Predicción del rendimiento académico de estudiantes

Julia García Vega

23/5/2022

Resumen

El rendimiento académico de los estudiante en los últimos cursos previos a la universidad es uno de los aspectos más importantes en cuanto a su futuro académico y posteriormente laboral. Este rendimiento puede estar influenciado ya no solo por el trabajo del estudiante si no por una multitud de factores relacionados con su entorno. Este estudio, con la información recodiga de estudiantes de dos centros educativos portugueses de cuarto de la ESO y bachiller, explora los posibles factores sociales significativos junto con la influencia de las notas de los trimestres previos con respecto a la nota en dos de las asignaturas troncales, las matemáticas y el portugués, mediante las siguientes técnicas: regresión lineal, regresión logística, redes neuroales, máquinas de soporte vectorial, Bayes ingenuo, arboles de clasificación y random forest. Se evaluán tres escenarios distintos: considerando que no se tiene ninguna nota previa, considerando que solo se tienen la nota del primer trimestre, la variable G1, y finalmente considerando que se tienen las notas de los dos trimestres previos, las variables G1 y G2. Los resultados obtenidos indican que las variables G1 y G2 están altamente correlacionadas con G3 y por ello tienen gran influencia en la nota final. En el escenario de no tener niguna nota previa, en la asignatura de portugués, se observa como el colegio Mousinho da Silveira, el género masculino, los suspensos, el apoyo del colegio, la salud regular o muy bien y las ausencias son factores significativos que influyen de forma negativa en la nota final, especialmente los suspensos. Sin embargo, la edad, el tiempo de estudio superior a diez horas, el querer continuar con su educación, una buena o muy buena relación de familia, salir poco y tener poco tiempo libre repercuten de manera positiva, especialmente el querer continuar con su educación. En la asigntura de matemáticas son menos las variables significativas pero sus coeficientes repercuten en la nota final de manera similar que en la asignatura de portugués. La pequeña diferencia entre ambas asignaturas se podría deber a la diferencia de datos entre ellas. En cuanto a la predicción de forma binaria en aprobado o suspenso los árboles de clasificación son el mejor método de predicción teniendo un porcentaje de clasificación correcta del 90% para la asignatura de portugués y de un 80% para la asignatura de matemáticas.

Introduccion

Los datos se pueden obtener de la página web *Kaggle* o del repositorio UCI,los cuales a su vez, proceden de un estudio realizado por Paulo Cortez y Alice Silva de la Universidad de Minhoa a alumnos de dos colegios portugueses que cursan las asignaturas de matemáticas y/o portugués en el año escolar 2005-2006. Se cuenta con dos archivos csv,archivo cuyos valores están separados por comas, de misma organización pero uno con la información de los individuos que cursan matemáticas, que son 395 alumnos, y otro con la información de los individuos que cursan portugués, que son 649 alumnos. Existen individuos que cursan ambas asignaturas y por lo que están en ambos.Las filas de los archivos corresponden con los alumnos y las columnas con las 33 variables. Estas variables son las siguientes:

- **school**: colegio del estudiante ('GP' = Gabriel Pereira, 'MS' = Mousinho da Silveira)
- **sex**: género del estudiante ('F' = femenino, 'M' = masculino)
- **age**: edad del estudiante
- **address**: tipo de domicilio del estudiante ('U' = urbano o 'R' = rural)
- **famsize**: tamaño de la familia del estudiante ('LE3' = menor o igual que 3, 'GT3' - mayor que 3)
- **Pstatus**: estado de convivencia de los padres del estudiante ('T' = juntos, 'A' = separados)

