

# PRA2 Limpieza y análisis

Luis Piñuela y Eduardo Ranedo

2023-06-13

Primero leemos nuestro dataset y seleccionamos aquellas variables con las que vamos a trabajar. Eliminaremos 3 variables; "X", "LunchType" y "TestPrep".

```
data <- read.csv("datasetpra2.csv", sep=',')  
  
# Vemos cada una de las variables con el tipo de datos que contienen  
str(data)  
  
## 'data.frame': 30641 obs. of 15 variables:  
## $ X : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ Gender : chr "female" "female" "female" "male" ...  
## $ EthnicGroup : chr "" "group C" "group B" "group A" ...  
## $ ParentEduc : chr "bachelor's degree" "some college" "master's degree" "associate's degree"  
## $ LunchType : chr "standard" "standard" "standard" "free/reduced" ...  
## $ TestPrep : chr "none" "" "none" "none" ...  
## $ ParentMaritalStatus: chr "married" "married" "single" "married" ...  
## $ PracticeSport : chr "regularly" "sometimes" "sometimes" "never" ...  
## $ IsFirstChild : chr "yes" "yes" "yes" "no" ...  
## $ NrSiblings : int 3 0 4 1 0 1 1 1 3 NA ...  
## $ TransportMeans : chr "school_bus" "" "school_bus" "" ...  
## $ WklyStudyHours : chr "< 5" "5 - 10" "< 5" "5 - 10" ...  
## $ MathScore : int 71 69 87 45 76 73 85 41 65 37 ...  
## $ ReadingScore : int 71 90 93 56 78 84 93 43 64 59 ...  
## $ WritingScore : int 74 88 91 42 75 79 89 39 68 50 ...  
  
# Eliminamos columnas innecesarias  
dataset <- subset(data, select = -c(X, LunchType))  
str(dataset)  
  
## 'data.frame': 30641 obs. of 13 variables:  
## $ Gender : chr "female" "female" "female" "male" ...  
## $ EthnicGroup : chr "" "group C" "group B" "group A" ...  
## $ ParentEduc : chr "bachelor's degree" "some college" "master's degree" "associate's degree"  
## $ TestPrep : chr "none" "" "none" "none" ...  
## $ ParentMaritalStatus: chr "married" "married" "single" "married" ...  
## $ PracticeSport : chr "regularly" "sometimes" "sometimes" "never" ...  
## $ IsFirstChild : chr "yes" "yes" "yes" "no" ...  
## $ NrSiblings : int 3 0 4 1 0 1 1 1 3 NA ...  
## $ TransportMeans : chr "school_bus" "" "school_bus" "" ...  
## $ WklyStudyHours : chr "< 5" "5 - 10" "< 5" "5 - 10" ...  
## $ MathScore : int 71 69 87 45 76 73 85 41 65 37 ...  
## $ ReadingScore : int 71 90 93 56 78 84 93 43 64 59 ...  
## $ WritingScore : int 74 88 91 42 75 79 89 39 68 50 ...  
  
#Limpieza de datos Vamos a ver el número de valores NA y vacíos que contiene nuestro dataset, para ver
```

como vamos a tratarlos

```
colSums(is.na(dataset))

##           Gender      EthnicGroup    ParentEduc     TestPrep
##             0              0              0              0
## ParentMaritalStatus PracticeSport IsFirstChild NrSiblings
##             0              0              0              1572
## TransportMeans      WklyStudyHours MathScore   ReadingScore
##             0                  0              0              0
## WritingScore          WritingScore
##             0
```

Hay 1572 valores NA en la variable NrSiblings

```
colSums(dataset == "")

##           Gender      EthnicGroup    ParentEduc     TestPrep
##             0              1840            1845            1830
## ParentMaritalStatus PracticeSport IsFirstChild NrSiblings
##             1190            631             904             NA
## TransportMeans      WklyStudyHours MathScore   ReadingScore
##             3134            955              0              0
## WritingScore          WritingScore
##             0
```

Vemos que hay bastantes valores en blanco en diferentes variables. Como tenemos una muestra bastante grande, hemos considerado que vamos a eliminar todos los registros que tengan tanto NA como valores en blanco en cualquiera de las variables

```
dataset <- dataset[!rowSums(is.na(dataset) | dataset == "") > 0, ]
nrow(dataset)
```

```
## [1] 19243
```

Hemos reducido nuestro número de registros de 30641 a 20445.

Vamos a dejar preparadas las variables, de tal manera que posteriormente podamos utilizarlas para realizar el análisis.

Modificamos los valores de la variable “EthnicGroup” para hacerlos más entendibles, ya que actualmente se encuentran en una nomenclatura especial utilizada en Estados Unidos.

```
equivalencias <- c("group A" = "Americana",
                    "group B" = "Negra",
                    "group C" = "Asiática",
                    "group D" = "Hispana",
                    "group E" = "Caucásica")
```

```
dataset$EthnicGroup <- equivalencias[dataset$EthnicGroup]
```

Ahora vamos a dicotomizar algunas de las variables:

- “IsFirstChild”: Tomará el valor 0 (sustituye a “yes”) en caso de que sea el primer hijo y 1 (sustituye a “no”) en caso de que no lo sea
- “NrSiblings”: Tomará el valor 0 en caso de que sea hijo único (no es necesario hacer ningún cambio) y 1 en caso de que no lo sea (habrá que asignar 1 a todos los valores distintos de 0)
- “TransportMeans”: Tomará el valor 0 en caso de que el desplazamiento sea mediante autobús escolar (school\_bus) y tomará valor 1 cuando el desplazamiento sea privado (private)

```

# Variable IsFirstChild
dataset$IsFirstChild <- ifelse(dataset$IsFirstChild == "yes", 0, 1)

# Variable NrSiblings
dataset$NrSiblings <- ifelse(dataset$NrSiblings == 0, 0, 1)

# Variable TransportMeans
dataset$TransportMeans <- ifelse(dataset$TransportMeans == "school_bus", 0, 1)

```

Para poder tener una visión general de las notas de los alumnos, vamos a crear una nueva variable que será la media del alumno de las puntuaciones obtenidas en MathScore, ReadingScore y WritingScore

```
dataset$OverallScore <- round((dataset$MathScore + dataset$ReadingScore + dataset$WritingScore) / 3)
```

Podemos observar que los datos de la variable WklyStudyHours tiene unos datos que se han interpretado incorrectamente como fechas, vamos a modificarlo para adaptarlo a los valores correctos.

```
dataset$WklyStudyHours <- ifelse(dataset$WklyStudyHours == "5-oct", '5-10', dataset$WklyStudyHours)
```

