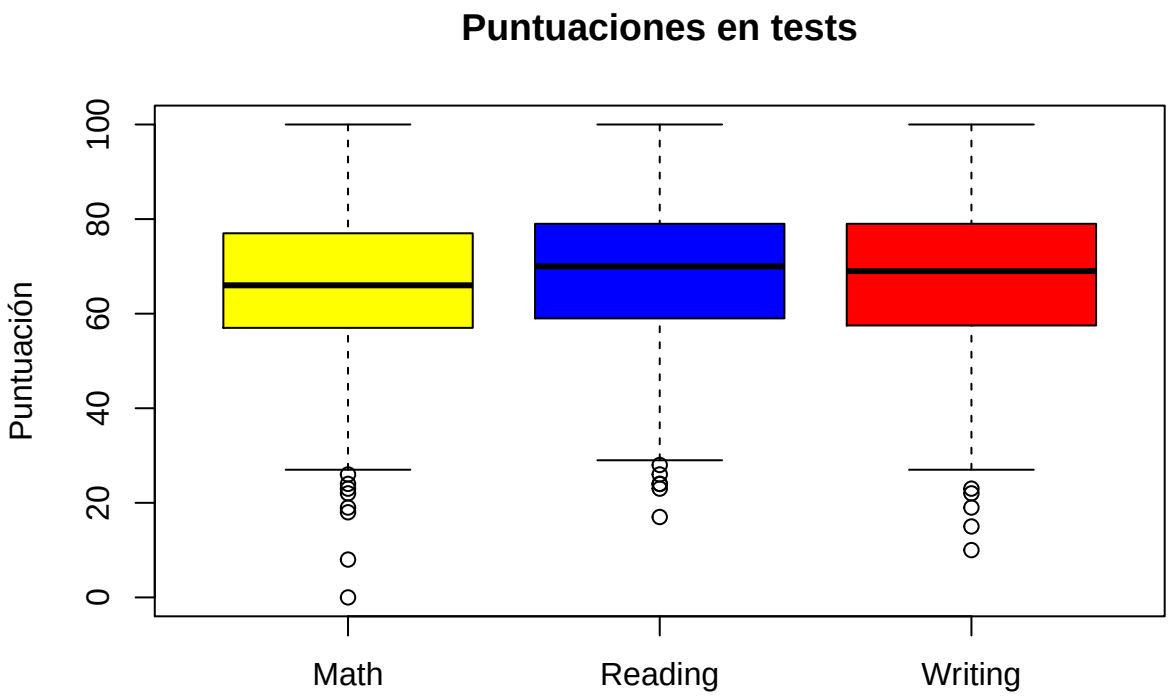
Queremos tener un dataframe sin valores outliers, por lo que vamos a crear una función que acota los datos hasta que no queden y nos devuelva el valor outlier máximo:

```
find_outliers = function(column){

  k <- 1
  parar <- FALSE

  mres <- NULL

  maxOut <- NULL

  mout <- NULL

  while(!parar){
    stats <- boxplot.stats(column)

    if (length(stats$out) == 0) {
      parar <- TRUE
    } else {
      maxOut <- max(stats$out)
      column <- column[column > maxOut]
    }

    vres <- c(k,0,maxOut,parar)

    vout <- c(stats$out)

    if (is.null(mout)) {
        mout <- vout
      } else {
        mout <- rbind(mout,vout)
      }

    if (is.null(mres)) {
        mres <- vres
      } else {
        mres <- rbind(mres,vres)
      }

    if(!parar) {
      k <- k+1
    }

  }


  if(k == 1) {
    print("No se han encontrado valores outliers")
    return(0)
  } else {
    print("Se han encontrado valores outliers")
    colnames(mres) = c("k","outliers","maxOut","Fin")
    rownames(mres) = NULL

    dfres <- as.data.frame(mres)
```

```
    if(k == 2){
      dfres$outliers[1] <- list(mout)
    } else {
      colnames(mout) = NULL
      rownames(mout) = NULL
      for (i in 1:k-1) {
        dfres$outliers[i] <- list(mout[i,])
      }
    }


    return(dfres)
  }

}
```

Comprobamos si existen valores outliers en las puntuaciones de los tests mediante un gráfico de caja y bigotes que genera la función "boxplot":

```
boxplot(data[,6:8], main = "Puntuaciones en tests", ylab = "Puntuación", col =
          c("yellow","blue","red"))
```



Como se puede observar, en los tres tests se encuentran valores outliers, por lo que vamos tratar cada caso individualmente.