- **Medu**: educación de la madre del estudiante (0 = ninguna, 1 = hasta 4º EP, 2 = de 5º EP a 3º ESO, 3 = de 4º ESO a 2º Bachiller, 4 = estudios superiores)
- **Fedu**: educación del padre del estudiante (0 = ninguna, 1 = hasta 4º EP, 2 = de 5º EP a 3º ESO, 3 = de 4º ESO a 2º Bachiller, 4 = estudios superiores)
- **Mjob**: trabajo de la madre del estudiante ('teacher' = profesora, 'health' = sanitaria, 'services' = servicios civiles (p. ej. administrativa o policia), 'at_home' = ama de casa, 'other' = otro)
- **Fjob**: trabajo del padre del estudiante ('teacher' = profesor, 'health' = sanitario, 'services' = servicios civiles (p. ej. administrativo o policia), 'at_home' = amo de casa, 'other' = otro)
- **reason**: razón para elegir el colegio ('home' = cerca de casa, 'reputation' = su reputación, 'course' = sus asignaturas, 'other' = otras)
- **guardian**: tutor legal del estudiante ('mother' = madre, 'father' = padre, 'other' = otros)
- **traveltime**: tiempo de viaje de la casa del estudiante al colegio (1 = <15 min., 2 = 15-30 min., 3 = 30 min.-1 hora, 4 = >1 hora)
- **studytime**: tiempo de estudio semanal (1 = <2 horas, 2 = 2-5 horas, 3 = 5-10 horas, 4 = >10 horas)
- **failures**: numero de suspensos en asignaturas anteriores (n si n<3, si no 3)
- **schoolsup**: apoyo educativo adicional (yes, no)
- **famsup**: apoyo educativo familiar (yes, no)
- **paid**: clases extra pagadas para la asignatura en cuestión (matemáticas o portugués) (yes, no)
- **activities**: actividades extraescolares (yes, no)
- **nursery**: fue a la guarderia (yes, no)
- **higher**: quiere continuar con estudios superiores (yes, no)
- **internet**: acceso a internet en casa (yes, no)
- **romantic**: en una relación romántica (yes, no)
- **famrel**: calidad de las relaciones familiares (1 = muy mal, 2 = mal, 3 = regular, 4 = bien, 5 = muy bien)
- **freetime**: tiempo libre después del colegio (1 = nada o muy poco, 2 = poco, 3 = algo, 4 = suficiente, 5 = mucho)
- **goout**: salir con amigos (1 = nada o muy poco, 2 = poco, 3 = algo, 4 = suficiente, 5 = mucho)
- **Dalc**: consumo de alcohol en el día lectivo (1 = nada o muy poco, 2 = poco, 3 = algo, 4 = suficiente, 5 = mucho)
- **Walc**: consumo de alcohol en el fin de semana (1 = nada o muy poco, 2 = poco, 3 = algo, 4 = suficiente, 5 = mucho)
- **health**: estado de salud actual (1 = muy mal, 2 = mal, 3 = regular, 4 = bien, 5 = muy bien)
- **absences**: número de ausencias escolares (de 0 a 93)
- **G1**: nota del primer trimestre en la asignatura en cuestión (matemáticas o portugués) (de 0 a 20)
- **G2**: nota del segundo trimestre asignatura en cuestión (matemáticas o portugués) (de 0 a 20)
- **G3**: nota final en la asignatura en cuestión (matemáticas o portugués) (de 0 a 20)

La variable respuesta es la última, G3, que es la nota final en la asignatura (matemáticas o portugués). Esta variable estará fuertemente correlacionada con las variables G1 y G2 ya que se trata de la nota final. Las notas están representadas en una escala de 0 a 20 debido a que ese es el sistema de puntuación en portugal.

Para contexto, a continuación se muestra un mapa con la ubicación de los colegios de los cuales proceden los datos.

```
library(ggmap, warn.conflicts=F, quietly=T)
```

```
## Warning: package 'ggmap' was built under R version 4.0.5
```