#Analisis de outliers y otras anomalías

Tras preparar los datos vamos a hacer un análisis general de las variables para detectar outliers y otras anomalías.

```
summary(dataset)
```

```

##      Gender          EthnicGroup        ParentEduc        TestPrep
##  Length:19243      Length:19243      Length:19243      Length:19243
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
## 
## 
## 
##  ParentMaritalStatus PracticeSport        IsFirstChild        NrSiblings
##  Length:19243          Length:19243      Min.   :0.0000      Min.   :0.0000
##  Class :character      Class :character  1st Qu.:0.0000     1st Qu.:1.0000
##  Mode  :character      Mode  :character  Median :0.0000     Median :1.0000
##                               Mean   :0.3547     Mean   :0.8964
##                               3rd Qu.:1.0000    3rd Qu.:1.0000
##                               Max.  :1.0000    Max.  :1.0000
## 
## 
## 
##  TransportMeans    WklyStudyHours       MathScore       ReadingScore
##  Min.   :0.0000    Length:19243      Min.   : 0.00  Min.   : 10.00
##  1st Qu.:0.0000    Class :character  1st Qu.: 56.00  1st Qu.: 59.00
##  Median :0.0000    Mode  :character  Median : 67.00  Median : 70.00
##  Mean   :0.4138    Mean   :66.64    Mean   :66.53  Mean   :69.53
##  3rd Qu.:1.0000    3rd Qu.: 78.00  3rd Qu.: 80.00  3rd Qu.: 80.00
##  Max.   :1.0000    Max.   :100.00  Max.   :100.00  Max.   :100.00
## 
##  WritingScore      OverallScore
##  Min.   : 4.0      Min.   : 9.00
##  1st Qu.: 58.0    1st Qu.: 58.00
##  Median : 69.0    Median : 68.00
##  Mean   : 68.6    Mean   : 68.25
##  3rd Qu.: 80.0    3rd Qu.: 79.00
##  Max.   :100.0    Max.   :100.00

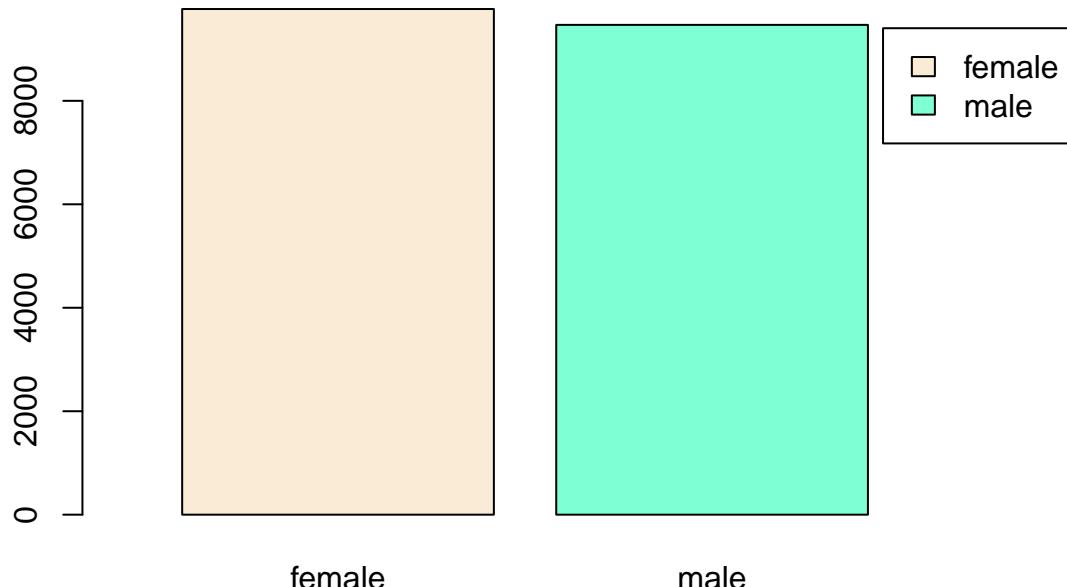
```

En principio parece que las variables numéricas son correctas ya que todas las puntuaciones se encuentran en el rango de 0 a 100, por lo que no habría ningún error en sus valores extremos pero estudiemos sus

distribuciones.

### Comprobación de Normalidad y homogeneidad de la varianza en Gender

```
#Histograma para normalidad
datos <- table(dataset$Gender)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A"),
        xlim = c(0, 3))
```



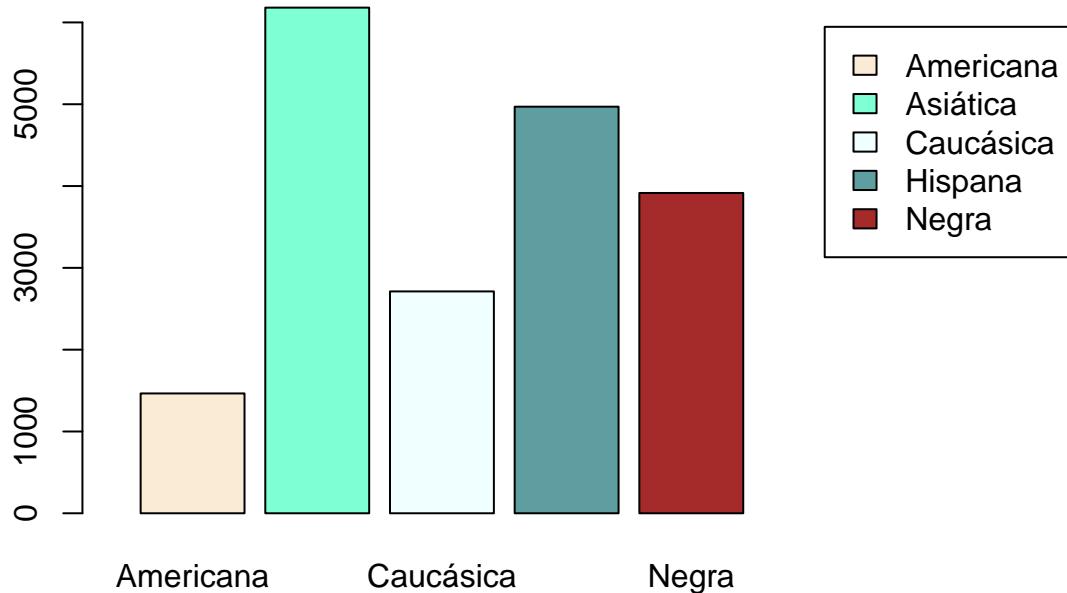
```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))
```

```
##
##  Chi-squared test for given probabilities
##
##  data:  datos
##  X-squared = 4.8978, df = 1, p-value = 0.02689
```

### Comprobación de Normalidad y homogeneidad de la varianza en EthnicGroup

```
#Histograma para normalidad
datos <- table(dataset$EthnicGroup)
barplot(datos,
        legend = rownames(datos),
```

```
col=c("#FAEBD7", "aquamarine", "#FOFFF", "#5F9EA0", "#A52A2A"),
      xlim = c(0, 9))
```

