Práctica 2: Limpieza y validación de los datos

Alberto Bayón Valtierra / Elena Ruiz Martinez

11 de junio 2019

Índice

---

1. Descripción del dataset

El conjunto de datos objeto de análisis se ha obtenido a partir de el siguiente enlace en Kaggle: https://www.kaggle.com/uciml/student-alcohol-consumption. Partimos del archivo .csv: **"student-por.csv"** que contienen los datos obtenidos de una encuesta hecha a estudiantes de lengua portuguesa de dos escuela de secundaria: Gabriel Pereira y Mousinho da Silveira. Estos datos contienen mucha información social, de género y de estudio sobre los estudiantes. Nos podríamos preguntar como de influyentes son los diferentes factores sociales sobre la calificación de los estudiantes y si podríamos predecir la calificación final del alumno a partir de esta información.

El dataset está formado por 33 atributos (columnas) y 649 alumnos(filas o registros). Entre los atributos de este conjunto de datos, encontramos los siguientes:

- **school**: escuela de secundaria (binario: 'GP' - Gabriel Pereira o 'MS' - Mousinho da Silveira)

- **sex**: sexo del estudiante (binario: 'F' - femenino o 'M' - masculino)

- **age**: edad del estudiante (numérico: de 15 a 22)

- **address**: tipo de domicilio del estudiante (binario: 'U' - urbano o 'R' - rural)

- **famsize**: tamaño de la familia (binario: 'LE3' - menor o igual a 3 o 'GT3' - mayor que 3)

- **Pstatus**: estado de convivencia de los padres (binario: 'T' - viviendo juntos o 'A' - separados)

- **Medu**: educación de la madre (numérico: 0 - ninguna, 1 - educación primaria (4º grado), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

- **Fedu**: educación del padre (numérico: 0 - ninguna, 1 - educación primaria (4º grado), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

- **Mjob**: trabajo de la madre (nominal: 'teacher', 'health' relacionado con el cuidado, civil 'services' (por ejemplo, administrativo o policial), 'at_home' o 'other')

- **Fjob**: trabajo del padre (nominal: 'teacher', 'health' relacionado con el cuidado, civil 'services' (por ejemplo, administrativo o policial), 'at_home' o 'other')

- **reason**: razón para elegir esta escuela (nominal: 'home' cerca de casa, 'reputation' reputacion de la escuela, 'course' preferencia de curso o 'other') - **guardian**: tutor del estudiante (nominal: 'mother', 'father' or 'other')

- **traveltime**: tiempo de viaje de la casa a la escuela (numérico: 1 - <15 min., 2 - 15 a 30 min., 3 - 30 min. a 1 hora, o 4 - >1 hora)

- **studytime**: tiempo de estudio semanal (numérico: 1 - <2 horas, 2 - 2 a 5 horas, 3 - 5 a 10 horas, o 4 - >10 horas)

- **failures**: número de faltas a clase (numérico: n si 1<=n<3, sino 4)

- **schoolsup**: apoyo educativo extra (binario: yes o no)

- **famsup**: apoyo educativo familiar (binario: yes o no)

- **paid**: clases extra pagadas dentro de la materia del curso (matemáticas o portugués) (binario: yes o no)

- **activities**: actividades extracurriculares (binario: yes o no)
- **nursery**: si fué a la guardería (binario: yes o no)
- **higher**: quiere hacer educación superior (binario: yes o no)
- **internet**: acceso a Internet en casa (binario: yes o no)
- **romantic**: con una relación romántica (binario: yes o no)
- **famrel**: calidad de las relaciones familiares (numérico: desde 1 - muy mala a 5 - excelente)
- **freetime**: tiempo libre después de la escuela (numérico: desde 1 - muy poco tiempo a 5 - mucho tiempo)
- **goout**: salir con amigos (numérico: desde 1 - muy bajo a 5 - muy alto)
- **Dalc**: consumo de alcohol durante la jornada laboral (numérico: de 1 - muy bajo a 5 - muy alto)
- **Walc**: consumo de alcohol durante el fin de semana (numérico: de 1 - muy bajo a 5 - muy alto)
- **health**: estado de salud actual (numérico: de 1 - muy malo a 5 - muy bueno)
- **absences**: Número de ausencias escolares (numérico: de 0 a 93)
- **G1** - calificación primer grado (numérico: de 0 a 20)
- **G2** - calificación segundo grado (numérico: de 0 a 20)
- **G3** - calificación final (numérico: de 0 a 20, target)

Cargamos el datasets

```
# Cargamos los datos de los estudiantes de portugués
alumnos=read.csv("student-por.csv")
```

Con el siguiente comando se observa el tamaño del dataset: 649 alumnos que participan en la encuesta y 33 atributos que sirven para caracterizar a los alumnos.

```
# Dimensiones del dataset
dim(alumnos)
```

```
## [1] 649  33
```

Como parte final de este apartado se inluye un resumen por columnas con el valor mínimo, la media, la mediana, el valor máximo, el primer y el tercer cuartiles para los datos numéricos. Y en el caso de los datos de tipos cualitativos indica la cardinalidad de cada uno de los valores.

```
options(knitr.kable.NA = '')
kable(summary(alumnos), caption='Resumen del dataset "alumnos')
```

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	
GP:423	F:383	Min. :15.00	R:197	GT3:457	A: 80	Min. :0.000	Min. :0.000	at_home :135	at_
MS:226	M:266	1st Qu.:16.00	U:452	LE3:192	T:569	1st Qu.:2.000	1st Qu.:1.000	health : 48	he
		Median :17.00				Median :2.000	Median :2.000	other :258	ot
		Mean :16.74				Mean :2.515	Mean :2.307	services:136	ser
		3rd Qu.:18.00				3rd Qu.:4.000	3rd Qu.:3.000	teacher : 72	tea
		Max. :22.00				Max. :4.000	Max. :4.000		

---

2. Integración y selección de los datos

Comprobamos qué tipo de datos contiene cada atributo.

```
# Tipo de dato asignado a cada campo
res <- sapply(alumnos,class)
kable(data.frame(atributo=names(res),clase=as.vector(res)))
```

atributo	clase
school	factor
sex	factor
age	integer
address	factor
famsize	factor
Pstatus	factor
Medu	integer
Fedu	integer
Mjob	factor
Fjob	factor
reason	factor
guardian	factor
traveltime	integer
studytime	integer
failures	integer
schoolsup	factor
famsup	factor
paid	factor
activities	factor
nursery	factor
higher	factor
internet	factor
romantic	factor

atributo	clase
famrel	integer
freetime	integer
goout	integer
Dalc	integer
Walc	integer
health	integer
absences	integer
G1	integer
G2	integer
G3	integer

Excepto: age, absences, G1, G2 y G3 (que son variables cuantitativas discretas), todos las demás variables deberían de ser de tipo "factor" (cualitativas), así que transformamos todas aquellas con la clase incorrecta a tipo "factor":

```
#alumnos$subject <- as.factor(alumnos$subject)
variables_erroneas<-c("Medu", "Fedu", "traveltime", "studytime", "failures", "famrel", "freetime", "goou
alumnos[variables_erroneas] <- lapply(alumnos[variables_erroneas], function(x) as.factor(x))

res <- sapply(alumnos,class)
kable(data.frame(atributo=names(res),clase=as.vector(res)))
```

atributo	clase
school	factor
sex	factor
age	integer
address	factor
famsize	factor
Pstatus	factor
Medu	factor
Fedu	factor
Mjob	factor
Fjob	factor
reason	factor
guardian	factor
traveltime	factor
studytime	factor
failures	factor
schoolsup	factor
famsup	factor
paid	factor
activities	factor
nursery	factor
higher	factor
internet	factor
romantic	factor
famrel	factor
freetime	factor
goout	factor
Dalc	factor

atributo	clase
Walc	factor
health	factor
absences	integer
G1	integer
G2	integer
G3	integer

---

3. Limpieza de los datos

3.1. Valores nulos

Comprobamos si tenemos valores nulos para cada uno de los atríbutos

```
# Números de valores desconocidos por campo
sapply(alumnos, function(x) sum(is.na(x)))
```

```
##     school        sex        age    address    famsize    Pstatus
##          0          0          0          0          0          0
##       Medu       Fedu       Mjob       Fjob     reason   guardian
##          0          0          0          0          0          0
## traveltime  studytime   failures  schoolsup     famsup       paid
##          0          0          0          0          0          0
## activities    nursery     higher   internet   romantic     famrel
##          0          0          0          0          0          0
##   freetime      goout       Dalc       Walc     health   absences
##          0          0          0          0          0          0
##         G1         G2         G3
##          0          0          0
```

Como podemos observar ninguna de las variables contiene valores nulos.

3.2. Valores ceros

Hacemos un análisis de los ceros que aparecen en cada una de las columnas

```
# Números de valores desconocidos por campo
kable(colSums(alumnos==0))
```

	x
school	0
sex	0
age	0

	x
address	0
famsize	0
Pstatus	0
Medu	6
Fedu	7
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	549
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	244
G1	1
G2	7
G3	15

De acuerdo con los resultados obtenidos, las columnas con pocos valores son susceptibles de ser valores perdidos. Vamos a analizar cada uno de ellos.

Los campos Medu y Fedu hacen referencia a la educación del padre y de la madre. El valor 0 tiene el significado de no poseer ningún nivel de educación. De 649 muestras obtener 6 y 7 casos de madres y padres sin educacón es un dato razonable que no tiene porqué ser una valor perdido.