3.3.1. Tests de matemáticas

Vamos a buscar cuáles son los outliers en los tests de matemáticas mediante la función "boxplot.stats":

```
statsMath <- boxplot.stats(data$Math)
statsMath
```

```
## $stats
## [1]  27  57  66  77 100
##
## $n
## [1] 1000
##
## $conf
## [1] 65.00072 66.99928
##
## $out
## [1] 18  0 22 24 26 19 23  8
```

Como podemos observar, los valores outliers en las puntuaciones del test de matemáticas son: {18, 0, 22, 24, 26, 19, 23, 8}. Por lo que ahora vamos a aplicar la función para enncontrar el valor outlier máximo:

```
resMathOut = find_outliers(data$Math)
```

```
## [1] "Se han encontrado valores outliers"
```

```
n.Math = nrow(resMathOut)
knitr::kable(resMathOut)
```

k	outliers	maxOut	Fin
1	18, 0, 22, 24, 26, 19, 23, 8	26	0
2	0	26	1

El valor outlier máximo en las puntuaciones del test de matemáticas es: 26.

3.3.2. Tests de lectura

Vamos a continuar buscando los outliers en los tests de lectura:

```
statsReading <- boxplot.stats(data$Reading)
statsReading
```

```
## $stats
## [1]  29  59  70  79 100
##
## $n
## [1] 1000
##
## $conf
## [1] 69.00072 70.99928
##
## $out
## [1] 17 26 28 23 24 24
```

Como podemos observar, los valores outliers en las puntuaciones del test de lectura son: {17, 26, 28, 23, 24, 24}. Por lo que ahora vamos a aplicar la función para enncontrar el valor outlier máximo:

```
resReadingOut = find_outliers(data$Reading)
```

```
## [1] "Se han encontrado valores outliers"
```

```
n.Reading = nrow(resReadingOut)
knitr::kable(resReadingOut)
```

k	outliers	maxOut	Fin
1	17, 26, 28, 23, 24, 24	28	0
2	29, 29, 29, 29, 29, 29	29	0
3	0	29	1

El valor outlier máximo en las puntuaciones del test de lectura es: 29.

3.3.3. Tests de redacción

Vamos a finalizar la búsqueda de los outliers en los tests de redacción:

```
statsWriting <- boxplot.stats(data$Writing)
statsWriting
```

```
## $stats
## [1]  27.0  57.5  69.0  79.0 100.0
##
## $n
## [1] 1000
##
## $conf
## [1] 67.92577 70.07423
##
## $out
## [1] 10 22 19 15 23
```

Como podemos observar, los valores outliers en las puntuaciones del test de redacción son: {10, 22, 19, 15, 23}. Por lo que ahora vamos a aplicar la función para enncontrar el valor outlier máximo:

```
resWritingOut = find_outliers(data$Writing)
```

```
## [1] "Se han encontrado valores outliers"
```

```
n.Writing = nrow(resWritingOut)
knitr::kable(resWritingOut)
```

k	outliers	maxOut	Fin
1	10, 22, 19, 15, 23	23	0
2	0	23	1

El valor outlier máximo en las puntuaciones del test de redacción es: 23.