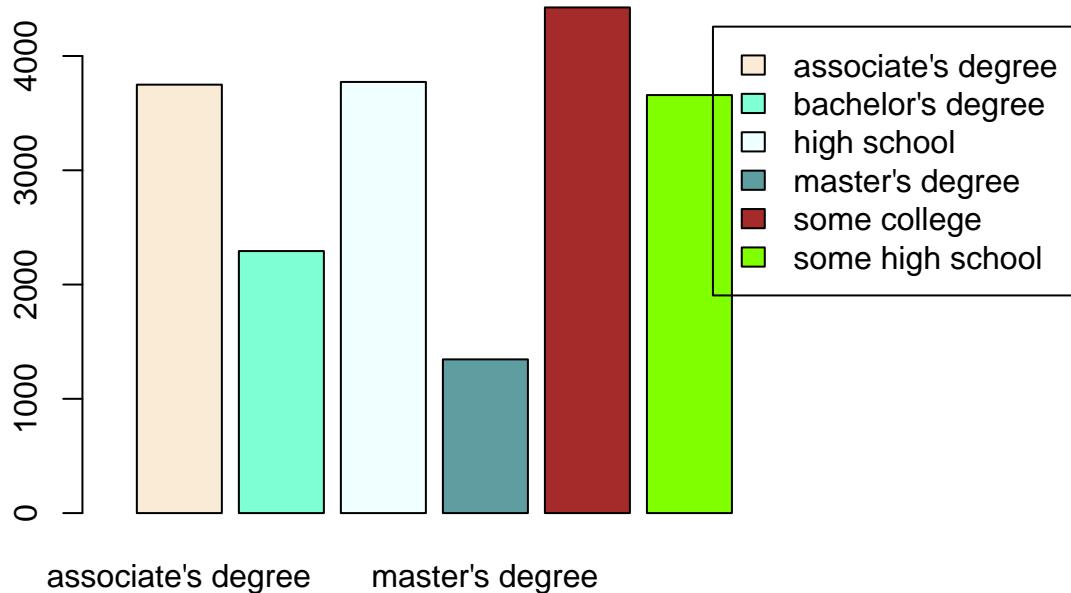


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))
```

```
##
## Chi-squared test for given probabilities
##
## data: datos
## X-squared = 3553.4, df = 4, p-value < 2.2e-16
```

## Comprobación de Normalidad y homogeneidad de la varianza en ParentEduc

```
#Histograma para normalidad
datos <- table(dataset$ParentEduc)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 11))
```

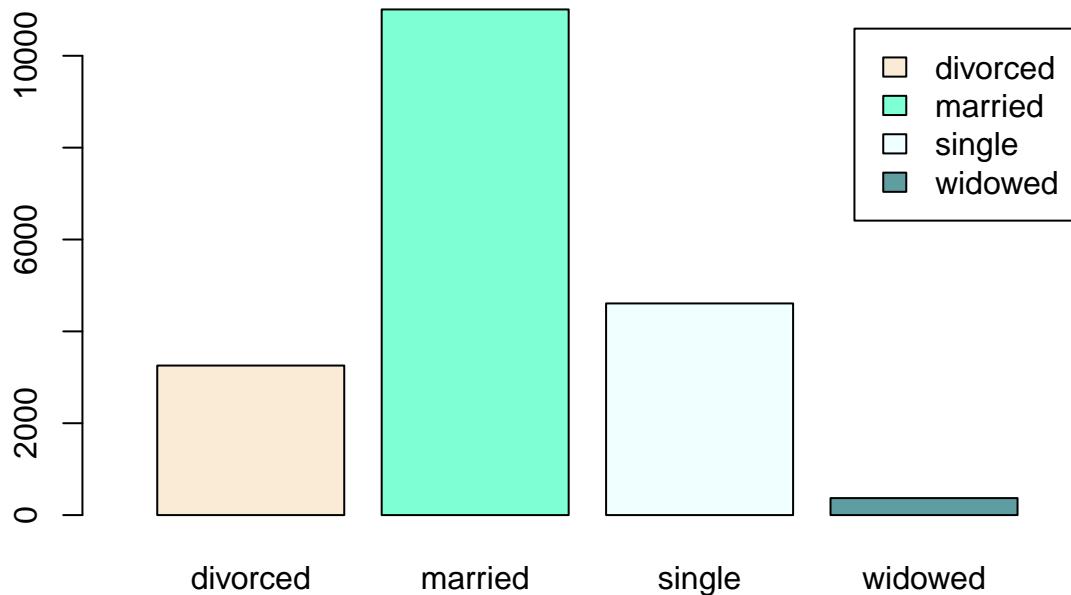


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
## Chi-squared test for given probabilities
##
## data: datos
## X-squared = 2059, df = 5, p-value < 2.2e-16
```

## Comprobación de Normalidad y homogeneidad de la varianza en ParentMaritalStatus

```
#Histograma para normalidad
datos <- table(dataset$ParentMaritalStatus)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 5))
```

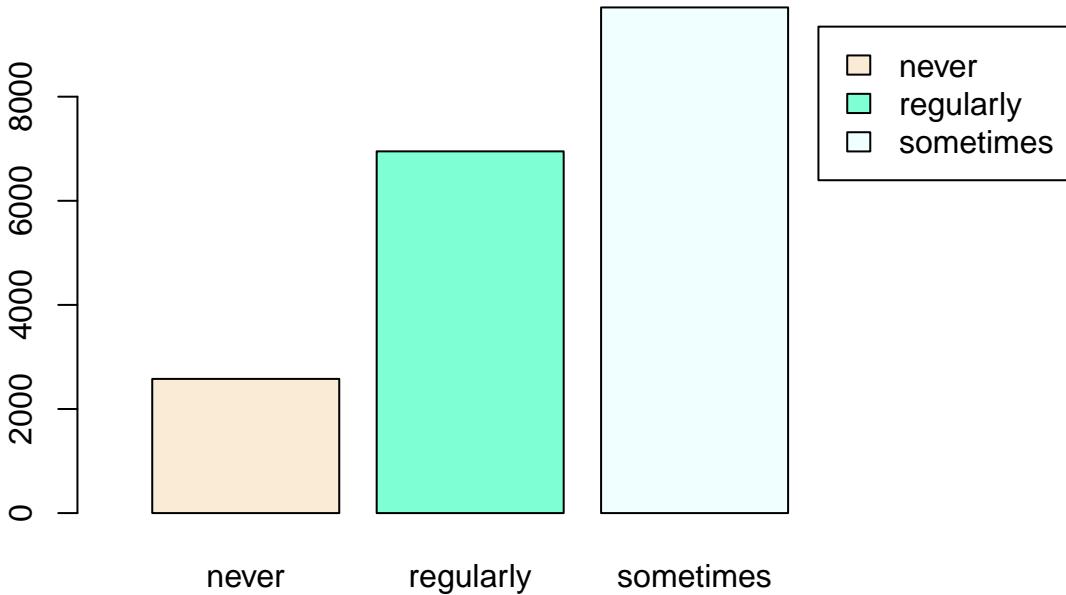


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
## Chi-squared test for given probabilities
##
## data: datos
## X-squared = 12596, df = 3, p-value < 2.2e-16
```

## Comprobación de Normalidad y homogeneidad de la varianza en PracticeSport

```
#Histograma para normalidad
datos <- table(dataset$PracticeSport)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 5))
```

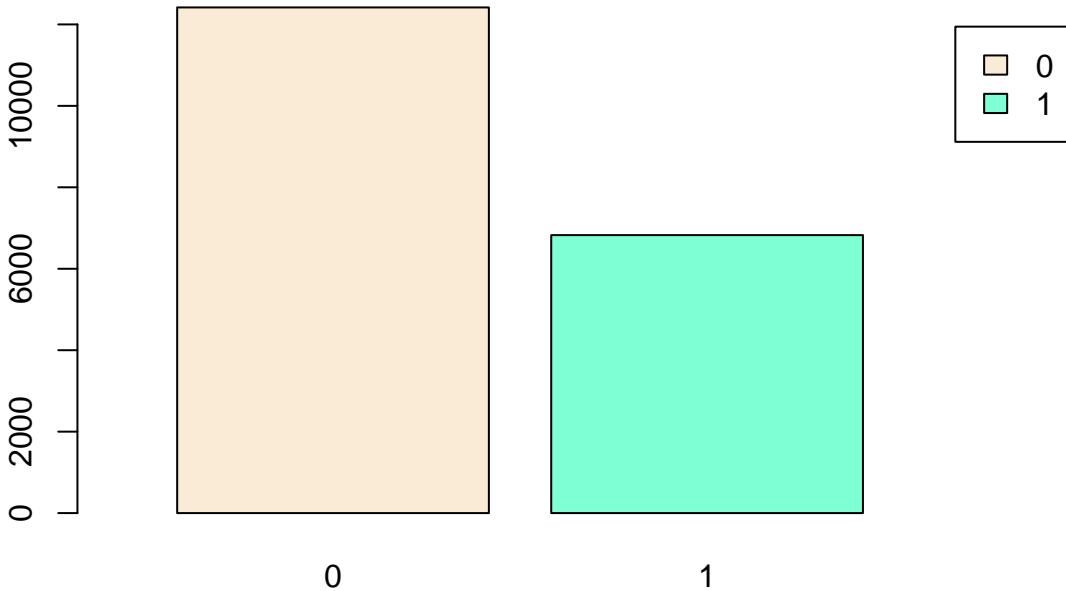


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
##  Chi-squared test for given probabilities
##
## data:  datos
## X-squared = 4037.6, df = 2, p-value < 2.2e-16
```