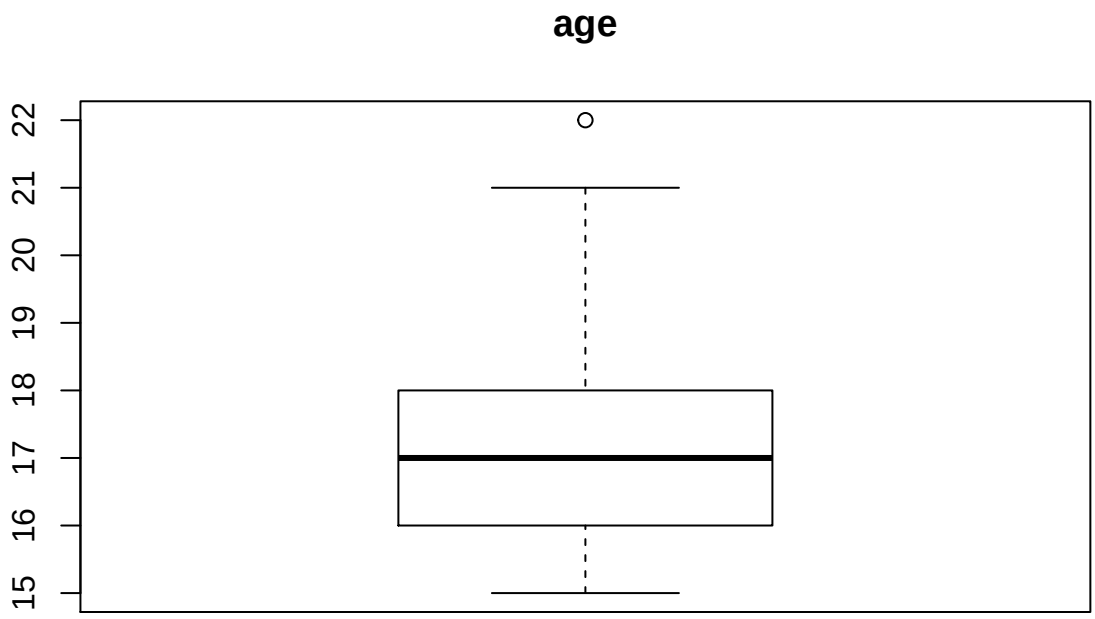
En el caso de los campos G1, G2 y G3 ser corresponde a valores numéricos (de 0 a 20) de 3 notas diferentes obtenidas por un alumno. Los datos de las tres notas con valor cero (1, 7 y 15 respectivamente) forman parte de la normalidad de los resultados por lo que no se pueden considerar tampoco como valores perdidos.

3.3. Valores extremos

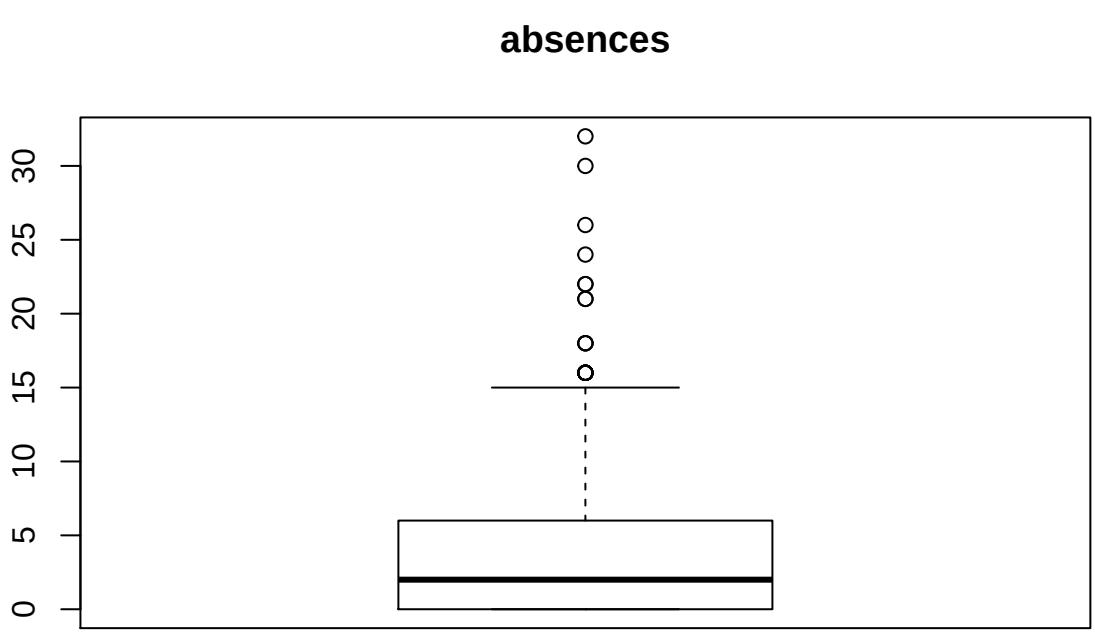
Veamos si existen valores extremos entre las variables cuantitativas.

```
for (i in 1:ncol(alumnos)) {
  if (is.integer(alumnos[,i])) {
    extreme<-boxplot(alumnos[names(alumnos[i])],main=names(alumnos[i]))
    cat(names(alumnos[i]))
    cat(": ")
    cat(extreme$out)
```