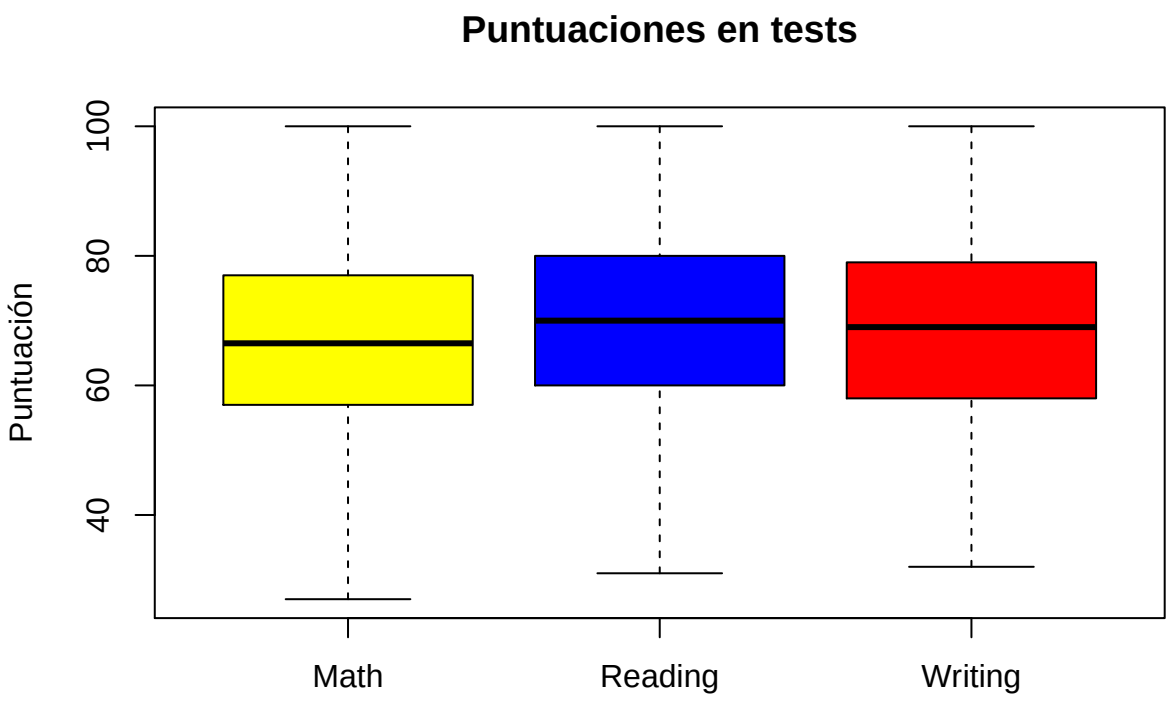
3.3.4. Eliminación de valores outliers

Por último, vamos a quedarnos con el resto de los valores:

```
data_sin_mathOut <- data[data$Math > resMathOut$maxOut[n.Math],]

data_sin_readingOut <- data_sin_mathOut[data_sin_mathOut$Reading >
                                          resReadingOut$maxOut[n.Reading],]

final_data <- data_sin_readingOut[data_sin_readingOut$Writing >
                                    resWritingOut$maxOut[n.Writing],]
```

Comprobamos que ya no haya valores outliers:

```
boxplot(final_data[,6:8], main = "Puntuaciones en tests", ylab = "Puntuación", col =
          c("yellow","blue","red"))
```




```
# Stats finales de matemáticas
boxplot.stats(final_data$Math)
```

```
## $stats
## [1]  27.0  57.0  66.5  77.0 100.0
```

```
##
## $n
## [1] 986
##
## $conf
## [1] 65.49365 67.50635
##
## $out
## integer(0)
```

```
# Stats finales de lectura
boxplot.stats(final_data$Reading)
```

```
## $stats
## [1]  31  60  70  80 100
##
## $n
## [1] 986
##
## $conf
## [1] 68.99365 71.00635
##
## $out
## integer(0)
```

```
# Stats finales de redacción
boxplot.stats(final_data$Writing)
```

```
## $stats
## [1]  32  58  69  79 100
##
## $n
## [1] 986
##
## $conf
## [1] 67.94333 70.05667
##
## $out
## integer(0)
```

Finalizamos con el resumen del nuevo dataframe sin valores outliers:

```
summary(final_data)
```

```
##     Gender      Ethnicity             Parent_Education          Lunch
##  female:509   group A: 88   associate's degree:221     free/reduced:344
##  male  :477   group B:183   bachelor's degree :118     standard    :642
##               group C:315   high school       :191
##               group D:261   master's degree   : 59
##               group E:139   some college      :222
##                             some high school  :175
##     Preparation       Math           Reading          Writing
```

```
##  completed:357   Min.   : 27.00   Min.   : 31.00   Min.   : 32.00
##  none     :629   1st Qu.: 57.00   1st Qu.: 60.00   1st Qu.: 58.00
##                  Median : 66.50   Median : 70.00   Median : 69.00
##                  Mean   : 66.69   Mean   : 69.72   Mean   : 68.65
##                  3rd Qu.: 77.00   3rd Qu.: 80.00   3rd Qu.: 79.00
##                  Max.   :100.00   Max.   :100.00   Max.   :100.00
```

Nuestro data frame final tendría 986 filas.

Antes de terminar con este apartado, vamos a guardar de nuevo nuestro data frame en un archivo RData:

```
save(final_data, file = "final_data.RData")
```

4. Transformación

4.1. Discretización

Vamos a comprobar cuál es el tipo de nuestro valores numéricos utilizando la función "str":

```
str(final_data[,6:8])
```

```
## 'data.frame':    986 obs. of  3 variables:
##  $ Math   : int  72 69 90 47 76 71 88 40 64 38 ...
##  $ Reading: int  72 90 95 57 78 83 95 43 64 60 ...
##  $ Writing: int  74 88 93 44 75 78 92 39 67 50 ...
```

Como se puede observar, los 3 tipos de puntuaciones son del tipo entero, por lo que no pueden contener decimales. Por lo tanto, se tratan de variables discretas y no resultaría necesario el proceso de discretización.