### Comprobación de Normalidad y homogeneidad de la varianza en IsFirstChild

```
#Histograma para normalidad
datos <- table(dataset$IsFirstChild)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 3))
```

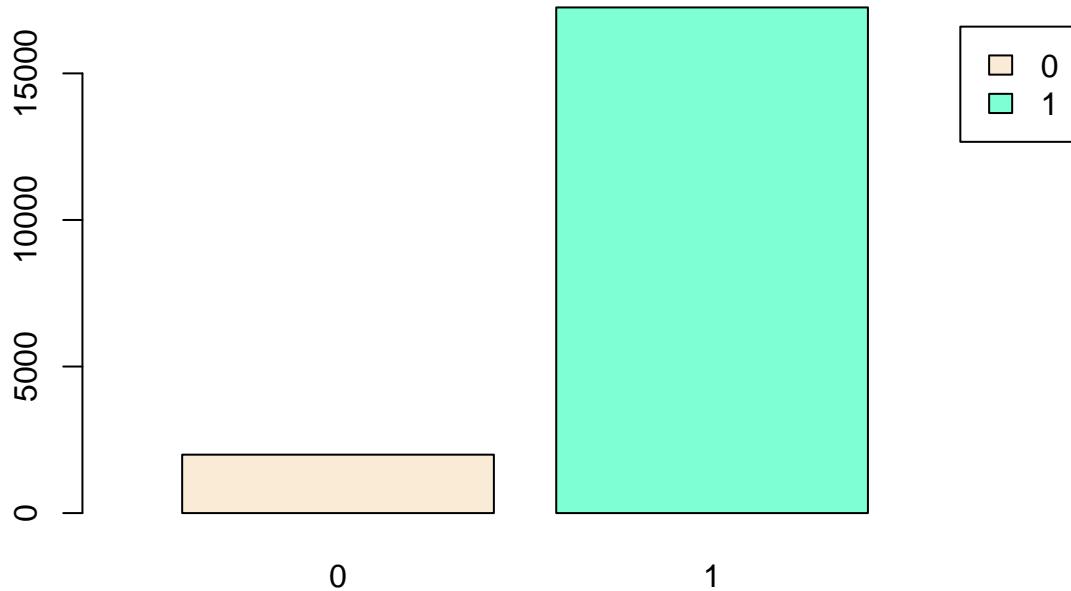


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
##  Chi-squared test for given probabilities
##
## data:  datos
## X-squared = 1624.4, df = 1, p-value < 2.2e-16
```

### Comprobación de Normalidad y homogeneidad de la varianza en NrSiblings

```
#Histograma para normalidad
datos <- table(dataset$NrSiblings)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 3))
```

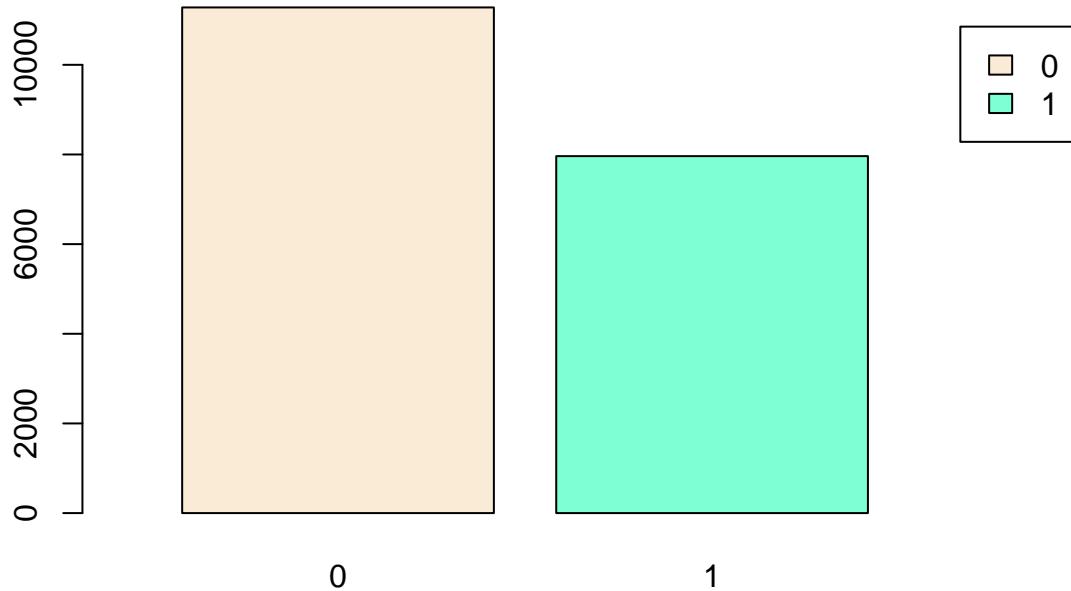


```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
##  Chi-squared test for given probabilities
##
## data:  datos
## X-squared = 12097, df = 1, p-value < 2.2e-16
```

### Comprobación de Normalidad y homogeneidad de la varianza en TransportMeans

```
#Histograma para normalidad
datos <- table(dataset$TransportMeans)
barplot(datos,
        legend = rownames(datos),
        col=c("#FAEBD7", "aquamarine", "#FOFFFF", "#5F9EA0", "#A52A2A", "#7FFF00"),
        xlim = c(0, 3))
```



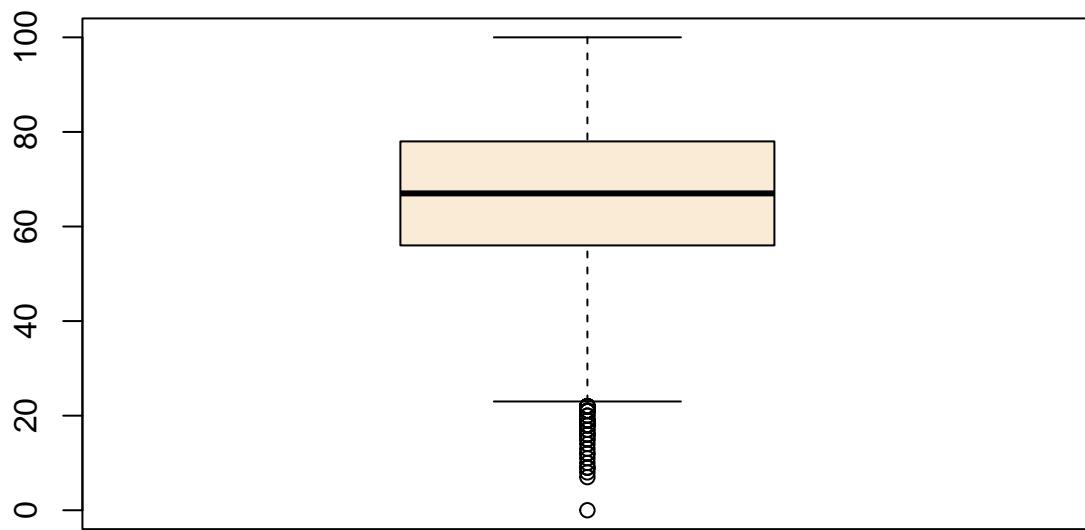
```
# Test chi-cuadrado para Homogeniedad
print(chisq.test(datos))