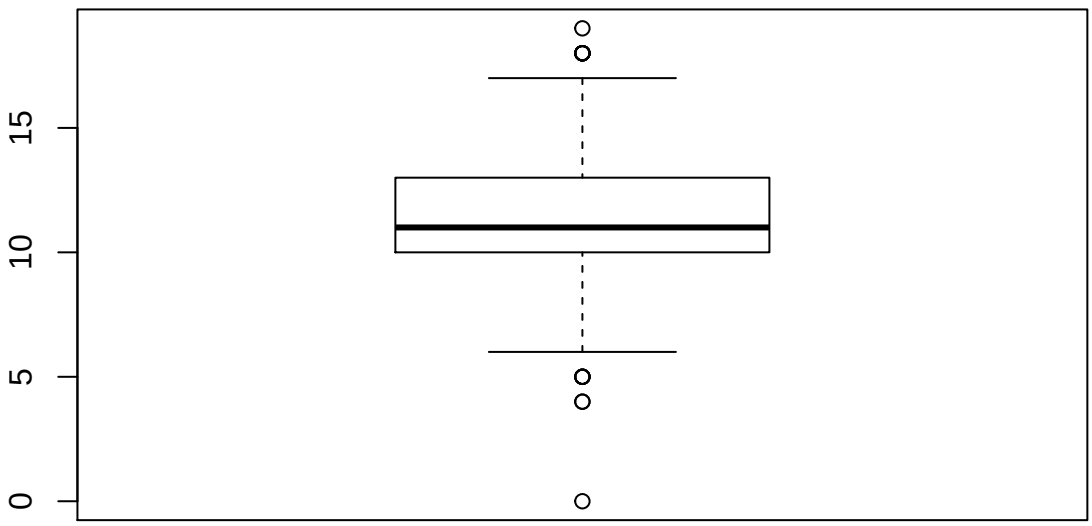
```
    cat("\n")
  }
}
```



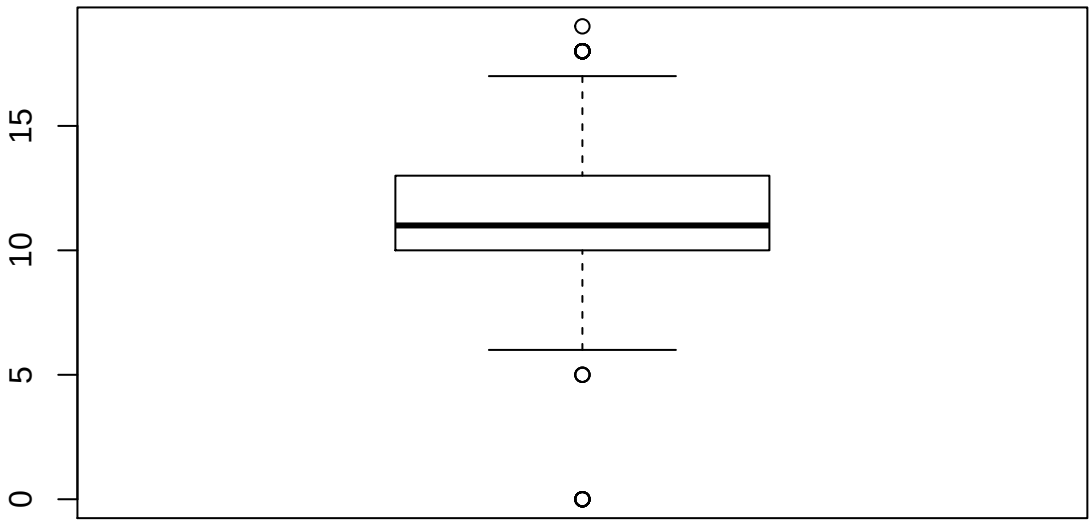

```
## age: 22
```



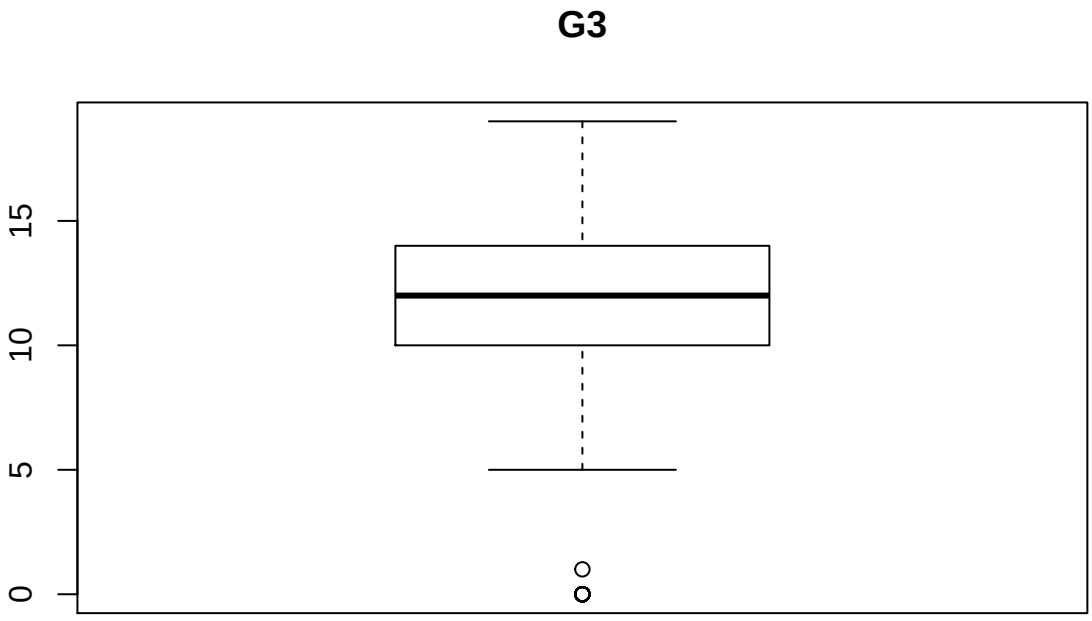

```
## absences: 16 16 24 22 16 32 16 16 30 21 16 18 16 26 16 16 22 18 18 16 21
```




```
## G1: 0 18 18 18 18 18 5 4 4 5 18 5 5 18 19 5
```




```
## G2: 18 18 18 18 19 18 18 18 18 18 0 5 18 0 0 5 18 18 0 0 0 18 0 5 18
```




```
## G3: 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Tanto los datos correspondientes a las edades, a las notas, como a las ausencias a clase, comprobamos que son valores que perfectamente pueden darse. Es por ello que el manejo de estos valores extremos consistirá en simplemente dejarlos como actualmente están recogidos.

---

4. Análisis de los datos

4.1. Análisis estadístico descriptivo

A continuación veamos un breve estudio estadístico descriptivo de los datos con los que vamos a trabajar.

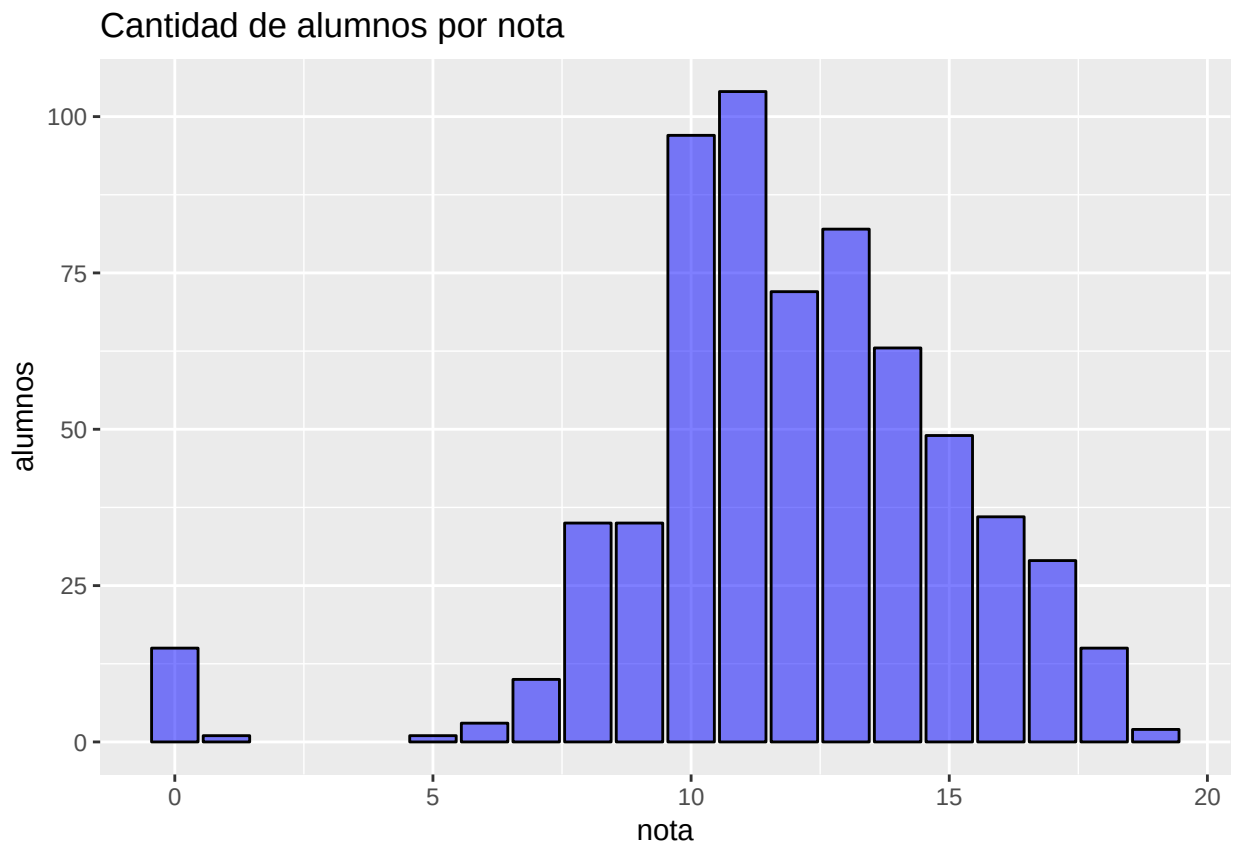
```
options(knitr.kable.NA = '')
kable(summary(alumnos))
```

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	
GP:423	F:383	Min. :15.00	R:197	GT3:457	A: 80	0: 6	0: 7	at_home :135	at_home : 42	co

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	
MS:226	M:266	1st Qu.:16.00	U:452	LE3:192	T:569	1:143	1:174	health : 48	health : 23	ho
		Median :17.00				2:186	2:209	other :258	other :367	ot
		Mean :16.74				3:139	3:131	services:136	services:181	repu
		3rd Qu.:18.00				4:175	4:128	teacher : 72	teacher : 36	
		Max. :22.00								

Vamos a representar mediante histogramas la nota final 'G3.'

```
ggplot(alumnos, aes(alumnos$G3)) + geom_bar(colour="black", fill="blue", alpha=.5, stat="count") + guid
```



4.2. Análisis estadístico inferencial

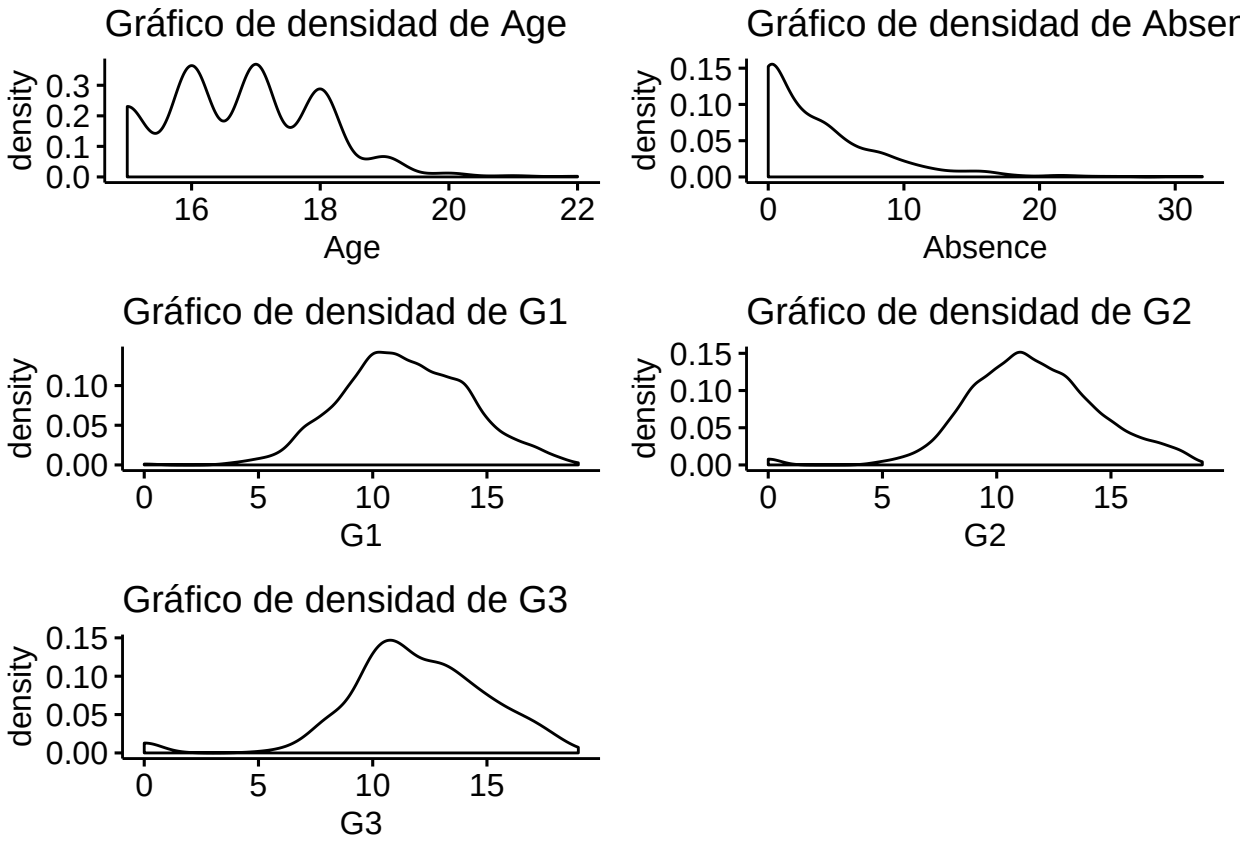
4.2.1. Normalidad

La normalidad se puede comprobar de un modo visual mediante gráficos de densidad o gráficos Q-Q

Utilizamos en primer lugar los gráficos de densidad

```
densityAge <- ggdensity(alumnos$age,
          main = "Gráfico de densidad de Age",
          xlab = "Age")
```