##
##  Chi-squared test for given probabilities
##
## data: datos
## X-squared = 571.77, df = 1, p-value < 2.2e-16
```

### Comprobación de Normalidad y homogeneidad de la varianza en MathScore

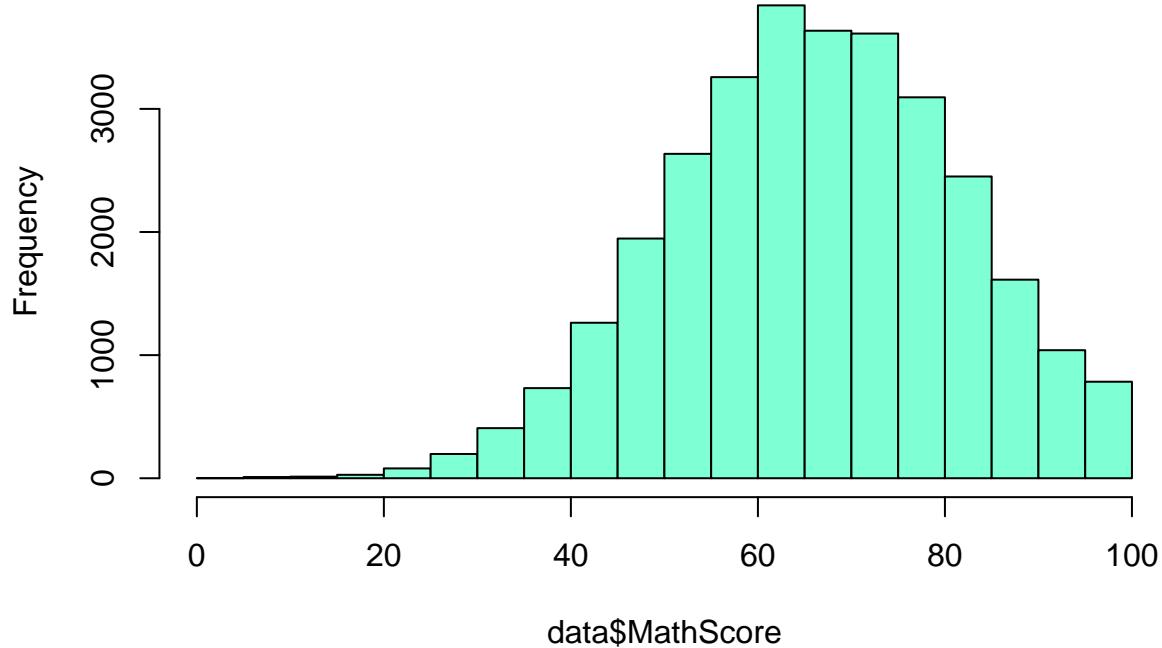
```
boxplot(dataset$MathScore, main="Variable MathScore", col="#FAEBD7")
```

## Variable MathScore



```
hist(data$MathScore, main="Distribución MathScore", col="aquamarine")
```

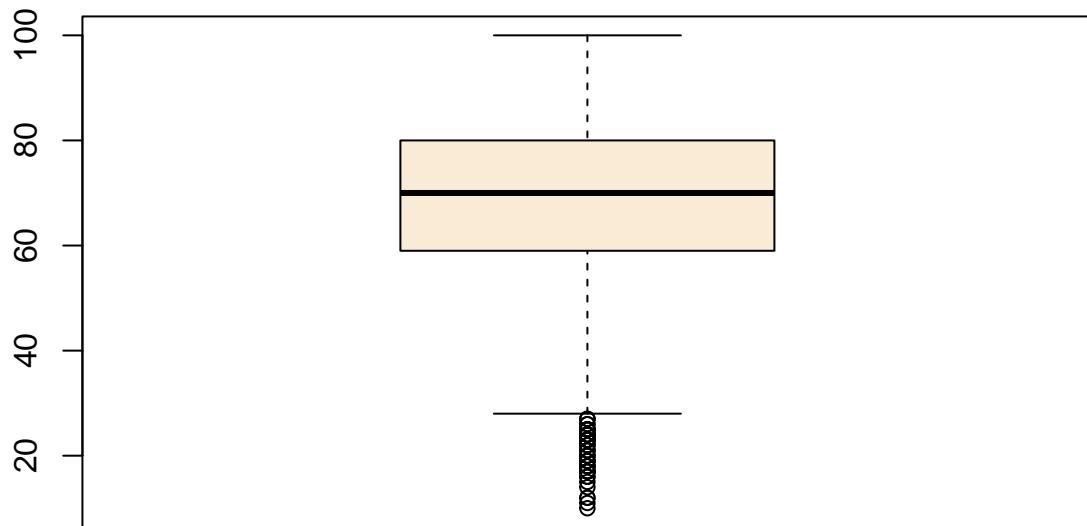
## Distribución MathScore



Comprobación de Normalidad y homogeneidad de la varianza en ReadingScore

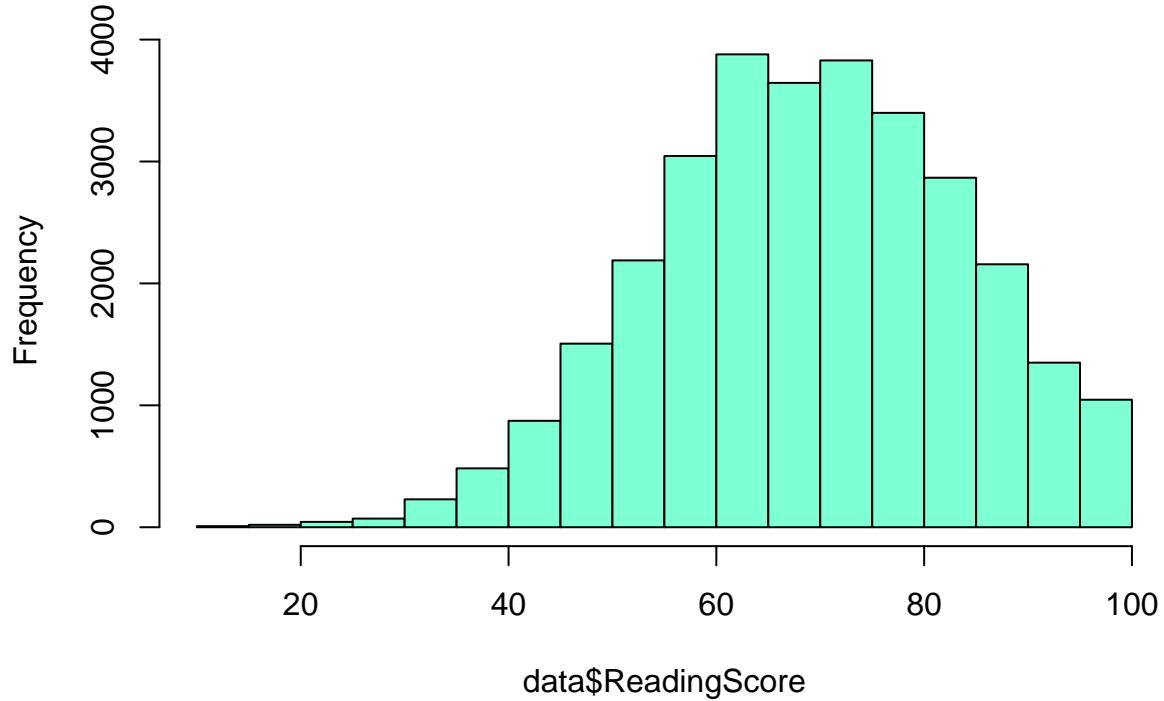
```
boxplot(dataset$ReadingScore, main="Variable ReadingScore", col="#FAEBD7")
```

## Variable ReadingScore



```
hist(data$ReadingScore, main="Distirbución ReadingScore", col="aquamarine")
```

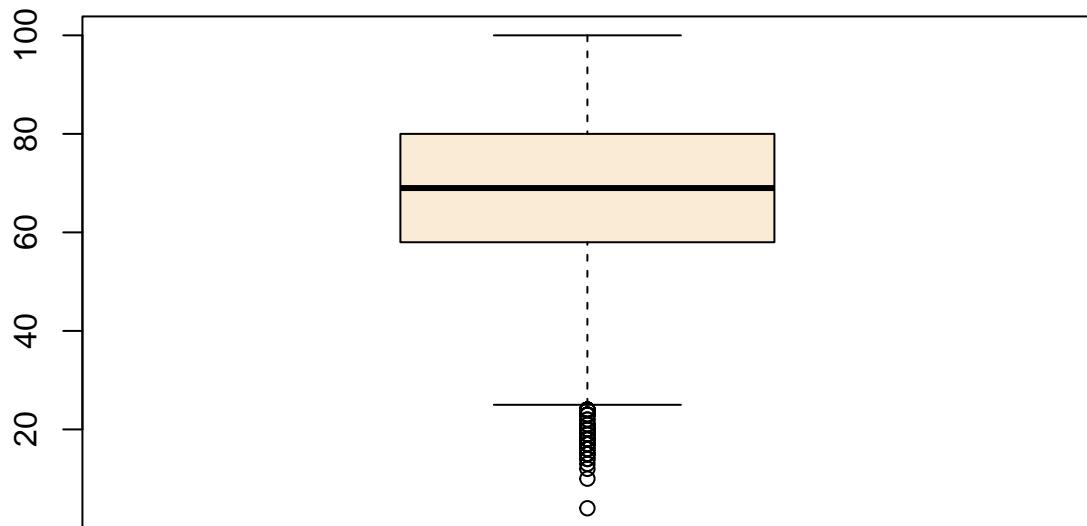
## Distirbución ReadingScore



Comprobación de Normalidad y homogeneidad de la varianza en WritingScore

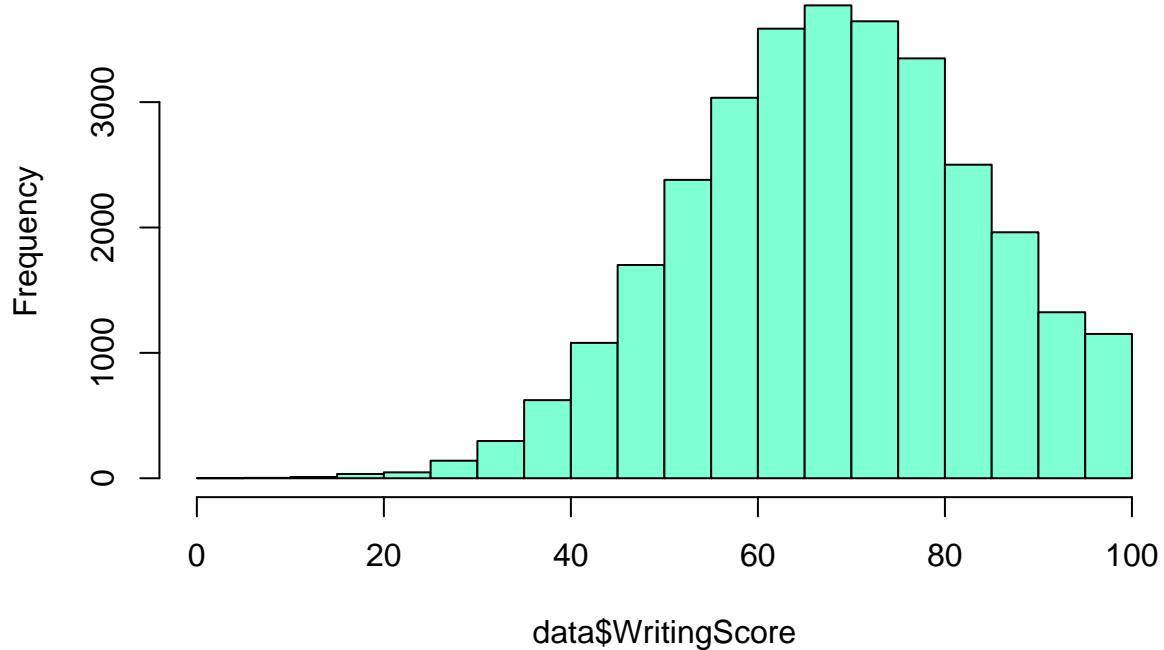
```
boxplot(dataset$WritingScore, main="Variable WritingScore", col="#FAEBD7")
```

## Variable WritingScore