```
densityAbsence <- ggdensity(alumnos$absence,
           main = "Gráfico de densidad de Absence",
           xlab = "Absence")

densityG1 <- ggdensity(alumnos$G1,
           main = "Gráfico de densidad de G1",
           xlab = "G1")

densityG2 <- ggdensity(alumnos$G2,
           main = "Gráfico de densidad de G2",
           xlab = "G2")

densityG3 <- ggdensity(alumnos$G3,
           main = "Gráfico de densidad de G3",
           xlab = "G3")

grid.arrange(densityAge,densityAbsence,densityG1,densityG2, densityG3, ncol=2)
```

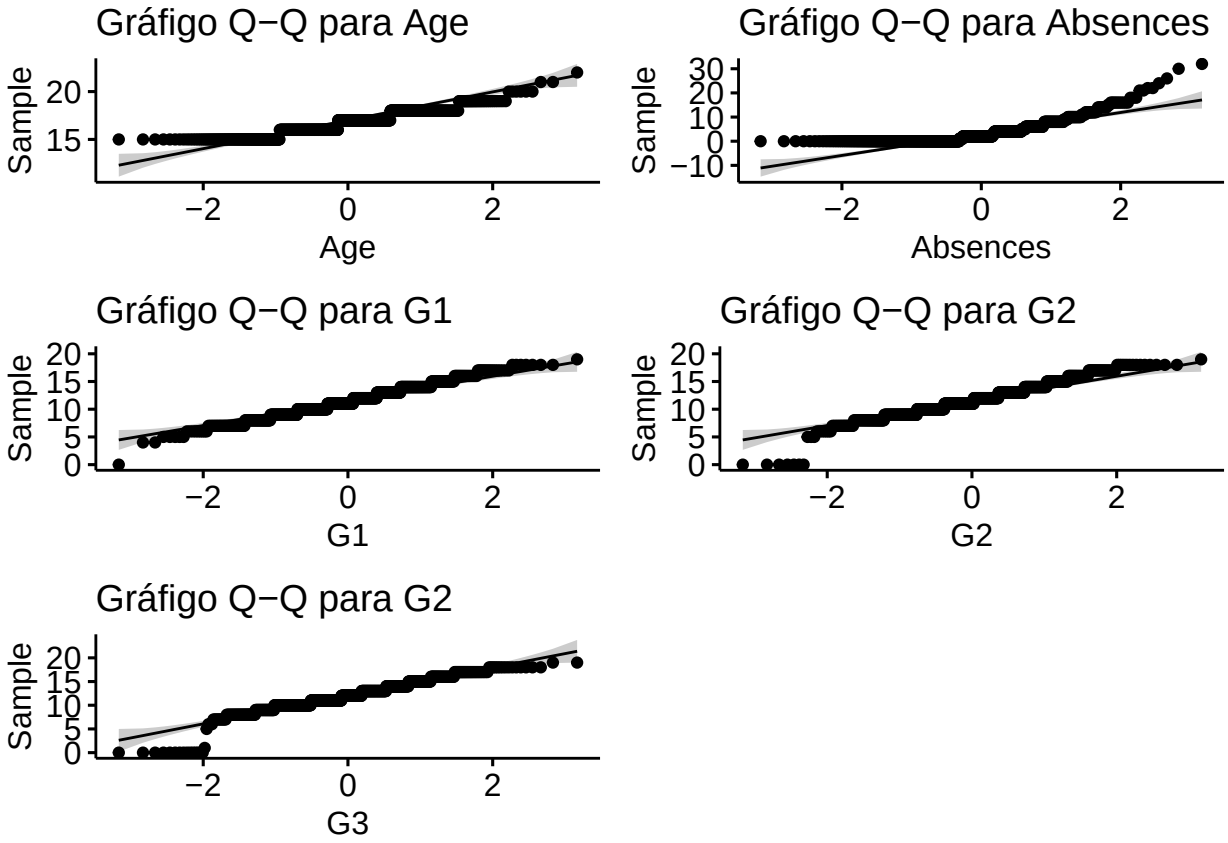


Viendo las gráficas de densidad parece que siguen una distribución normal las variables G1, G2 y G3

Y a continuación los gráfcos Q-Q

```
qqAge <- ggqqplot(alumnos$age, main="Gráfigo Q-Q para Age", xlab = "Age")

qqAbsences <- ggqqplot(alumnos$absences, main="Gráfigo Q-Q para Absences", xlab = "Absences")
```

```
qqG1 <- ggqqplot(alumnos$G1, main="Gráfigo Q-Q para G1", xlab = "G1")

qqG2 <- ggqqplot(alumnos$G2, main="Gráfigo Q-Q para G2", xlab = "G2")

qqG3 <- ggqqplot(alumnos$G3, main="Gráfigo Q-Q para G2", xlab = "G3")

grid.arrange(qqAge,qqAbsences,qqG1,qqG2, qqG3, ncol=2)
```



En este caso visualmente es más confuso determinar las variables que siguen la distriución normal, aunque da la impresión nuevamente que son las variables de las notas.

Sin embargo una de las formas más fiables de comprobar la normalidad es aplicar el test de Shapiro - Wilk para cada una de las variables.

```
shapiro.test(alumnos$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alumnos$age
## W = 0.91559, p-value < 2.2e-16
```

```
shapiro.test(alumnos$absences)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  alumnos$absences
## W = 0.77174, p-value < 2.2e-16
```

```
shapiro.test(alumnos$G1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alumnos$G1
## W = 0.98554, p-value = 4.934e-06
```

```
shapiro.test(alumnos$G2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alumnos$G2
## W = 0.96167, p-value = 5.583e-12
```

```
shapiro.test(alumnos$G3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alumnos$G3
## W = 0.92598, p-value < 2.2e-16
```

El test dice que si p-value es menor que 0,05 entonces no se considera que la variable siga una distribución normal. El resultado de todas ellas ha sido demasiado pequeño, por lo que se descarta su normalidad.

4.2.2. Homogeneidad

Se va a estudiar la homogeneidad de varianzas mediante el test de Fligner-Kileen. Como nuestro estudio se basa en la influencia de las diferentes variables sobre las notas finales de cada alumno (sobre G3), nos centraremos en analizar esta variable númerica sobre cada uno de los atributos categóricos. En los siguientes tests, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(G3 ~ school, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by school
## Fligner-Killeen:med chi-squared = 6.3761, df = 1, p-value =
## 0.01157
```

```r
fligner.test(G3 ~ sex, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by sex
## Fligner-Killeen:med chi-squared = 0.29686, df = 1, p-value =
## 0.5859
```

```r
fligner.test(G3 ~ address, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by address
## Fligner-Killeen:med chi-squared = 0.00054889, df = 1, p-value =
## 0.9813
```

```r
fligner.test(G3 ~ famsize, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by famsize
## Fligner-Killeen:med chi-squared = 0.60366, df = 1, p-value =
## 0.4372
```

```r
fligner.test(G3 ~ Pstatus, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Pstatus
## Fligner-Killeen:med chi-squared = 0.0037493, df = 1, p-value =
## 0.9512
```

```r
fligner.test(G3 ~ Medu, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Medu
## Fligner-Killeen:med chi-squared = 6.4495, df = 4, p-value = 0.168
```

```r
fligner.test(G3 ~ Fedu, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Fedu
## Fligner-Killeen:med chi-squared = 6.054, df = 4, p-value = 0.1951
```

```r
fligner.test(G3 ~ Mjob, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Mjob
## Fligner-Killeen:med chi-squared = 4.5431, df = 4, p-value = 0.3375
```

```r
fligner.test(G3 ~ Fjob, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Fjob
## Fligner-Killeen:med chi-squared = 2.6856, df = 4, p-value = 0.6117
```

```r
fligner.test(G3 ~ reason, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by reason
## Fligner-Killeen:med chi-squared = 2.4788, df = 3, p-value = 0.4791
```

```r
fligner.test(G3 ~ guardian, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by guardian
## Fligner-Killeen:med chi-squared = 6.8708, df = 2, p-value =
## 0.03221
```

```r
fligner.test(G3 ~ traveltime, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by traveltime
## Fligner-Killeen:med chi-squared = 3.9833, df = 3, p-value = 0.2633
```

```r
fligner.test(G3 ~ studytime, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by studytime
## Fligner-Killeen:med chi-squared = 3.9943, df = 3, p-value = 0.2621
```

```r
fligner.test(G3 ~ failures, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by failures
## Fligner-Killeen:med chi-squared = 3.6314, df = 3, p-value = 0.3041
```

```r
fligner.test(G3 ~ schoolsup, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by schoolsup
## Fligner-Killeen:med chi-squared = 14.913, df = 1, p-value =
## 0.0001126
```

```r
fligner.test(G3 ~ famsup, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by famsup
## Fligner-Killeen:med chi-squared = 1.5335, df = 1, p-value = 0.2156
```

```r
fligner.test(G3 ~ paid, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by paid
## Fligner-Killeen:med chi-squared = 2.4767, df = 1, p-value = 0.1155
```

```r
fligner.test(G3 ~ activities, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by activities
## Fligner-Killeen:med chi-squared = 0.013073, df = 1, p-value =
## 0.909
```

```r
fligner.test(G3 ~ nursery, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by nursery
## Fligner-Killeen:med chi-squared = 2.1469, df = 1, p-value = 0.1429
```

```r
fligner.test(G3 ~ higher, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by higher
## Fligner-Killeen:med chi-squared = 5.6016, df = 1, p-value =
## 0.01794
```

```r
fligner.test(G3 ~ internet, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by internet
## Fligner-Killeen:med chi-squared = 1.2913, df = 1, p-value = 0.2558
```

```r
fligner.test(G3 ~ romantic, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by romantic
## Fligner-Killeen:med chi-squared = 2.7856, df = 1, p-value =
## 0.09512
```

```r
fligner.test(G3 ~ famrel, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by famrel
## Fligner-Killeen:med chi-squared = 4.4347, df = 4, p-value = 0.3504
```

```r
fligner.test(G3 ~ freetime, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by freetime
## Fligner-Killeen:med chi-squared = 4.5844, df = 4, p-value = 0.3327
```

```r
fligner.test(G3 ~ goout, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by goout
## Fligner-Killeen:med chi-squared = 4.0307, df = 4, p-value = 0.4019
```

```
fligner.test(G3 ~ Dalc, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Dalc
## Fligner-Killeen:med chi-squared = 6.8176, df = 4, p-value = 0.1458
```

```
fligner.test(G3 ~ Walc, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by Walc
## Fligner-Killeen:med chi-squared = 5.8521, df = 4, p-value = 0.2105
```

```
fligner.test(G3 ~ health, data = alumnos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  G3 by health
## Fligner-Killeen:med chi-squared = 3.7998, df = 4, p-value = 0.4338
```

El valor obtenido de p-value sobre las variables `school, guardian, schoolsup y higher` son inferiores a 0,05 por lo que se puede considerar la hipótesis nula como no válida y por lo tanto que las varianzas de estas muestras sobre G3 no son homogéneas. Para el resto de variables categóricas se muestra un p-value superior a 0.05, asi que aceptamos las hipotesis nula de que las varianzas son iguales.