```
hist(data$WritingScore, main="Distirbución WritingScore", col="aquamarine")
```

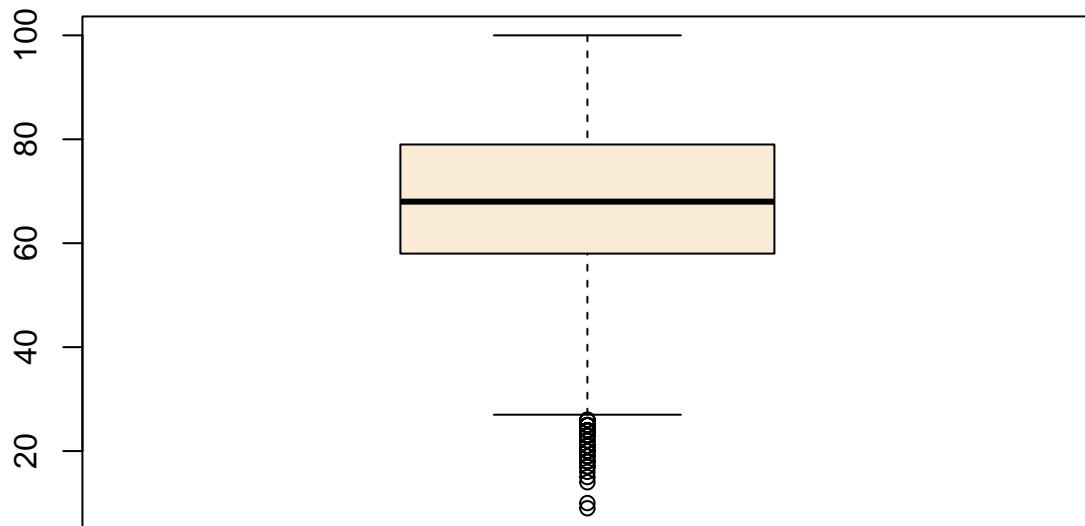
## Distirbución WritingScore



Comprobación de Normalidad y homogeneidad de la varianza en OverallScore

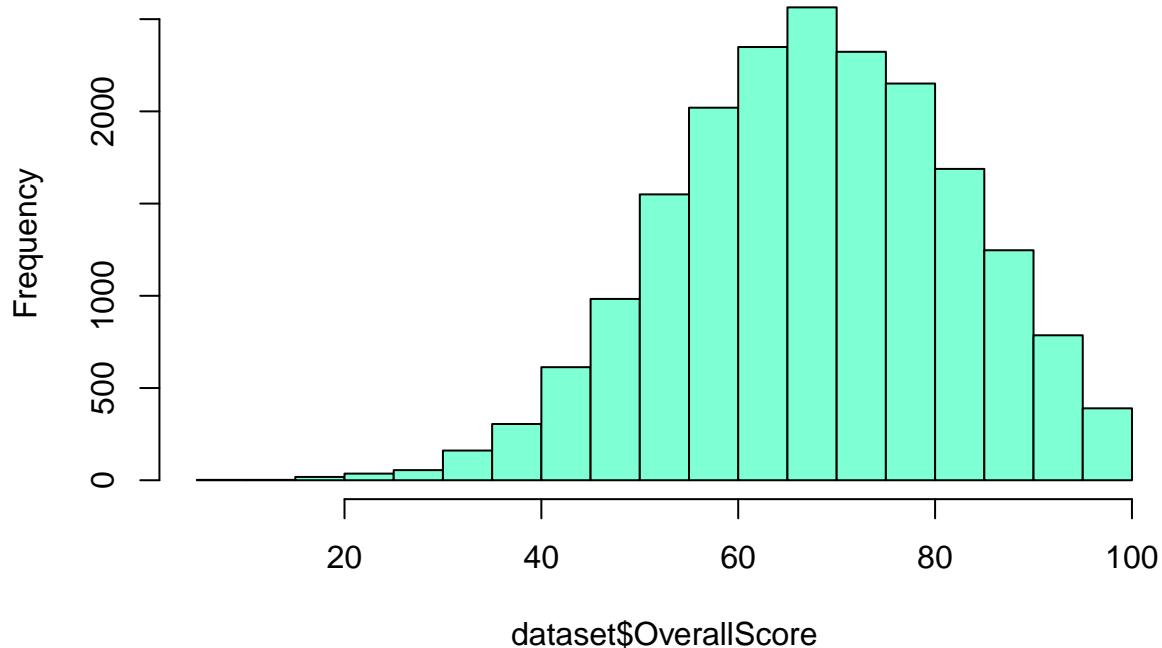
```
boxplot(dataset$OverallScore, main="Variable OverallScore", col="#FAEBD7")
```

## Variable OverallScore



```
hist(dataset$OverallScore, main="Distribución OverallScore", col="aquamarine")
```

## Distribución OverallScore



Podemos ver que la distribución de las variables se parece a la de una normal y que las puntuaciones bajas actúan como outliers. No vemos necesario aplicar ninguna medida respecto a los outliers, ya que parecen ser datos realistas dentro del rango de notas posible.

##Aplicación de pruebas estadísticas para comparar los grupos de datos. Queremos comparar la variable OverallScore con diferentes variables para tratar de entender cuál es el perfil de los estudiantes con mejores notas y los que tienen un peor rendimiento académico.

Primero tendríamos analizar la normalidad de la variable OverallScore. En este caso no podríamos utilizar la función Shapiro.test ya que nuestra muestra tiene mas de 5.000 registros y este test no adminte muestras superiores a esa cifra. Por tanto, tras comprobar en el histograma que la distribución se asemeja a la de una normal y basándonos en el teorema del límite central, que indica que para muestras grandes como la nuestra se seguirá una distribución normal, podemos concluir que la variable OverallScore sigue una distribución normal.

```
if (!requireNamespace("car", quietly = TRUE)) {install.packages("car")}  
library(car)  
  
## Loading required package: carData  
levene_genero <- leveneTest(OverallScore ~ Gender, data = dataset)  
  
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.  
print(levene_genero)  
  
## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)
```

```
## group      1  1.6648  0.197
##          19241
```

Podemos ver que el p-valor (0.2235) es superior al nivel de significancia (0.05) por lo que aceptamos la hipótesis nula de homocedasticidad.

Viendo que se cumplen los principios de normalidad y homocedasticidad, pasamos a realizar la prueba de t-Student con la variable género

```
t.test(OverallScore ~ Gender, data = dataset)
```

```
##
```

```
## Welch Two Sample t-test
```

```
##
```

```
## data: OverallScore by Gender
```

```
## t = 18.402, df = 19195, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means between group female and group male is not equal to
```

```
## 95 percent confidence interval:
```

```
## 3.404555 4.216310
```

```
## sample estimates:
```

```
## mean in group female   mean in group male
```

```
##           70.12972       66.31929
```

El p-valor es menor que el nivel de significancia, lo que muestra que si existen diferencias significativas entre los grupos y que la media de las mujeres es superior al de los hombres.

Ahora queremos saber si existen diferencias significativas entre OverallScore y el grupo étnico del estudiante. Por ello, realizaremos un ANOVA.

```
res.aov <- aov(OverallScore ~ EthnicGroup, data = dataset)
```

```
summary(res.aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
## EthnicGroup	4	164359	41090	204.2	<2e-16 ***						
## Residuals	19238	3871256	201								
## ---											
## Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'. '	0.1	' '	1

Podemos ver que el p valor es menor al nivel de significancia por lo que podemos concluir que existen diferencias significativas en la variable OverallScore respecto a EthnicGroup.

Vamos a comparar la variable ParentEducation con OverallScore

```
res.aov <- aov(OverallScore ~ ParentEduc, data = dataset)
```

```
summary(res.aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
## ParentEduc	5	213865	42773	215.3	<2e-16 ***						
## Residuals	19237	3821750	199								
## ---											
## Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'. '	0.1	' '	1

Vemos que el p valor es menor al nivel de significancia por lo que podemos concluir que existen diferencias significativas en la variable OverallScore respecto a ParentEduc.

Probamos con la variable ParentMaritalStatus

```
res.aov <- aov(OverallScore ~ ParentMaritalStatus, data = dataset)
```

```
summary(res.aov)
```

```
##
```

```

## ParentMaritalStatus      3     896   298.7   1.424  0.234
## Residuals             19239 4034719   209.7

```

No se han encontrado asociaciones entre el estatus civil de los padres y la variable OverallScore.

Vemos ahora con la variable PracticeSport

```

res.aov <- aov(OverallScore ~ PracticeSport, data = dataset)
summary(res.aov)

```

```

##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## PracticeSport      2  10506   5253   25.11 1.29e-11 ***
## Residuals        19240 4025109     209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Existe una relación significativa entre la práctica de deporte y el rendimiento académico medido a través de la variable OverallScore.

Ahora queremos saber si el hecho de ser el primero de los hijos puede influir en OverallScore

```
t.test(OverallScore ~ IsFirstChild, data = dataset)
```

```

##
## Welch Two Sample t-test
##
## data: OverallScore by IsFirstChild
## t = 0.84036, df = 14302, p-value = 0.4007
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -0.2429003 0.6074855
## sample estimates:
## mean in group 0 mean in group 1
##       68.31956       68.13727

```

La t-student demuestra que no existe significancia entre ser el primer hijo y OverallScore. Veamos si hay significancia entre el número de hermanos y hermanas

```
t.test(OverallScore ~ NrSiblings, data = dataset)
```

```

##
## Welch Two Sample t-test
##
## data: OverallScore by NrSiblings
## t = 0.91131, df = 2495.9, p-value = 0.3622
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -0.3539159 0.9684835
## sample estimates:
## mean in group 0 mean in group 1
##       68.53036       68.22307

```

Tampoco hay significancia entre el número de hermanos y OverallScore

```
t.test(OverallScore ~ TransportMeans, data = dataset)
```

```

##
## Welch Two Sample t-test
##
## data: OverallScore by TransportMeans

```

```

## t = 0.366, df = 16865, p-value = 0.7144
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -0.3394163 0.4952718
## sample estimates:
## mean in group 0 mean in group 1
##       68.28715      68.20922

```

El medio de transporte para ir al centro educativo tampoco es significante

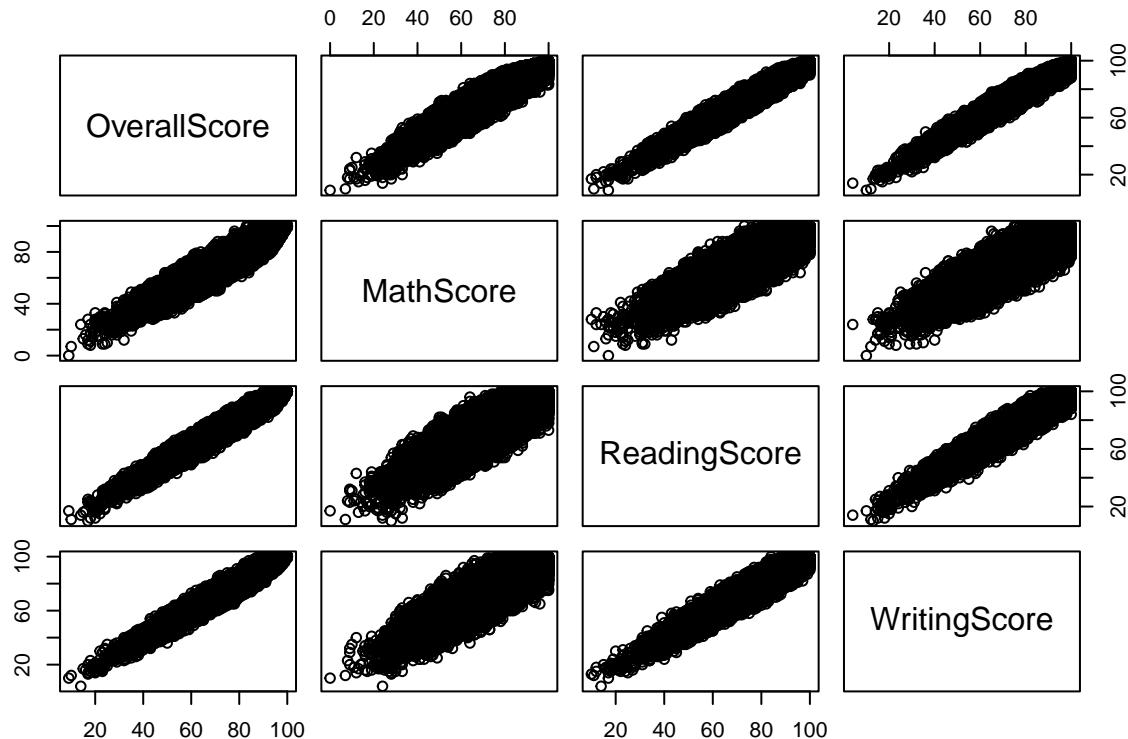
## Modelo de Regresión lineal simple para predecir las notas de los alumnos (3 variables)

```

# Creo un dataframe para el cálculo
dataset_lineal <- data.frame(c(dataset[,c("OverallScore", "MathScore", "ReadingScore", "WritingScore")])) 

# Proyecto para ver las relaciones
pairs(dataset_lineal)

```



```

# Relacion lineal
cor(dataset_lineal)

##          OverallScore MathScore ReadingScore WritingScore
## OverallScore    1.0000000  0.9203252   0.9693767   0.9664059
## MathScore       0.9203252  1.0000000   0.8189865   0.8085328
## ReadingScore    0.9693767  0.8189865   1.0000000   0.9526215

```

```

## WritingScore      0.9664059  0.8085328      0.9526215      1.0000000
# Ajustar el modelo de regresión lineal
modelo <- lm(OverallScore ~      + `MathScore` + `ReadingScore` + `WritingScore`, data = dataset_lineal)

# Imprimir los coeficientes del modelo
print(coef(modelo))

##  (Intercept)  MathScore ReadingScore WritingScore
## -0.001048677  0.333044475  0.333833406  0.333083447
# Realizar predicciones

nuevos_datos <- data.frame(c(dataset[,c("MathScore", "ReadingScore", "WritingScore")]))

predicciones <- predict(modelo, newdata = nuevos_datos)
head(predicciones)

##          1         2         3         4         5         6
## 90.33092 76.33060 78.66680 88.99867 40.99887 65.66185

```

## Modelo de Regresión logística simple para predecir las notas de los alumnos

```

# Generamos una columna para definir los sobresalientes como 85-100
dataset$OveralScore_sobre <- ifelse(dataset$OverallScore >= 85, 1, 0)

# Genero un subset para tratar
#dataset_logistico <- data.frame(c(dataset[,c("OverallScore", "MathScore", "ReadingScore", "WritingScore")], dataset$OveralScore_sobre))

# Generamos el modelo
modelo_glm <- glm(OveralScore_sobre ~ IsFirstChild + NrSiblings + TransportMeans + WklyStudyHours + MathScore,
                    family = "binomial")

## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
# Plasmamos los resultados
summary(modelo_glm)

## 
## Call:
## glm(formula = OveralScore_sobre ~ IsFirstChild + NrSiblings +
##     TransportMeans + WklyStudyHours + MathScore + ReadingScore +
##     WritingScore, family = "binomial", data = dataset)
## 
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -7.496e+03  3.582e+04  -0.209   0.834
## IsFirstChild        -9.726e-02  1.549e+02  -0.001   0.999
## NrSiblings           5.343e-01  2.349e+02   0.002   0.998
## TransportMeans       7.973e-02  1.451e+02   0.001   1.000
## WklyStudyHours> 10   6.047e-02  2.078e+02   0.000   1.000
## WklyStudyHours5 - 10 -4.818e-02  1.705e+02   0.000   1.000

```

```
## MathScore           2.957e+01  1.415e+02  0.209   0.834
## ReadingScore        2.957e+01  1.424e+02  0.208   0.836
## WritingScore        2.957e+01  1.421e+02  0.208   0.835
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1.5683e+04  on 19242  degrees of freedom
## Residual deviance: 1.4759e-04  on 19234  degrees of freedom
## AIC: 18
##
## Number of Fisher Scoring iterations: 25
```