4.2.3. Correlación

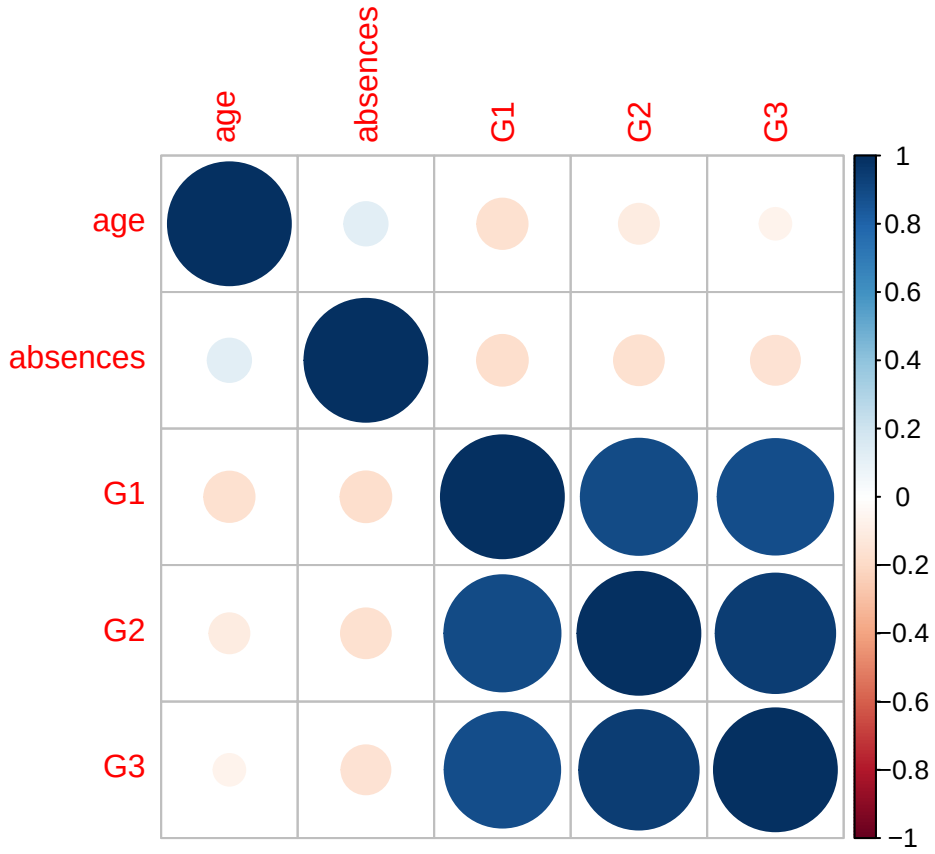
Comprobamos la correlación entre las variables numéricas. Puesto que no siguen una distribución normal en lugar de aplicar el metodo Pearson que es el que se usa por defecto con la función cor de R, vamos a aplicar el método de Spearman.

```
#
columnas_numericas<- sapply(alumnos, is.numeric)
correlacion<-cor(alumnos[,columnas_numericas], method = "spearman")
kable(correlacion)
```

	age	absences	G1	G2	G3
age	1.0000000	0.1242605	-0.1673732	-0.1055950	-0.0662769
absences	0.1242605	1.0000000	-0.1704299	-0.1638854	-0.1585100
G1	-0.1673732	-0.1704299	1.0000000	0.8930649	0.8832876
G2	-0.1055950	-0.1638854	0.8930649	1.0000000	0.9444512
G3	-0.0662769	-0.1585100	0.8832876	0.9444512	1.0000000

Gráficamente se puede ver en el siguiente diagrama

```
corrplot(correlacion)
```



Vemos que las variables G2 y G3 tienen una alta correlación: 0.94, seguido de G1 y G2 con una correlación de 0.89 y G1 con G3 de 0,88. Tiene toda su lógica, ya que G3 corresponde a la nota final del curso. Como las tres variables tienen un agran correlación, eliminaremos los campos correspondientes a G1 y G2 y nos quedaremos unicamente con G3, ya que para el estudio que queremos hacer los otros datos nos son irrelevantes.

```
alumnos<- select(alumnos, -G1, -G2)
```

4.3. Pruebas estadísticas

4.3.1. Pruebas por contraste de hipótesis

Puesto que G3 no seguía una distribución normal, usaremos pruebas no paramétricas para hacer diferentes contrastes de hipótesis sobre G3.

Para las variables que tienen dos clases: `school`, `sex`, `address`, `famsize`, `Pstatus`, `schoolsup`, `famsup`, `paid`, `activities`, `nursery`, `higher`, `internet` y `romantic`, utilizaremos comparaciones entre dos grupos de datos aplicando las pruebas de Wilcoxon. La hipótesis nula asume que las distribuciones de los grupos de datos son las mismas, por lo tanto para p-value inferior a 0.05 se rechazará la hipótesis nula y se concluirá que existen diferencias estadísticamente significativas entre los grupos de datos analizados.

```
wilcox.test(G3 ~ school, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by school
## W = 64220, p-value = 3.792e-13
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ sex, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by sex
## W = 58916, p-value = 0.0006317
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ address, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by address
## W = 35097, p-value = 1.567e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ famsize, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by famsize
## W = 42818, p-value = 0.6267
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ Pstatus, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by Pstatus
## W = 22964, p-value = 0.8965
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ schoolsup, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by schoolsup
## W = 22988, p-value = 0.02611
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ famsup, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by famsup
## W = 48222, p-value = 0.4549
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ paid, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by paid
## W = 13522, p-value = 0.1494
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ activities, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by activities
## W = 47656, p-value = 0.03693
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ nursery, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by nursery
## W = 30822, p-value = 0.1819
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ higher, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by higher
## W = 6856.5, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(G3 ~ internet, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  G3 by internet
## W = 29582, p-value = 6.39e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(G3 ~ romantic, data = alumnos)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  G3 by romantic
## W = 53186, p-value = 0.06719
## alternative hypothesis: true location shift is not equal to 0
```

Según los resultados podemos decir que hay diferencias significativas en las notas entre : - los alumnos de cada una de las escuelas (school) - los alumnos de sexo femenino y masculino (sex) - los alumnos que viven en una zona rural o urbana (address) - los alumnos que tienen un apoyo eductaivo extra y los que no lo tienen (schoolsup) - los alumnos que hacen actividades extracurriculares y los que no las hacen (activities) - los alumnos que quieren hacer educación superior y los que no (higher) - los alumnos que tienen internet en casa y lo que no tienen internet (internet)

Para las variables que tienen más de dos clases: `Medu, Fedu, Mjob, Fjob, reason, guardian, traveltime, studytime, failures, famrel, freetime, goout, Dalc, Walc y health`, utilizaremos comparaciones entre más de dos grupos de datos aplicando el test de Kruskal-Wallis.

```
kruskal.test(G3 ~ Medu, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Medu
## Kruskal-Wallis chi-squared = 57.215, df = 4, p-value = 1.115e-11
```

```
kruskal.test(G3 ~ Fedu, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Fedu
## Kruskal-Wallis chi-squared = 36.785, df = 4, p-value = 1.994e-07
```

```
kruskal.test(G3 ~ Mjob, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Mjob
## Kruskal-Wallis chi-squared = 37.136, df = 4, p-value = 1.689e-07
```

```r
kruskal.test(G3 ~ Fjob, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Fjob
## Kruskal-Wallis chi-squared = 17.265, df = 4, p-value = 0.001717
```

```r
kruskal.test(G3 ~ reason, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by reason
## Kruskal-Wallis chi-squared = 31.625, df = 3, p-value = 6.279e-07
```

```r
kruskal.test(G3 ~ guardian, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by guardian
## Kruskal-Wallis chi-squared = 8.0177, df = 2, p-value = 0.01815
```

```r
kruskal.test(G3 ~ traveltime, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by traveltime
## Kruskal-Wallis chi-squared = 14.751, df = 3, p-value = 0.002043
```

```r
kruskal.test(G3 ~ studytime, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by studytime
## Kruskal-Wallis chi-squared = 50.316, df = 3, p-value = 6.842e-11
```

```r
kruskal.test(G3 ~ failures, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by failures
## Kruskal-Wallis chi-squared = 130.66, df = 3, p-value < 2.2e-16
```

```r
kruskal.test(G3 ~ famrel, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by famrel
## Kruskal-Wallis chi-squared = 12.434, df = 4, p-value = 0.0144
```

```r
kruskal.test(G3 ~ freetime, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by freetime
## Kruskal-Wallis chi-squared = 19.546, df = 4, p-value = 0.0006136
```

```r
kruskal.test(G3 ~ goout, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by goout
## Kruskal-Wallis chi-squared = 19.766, df = 4, p-value = 0.0005553
```

```r
kruskal.test(G3 ~ Dalc, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Dalc
## Kruskal-Wallis chi-squared = 29.665, df = 4, p-value = 5.726e-06
```

```r
kruskal.test(G3 ~ Walc, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by Walc
## Kruskal-Wallis chi-squared = 24.297, df = 4, p-value = 6.963e-05
```

```r
kruskal.test(G3 ~ health, data = alumnos)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  G3 by health
## Kruskal-Wallis chi-squared = 10.997, df = 4, p-value = 0.0266
```

Dado que el p-valor obtenido es menor al nivel de significancia , se puede concluir que las notas G3 muestran diferencias significativas para las diferentes clases de las variables categoricas analizadas. Es decir, las variables : `Medu, Fedu, Mjob, Fjob, reason, guardian, traveltime, studytime, failures, famrel, freetime, goout, Dalc, Walc y health` tienen un peso significativo en las notas finales.

Como tenemos muchas variables en nuestro conjunto de datos, nos quedaremos con aquellas más significativas, aquellas que hemos obtenido un p-value inferior y y eliminamos las menos significativas.

```
# Seleccionamos las variables que nos interesan.
alumnos1<- alumnos[,c("school", "sex", "address", "schoolsup", "higher", "internet", "Medu", "Fedu", "M
```

4.3.2. Modelo de regresión lineal múltiple (regresores cuantitativos y cualitativos)

Estimaremos por mínimos cuadrados ordinarios un modelo lineal que explique la nota final (G3) de un individuo en función de todas las variables.

Para la futura evaluación del modelo, querremos dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento es el subconjunto del conjunto original de datos utilizado para construir un primer modelo; y el conjunto de prueba, el subconjunto del conjunto original de datos utilizado para evaluar la calidad del modelo.

Lo más correcto será utilizar un conjunto de datos de entrenamiento diferente del de prueba. Se utilizarán 2/3 para el conjunto de entrenamiento y 1/3, para el conjunto de prueba.

```
# Se crean los conjuntos de pruebas y de entrenamiento con 2/3 de los elementos
set.seed(666)
indexes = sample(1:nrow(alumnos1), size=floor((2/3)*nrow(alumnos1)))
train<-alumnos1[indexes,]
test<-alumnos1[-indexes,]
```

```
# Creamos el modelo de regresion lineal con los datos de entrenamiento
modelo1<- lm(G3~., data=train )
summary(modelo1)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6836  -1.3639  -0.0703   1.6029   6.5877
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.969183   1.717909   8.132 5.50e-15 ***
## schoolMS         -1.198887   0.303379  -3.952 9.18e-05 ***
## sexM             -0.712856   0.285669  -2.495 0.012988 *
## addressU          0.415858   0.298477   1.393 0.164321
## schoolsupyes     -1.237229   0.425526  -2.908 0.003848 **
## higheryes         1.288900   0.461339   2.794 0.005462 **
## internetyes       0.378703   0.321849   1.177 0.240043
## Medu1             0.274951   1.373392   0.200 0.841428
## Medu2            -0.469380   1.377599  -0.341 0.733493
## Medu3            -0.017625   1.390139  -0.013 0.989891
```

```
## Medu4             0.219596   1.422952   0.154 0.877433
## Fedu1            -3.104005   1.362336  -2.278 0.023232 *
## Fedu2            -2.643828   1.367490  -1.933 0.053906 .
## Fedu3            -2.647379   1.392121  -1.902 0.057937 .
## Fedu4            -2.537612   1.410157  -1.800 0.072697 .
## Mjobhealth        0.837299   0.601090   1.393 0.164411
## Mjobother         0.032057   0.357098   0.090 0.928515
## Mjobservices      0.486639   0.434731   1.119 0.263647
## Mjobteacher       0.983441   0.579734   1.696 0.090602 .
## reasonhome       -0.077567   0.328177  -0.236 0.813278
## reasonother      -1.010207   0.414047  -2.440 0.015132 *
## reasonreputation  0.144226   0.348184   0.414 0.678934
## studytime2        0.355690   0.301612   1.179 0.238987
## studytime3        1.344625   0.413862   3.249 0.001257 **
## studytime4        0.923336   0.576182   1.603 0.109841
## failures1        -2.960309   0.472667  -6.263 9.85e-10 ***
## failures2        -2.580808   0.911822  -2.830 0.004886 **
## failures3        -2.525669   1.011473  -2.497 0.012929 *
## Dalc2            -0.360360   0.369915  -0.974 0.330567
## Dalc3             0.074647   0.596554   0.125 0.900483
## Dalc4            -3.034818   0.779365  -3.894 0.000116 ***
## Dalc5            -1.694476   1.063630  -1.593 0.111934
## Walc2             0.066016   0.349036   0.189 0.850081
## Walc3            -0.494605   0.388681  -1.273 0.203935
## Walc4            -0.672174   0.479694  -1.401 0.161921
## Walc5            -0.005384   0.715773  -0.008 0.994002
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.506 on 396 degrees of freedom
## Multiple R-squared:  0.4344, Adjusted R-squared:  0.3844
## F-statistic: 8.689 on 35 and 396 DF,  p-value: < 2.2e-16
```

```
#Se aplican a los datos de TEST para realizar predicción y medir la precisión del modelo
predict_log <- predict(modelo1,newdata=test,type="response")
predict_log <- round(predict_log)

# Veamos una tabla con las 20 primeras notas predichas por el modelo comparadas con la nota real
tabla_predicciones<-data.frame(nota_predicha=predict_log,nota_real=test$G3)
kable(tabla_predicciones[0:10,])
```

	nota_predicha	nota_real
3	10	12
6	14	13
8	12	13
10	13	13
12	14	13
13	13	12
14	14	13
17	16	14
18	12	14
19	10	7

El coeficiente de determinación del modelo es muy bajo, por lo tanto una predicción muy ineficiente (siendo R-squared una medida de calidad del modelo que toma valores entre 0 y 1).

Por otra parte, han sido significativos los test parciales sobre los coeficientes de los regresores: schoolMS, failures1, failures2, failures3, higheryes y goout2.

Aunque hayamos obtenido una predicción muy ineficiente, si nos fijamos en la tabla, parece que se ha acercado bastante al valor real de la nota, así que aunque la predicción del modelo no sea exacta, podemos decir que se aproxima bastante.

En vez de querer predecir la nota, probemos ahora en predecir si un alumno aprueba o suspende en función de todas las variables, para ello urilizaremos un modelo de regresión logística:

4.3.3. Modelo de regresión logística

Para evaluar esta probabilidad se aplicará un modelo de regresión logística, donde la variable depediente será una variable binaria que indicará si el alumno ha aprobado o no la asignatura.

El primer paso será crear una variable binaria (aprobado) que indique la condición de aprobado (aprobado = 1) o no aprobado (aprobado = 0). Estimar el modelo de regresión logística donde la variable dependiente es "aprobado" y las explicativas son todas las variables del dataset excepto las correspondientes a las notas.

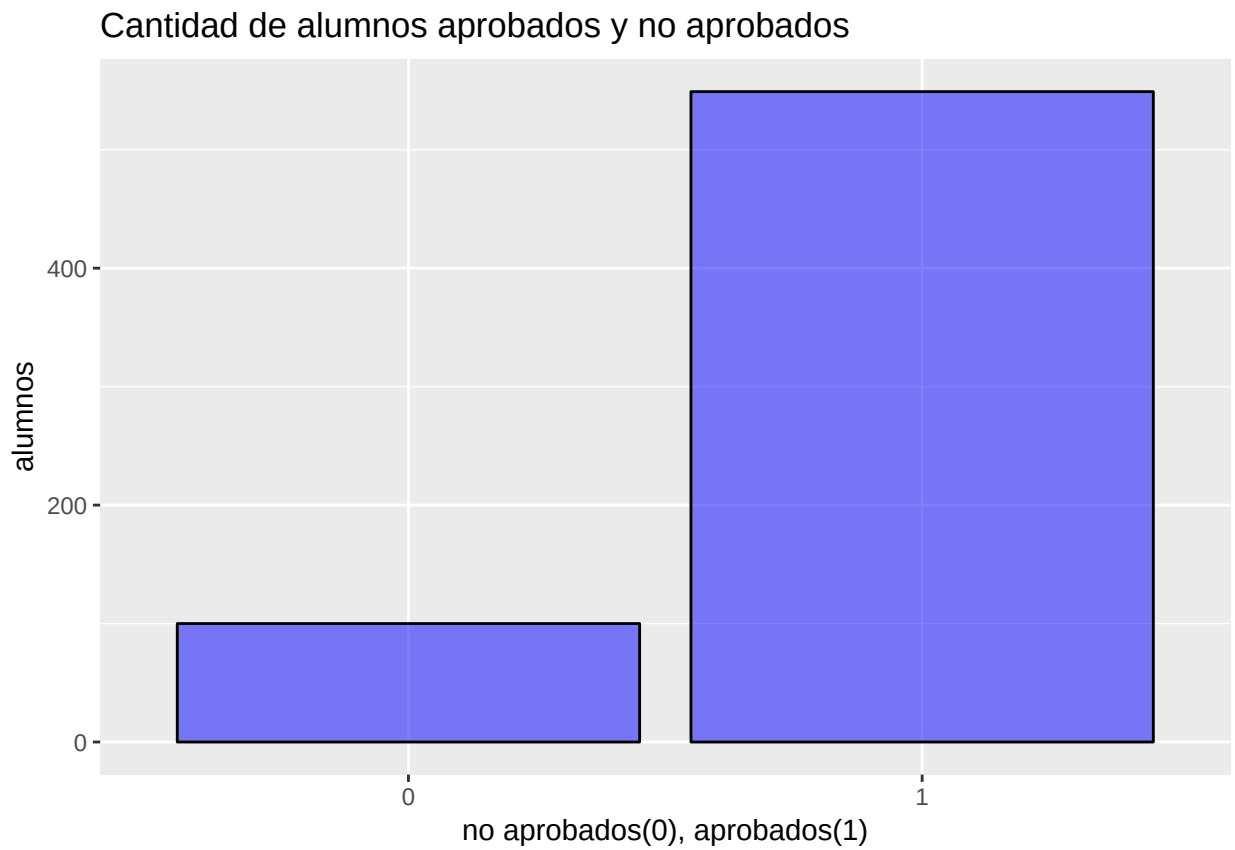
```
# Clasificación binaria del atributo G3 en aprobados o no aprobados
alumnos1$aprobado <- as.factor(ifelse(alumnos1$G3>9,1,0))

# Mostramos la cantidad de alumnos aprobados y suspendidos
table(alumnos1$aprobado)
```

```
##
##   0   1
## 100 549
```

```
# Eliminamos del dataset la variable correspondiente a las notas finales G3.
alumnos2<- select(alumnos1, -G3)
```

Vamos a representar mediante histogramas la cantidad de alumnos aprobados y no aprobados

```
ggplot(alumnos2, aes(alumnos2$aprobado)) + geom_bar(colour="black", fill="blue", alpha=.5, stat="count"
```



Dividimos los datos en un conjunto de entrenamiento y conjunto de prueba.

```
set.seed(666)

# Mediante "stratified" nos aseguramos tener la misma proporción en las clases del conjunto de entrenam
h2<-holdout(alumnos2$aprobado,ratio=2/3,mode="stratified")
data_train<-alumnos2[h2$tr,]
data_test<-alumnos2[h2$ts,]

# Visualizamos las proporciones de cada conjunto de datos
print((prop.table(table(data_train$aprobado))*100)%>% round(digits = 2))
```

```
##
##     0     1
## 15.47 84.53
```

```
print((prop.table(table(data_test$aprobado))*100)%>% round(digits = 2))
```

```
##
##     0     1
## 15.28 84.72
```

Observamos que la proporción de aprobados y no aprobados para el conjunto de entrenamiento y el de prueba es practicamente igual.

```r
# Estimamos el modelo
modelo2 =glm(aprobado~., family=binomial, data=data_train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(modelo2)
```

```
##
## Call:
## glm(formula = aprobado ~ ., family = binomial, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2353   0.1043   0.2142   0.4339   2.2602
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        35.15673 1911.08753   0.018  0.98532
## schoolMS           -2.00553    0.44495  -4.507 6.56e-06 ***
## sexM               -0.27753    0.42534  -0.652  0.51409
## addressU            0.17778    0.39232   0.453  0.65045
## schoolsupyes       -1.32626    0.63804  -2.079  0.03765 *
## higheryes           1.68261    0.52222   3.222  0.00127 **
## internetyes        -0.14412    0.43591  -0.331  0.74093
## Medu1             -15.77797 1358.81680  -0.012  0.99074
## Medu2             -15.68132 1358.81686  -0.012  0.99079
## Medu3             -16.64837 1358.81688  -0.012  0.99022
## Medu4             -15.60472 1358.81701  -0.011  0.99084
## Fedu1             -17.69810 1343.82768  -0.013  0.98949
## Fedu2             -16.85519 1343.82771  -0.013  0.98999
## Fedu3             -15.95103 1343.82784  -0.012  0.99053
## Fedu4             -18.07169 1343.82781  -0.013  0.98927
## Mjobhealth         -0.36384    0.89583  -0.406  0.68463
## Mjobother          -0.23527    0.49517  -0.475  0.63470
## Mjobservices        0.04766    0.65017   0.073  0.94156
## Mjobteacher        -0.09909    0.94849  -0.104  0.91680
## reasonhome         -0.03297    0.47379  -0.070  0.94452
## reasonother         0.27419    0.57004   0.481  0.63052
## reasonreputation    0.37245    0.55148   0.675  0.49944
## studytime2          0.71648    0.42174   1.699  0.08935 .
## studytime3          0.33548    0.58915   0.569  0.56906
## studytime4          1.46516    1.39246   1.052  0.29270
## failures1          -2.00422    0.47232  -4.243 2.20e-05 ***
## failures2          -2.71037    0.90089  -3.009  0.00262 **
## failures3          -3.14441    1.25877  -2.498  0.01249 *
## Dalc2               0.17076    0.52544   0.325  0.74520
## Dalc3               1.83891    0.90574   2.030  0.04233 *
## Dalc4              -1.79998    0.97401  -1.848  0.06460 .
## Dalc5              -0.46592    1.27764  -0.365  0.71536
## Walc2              -0.32665    0.50348  -0.649  0.51648
## Walc3              -0.16923    0.57408  -0.295  0.76815
## Walc4              -0.80066    0.63774  -1.255  0.20931
```

```
## Walc5              -0.61718    1.08334  -0.570  0.56888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 373.10  on 432  degrees of freedom
## Residual deviance: 230.89  on 397  degrees of freedom
## AIC: 302.89
##
## Number of Fisher Scoring iterations: 16
```

```
#Se aplican a los datos de TEST para realizar predicción y medir la precisión del modelo
predict_log2 <- predict(modelo2, newdata=data_test,type="response")
predict_log2 <- round(predict_log2)
```

Asumiendo como nivel de significancia a 0.05, todas aquellas variable con un p-valor inferior serán las más significativas. A continuación mostramos cuáles han sido:

```
sel <- which(summary(modelo2)$coefficients[-1,4] < 0.05)
names(sel)
```

```
## [1] "schoolMS"     "schoolsupyes" "higheryes"    "failures1"
## [5] "failures2"    "failures3"    "Dalc3"
```

En este caso, la bondad del modelo se evaluará mediante la medida AIC. Dado que esta medida tiene en cuenta tanto la bondad del ajuste como la complejidad del modelo, cuando se comparen varios modelos candidatos, se seleccionará aquel que resulte en el menor AIC.

Compararemos el modelo con uno que utilize menos variables para comprobar si mejoramos el AIC.

```
# Estimamos el modelo
modelo3 <-glm(aprobado~school+schoolsup+higher+failures+Dalc, family=binomial, data=data_train)
summary(modelo3)
```

```
##
## Call:
## glm(formula = aprobado ~ school + schoolsup + higher + failures +
##     Dalc, family = binomial, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9847   0.2440   0.2440   0.5468   1.8442
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.9873     0.4965   4.003 6.26e-05 ***
## schoolMS      -1.6749     0.3533  -4.741 2.12e-06 ***
## schoolsupyes  -0.7002     0.5567  -1.258 0.208506
## higheryes      1.5125     0.4250   3.559 0.000372 ***
## failures1     -1.8113     0.4039  -4.485 7.30e-06 ***
## failures2     -2.3237     0.7574  -3.068 0.002155 **
## failures3     -3.1629     0.9856  -3.209 0.001332 **
```

```
## Dalc2         -0.3422     0.4227  -0.810 0.418213
## Dalc3          0.9426     0.7251   1.300 0.193590
## Dalc4         -2.1680     0.7804  -2.778 0.005467 **
## Dalc5         -1.1503     0.7981  -1.441 0.149476
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 373.10  on 432  degrees of freedom
## Residual deviance: 262.04  on 422  degrees of freedom
## AIC: 284.04
##
## Number of Fisher Scoring iterations: 5
```

Utilizando unicamente las variables `school, schoolsup, higher, failures y Dalc` vemos que el AIC es inferior, con este modelo conseguiremos mejor bondad.

Calidad del ajuste

Calcularemos la matriz de confusión del modelo que hemos obtenido con mejor AIC, suponiendo un umbral de discriminación del 70 % observaremos cuantos falsos negativos y positivos.

```
# Calculamos la probabilidad para cada muestra del conjunto de prueba
prob_aprobado<- predict(modelo3, type = 'response', newdata=data_test)

# Si la probabilidad de aprobar es superior al 70% le asignamos la clase 1, si no le asignamos clase 0.
pred_aprobado <- ifelse(prob_aprobado > 0.7, 1, 0)
pred_aprobado <- factor(pred_aprobado, levels = c("0", "1"))

# Calculamos la matriz de confusión
confusionMatrix(pred_aprobado, data_test$aprobado)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  15  23
##          1  18 160
##
##                Accuracy : 0.8102
##                  95% CI : (0.7514, 0.8602)
##     No Information Rate : 0.8472
##     P-Value [Acc > NIR] : 0.9427
##
##                   Kappa : 0.3096
##
##  Mcnemar's Test P-Value : 0.5322
##
##             Sensitivity : 0.45455
##             Specificity : 0.87432
##          Pos Pred Value : 0.39474
##          Neg Pred Value : 0.89888
##              Prevalence : 0.15278
```

```
##          Detection Rate : 0.06944
##    Detection Prevalence : 0.17593
##       Balanced Accuracy : 0.66443
##
##        'Positive' Class : 0
##
```

```r
# Mostramos la precisión del modelo
confusionMatrix(pred_aprobado, data_test$aprobado)$overall[1]
```

```
##  Accuracy
## 0.8101852
```

Hay 23 falsos negativos. Corresponden a alumnos que han aprobado pero el modelo ha predicho que su probabilidad de ser aprobado es inferior a 0.7 y por lo tanto lo clasifica como "no aprobado".

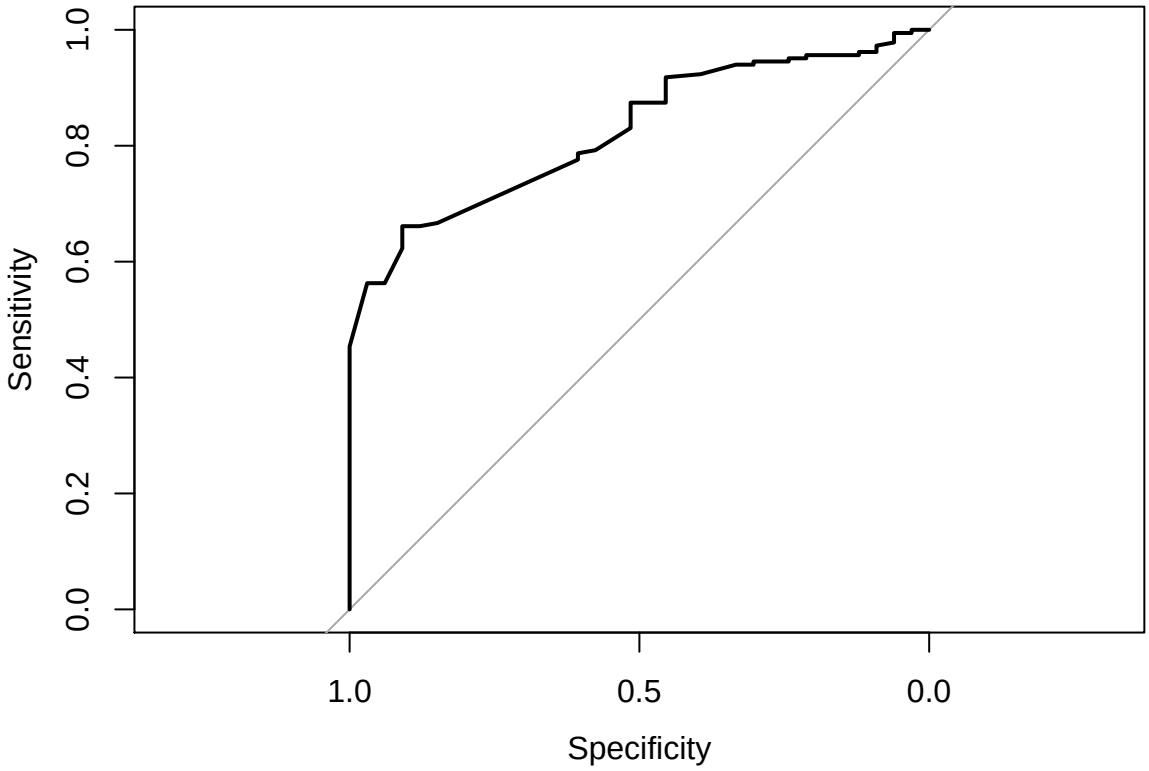
Hay 18 falsos positivos. Corresponden a alumnos "no aprobados", pero el modelo ha predicho que su probabilidad de ser aprobado es superior a 0.7 y por lo tanto los clasifica como "no aprobado".

La precisión del modelo es de un 81 %, no está nada mal, aunque seguramente estudiando otros modelos y utilizando otro tipo de entrenamiento podríamos conseguir mejores resultados.

Curva ROC

Realizaremos el dibujo de la curva ROC para representar la calidad del modelo predictivo obtenido. También calcularemos el AUROC, que nos proporciona información sobre la calidad del modelo, siendo menos preciso a medida que el AUC se acerca a 0.5 y mostrando una exactitud perfecta cuando es 1.

```r
g=roc(as.numeric(data_test$aprobado), prob_aprobado, data=data_test)
plot(g)
```




```
auc(g)
```

```
## Area under the curve: 0.8245
```

AUROC es 0.824.

El modelo logístico tiene un poder predictivo bastante bueno, ya que tiene un AUROC elevado, 0.824.

---

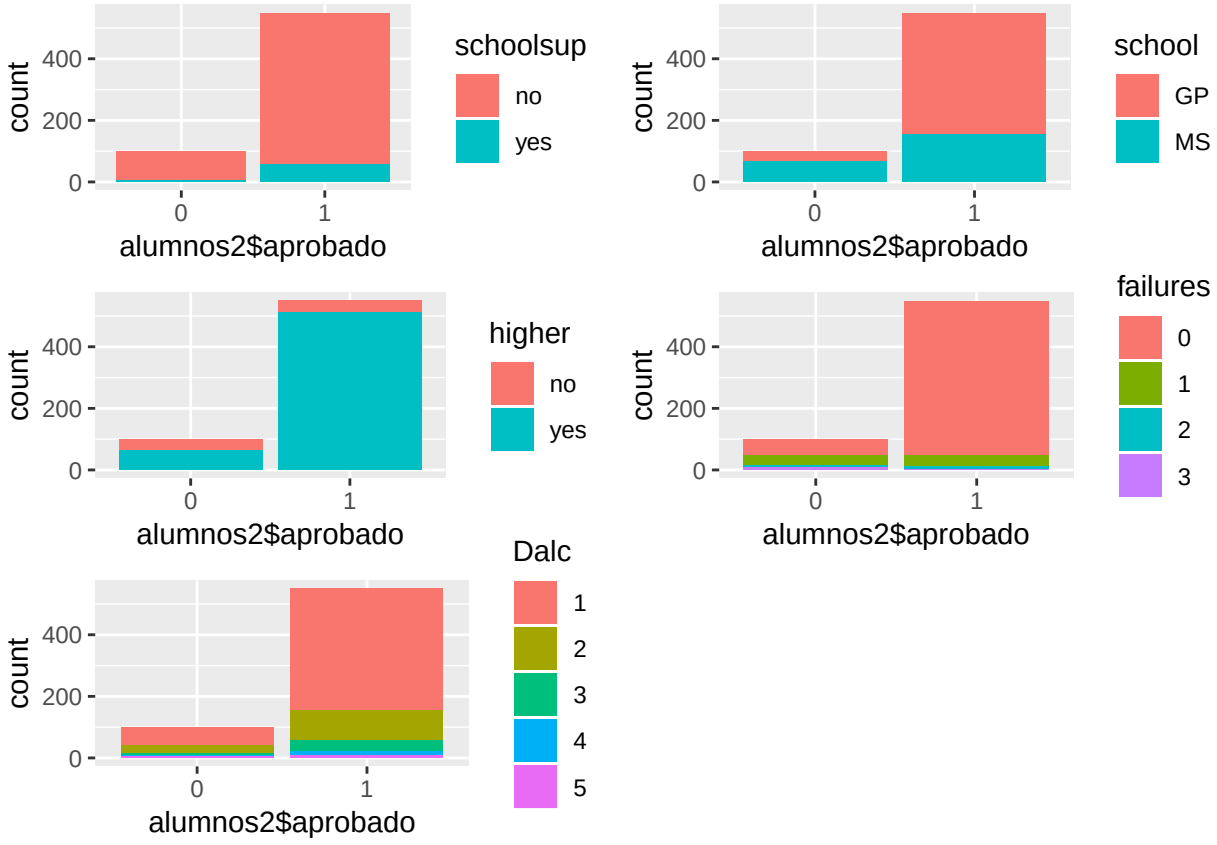
5. Representación de los resultados

Además de las diferentes representaciones a partir de tablas y gráficos hechas a lo largo de la práctica, vamos a representar graficamente las 5 variables que hemos usado para el modelo3 (modelo de regresión logística) por ser las más significativas y con las que obteníamos mejor bondad en el modelo. Veremos la proporción de clases de cada una de las variables sobre nuestro target "aprobado"

```
# Proporción de alumnos aprobados y no aprobados
proporcion_aprobados<-round(prop.table(table(alumnos2$aprobado))*100)
kable(proporcion_aprobados)
```

Var1	Freq
0	15
1	85

```
# Graficamos
barschool<-ggplot(data=alumnos2,aes(x=alumnos2$aprobado ,fill=schoolsup))+geom_bar()
barschoolsup<-ggplot(data=alumnos2,aes(x=alumnos2$aprobado ,fill=school))+geom_bar()
barhigher<-ggplot(data=alumnos2,aes(x=alumnos2$aprobado ,fill=higher))+geom_bar()
barfailures<-ggplot(data=alumnos2,aes(x=alumnos2$aprobado ,fill=failures))+geom_bar()
barDalc<-ggplot(data=alumnos2,aes(x=alumnos2$aprobado ,fill=Dalc))+geom_bar()
grid.arrange(barschool,barschoolsup,barhigher,barfailures,barDalc)
```



6. Resolución del problema

A partir de los datos obtenidos de una encuesta hecha a estudiantes de lengua portuguesa de dos escuelas de secundaria, queríamos saber como de influyentes son los diferentes factores sociales sobre la calificación de los estudiantes y poder predecir la calificación final del alumno a partir de esta información. Para ello hemos llevado a cabo una serie de pruebas estadísticas que nos han ayudado a obtener la información que estabamos

buscando. A partir del análisis de correlación y el contraste de hipótesis nos ha permitido conocer cuáles de estas variables ejercen una mayor in???uencia sobre las notas, obteniendo las siguientes variables: `school, schoolsup, higher, failures y Dalc`. El modelo de regresión lineal obtenido nos ha permitido predecir la nota final del alumno; aunque hemos visto que la prediccion de la nota es muy ineficiente, hemos comprobado mediante una tabla que el valor de la nota predicha se acerca bastante al valor real (aunque no sea el valor exacto). Finalmente hemos categorizado las notas en "aprobados=1" y "no aprobados=0" y hemos utilizado un modelo de regresión logística para predecir los alumnos aprobados o no aprobados. Este modelo nos ha dado una precisión de la predicción del 81 % de aciertos. Para finalizar hemos representado unos gráficos de las 5 variables más significantes para el modelo. A partir de la representación de los resultados podemos añadir que los alumnos aprobados representan un 85 % de los alumnos encuestados con respecto el 15 % de los alumnos que no han aprobado. Si nos fijamos en los gráficos podríamos decir que los alumnos aprobados se caracterizan por pertenecer mayoritariamente a la escuela de secundaria Gabriel Pereira (school=GP), no necesitar apoyo educativo (schoolsup=no), con intención de hacer estudios superiores (higher=yes), no haber faltado ninguna vez a clase (failures=0) y tener un consumo de alcohol diario muy bajo. Hemos conseguido predecir, con una precisión bastante buena (81 %) y un número de variables (5) basante inferior al de los datos de origen (33) , qué alumnos aprobarán.

---

7. Contribuciones

Contribuciones	Firma
Investigación previa	ABV, ERM
Redacción de las respuestas	ABV, ERM
Desarrollo código	ABV, ERM