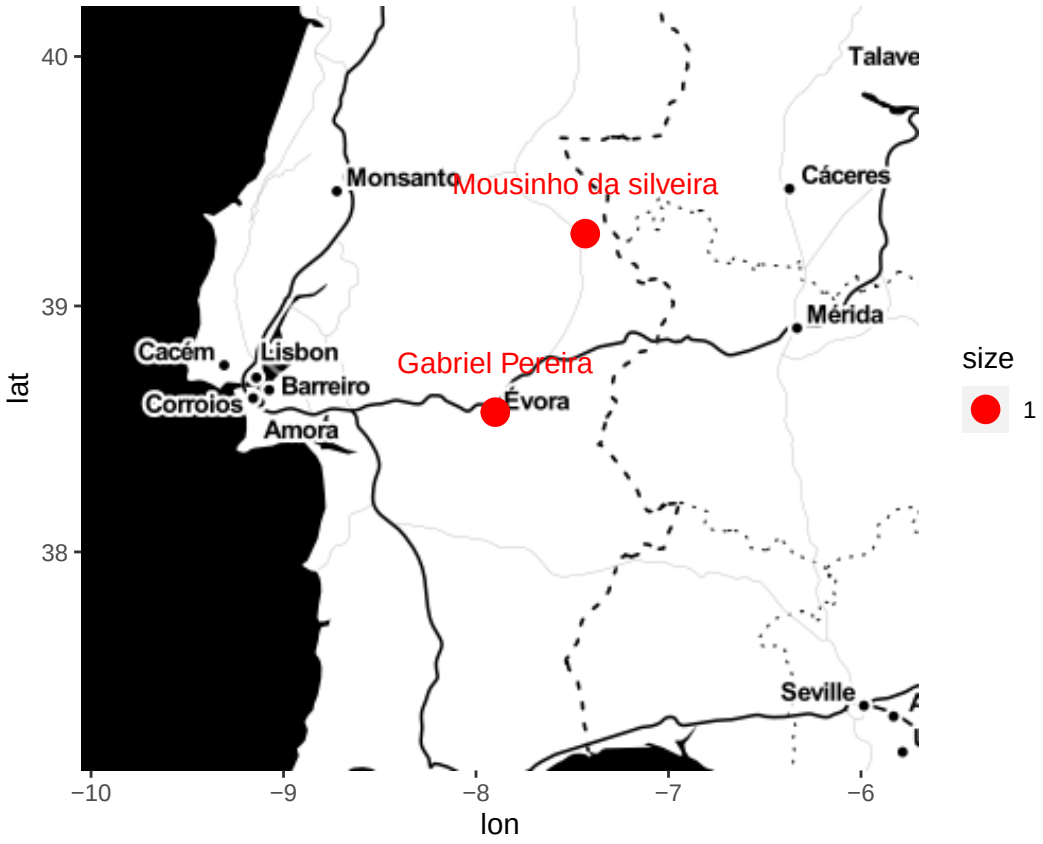
```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
bordes <- c(bottom = 37.1, top = 40.2, left = -10.05, right = -5.7)
localizacion<-data.frame("colegio"=c("Mousinho da silveira", "Gabriel Pereira"), "latitud"=c(39.2914, 38
mapK1 <- get_stamenmap(bordes, zoom=7, maptype = "toner")
```

```
## Source : http://tile.stamen.com/toner/7/60/48.png
## Source : http://tile.stamen.com/toner/7/61/48.png
## Source : http://tile.stamen.com/toner/7/60/49.png
## Source : http://tile.stamen.com/toner/7/61/49.png
ggmap(mapK1) + geom_point(data = localizacion, mapping = aes(x = altitud, y = latitud, size =1), colour=
geom_text(data=localizacion, mapping = aes(x = altitud, y = latitud+0.2, label = colegio), colour="red"
```



Métodos estadísticos

La nota final de ambas asignaturas se tratará de predecir de forma numérica mediante el método de regresión lineal múltiple y de forma binaria mediante los siguientes métodos: regresión logística, redes neuronales, máquinas de vectores de soporte, Naive Bayes y árboles de clasificación.

Regresión lineal múltiple

Este modelo es básico en la estadística. Consiste en expresar la variable que se quiere estudiar como una combinación lineal de otras variables independientes cada una con un peso determinado:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k$

Siendo y la variable respuesta, $\beta_0, \beta_1, ..., \beta_k$ los pesos denominados coeficientes de regresión y $x_1, x_2, ..., x_k$ los valores de los atributos para el individuo. Esta expresión es el valor predicho de la variable respuesta. Para que fue el valor actual habría que considerar también un error aleatorio como sumando.

La regresión lineal múltiple consiste en obtener aquellos coeficientes que hagan mínima la suma de las diferencias entre el valor actual y el predicho al cuadrado. Este método de obtención de los coeficientes de regresión se denomina método de mínimos cuadrados. Se pretende minimizar la siguiente ecuación para todos los individuos i:

$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{k}\beta_j x_{ij})$

Regresión logística

La regresión logística permite predecir el resultado de una variable categórica en función de otras variables denominadas predictoras. Esta técnica se basa en construir un modelo de regresión lineal basado en una variable respuesta transformada que inicialmente solo podía tomar los valores 0 y 1.

Sea $Pr[1|x_1, x_2, ..., x_j]$, la variable respuesta original, su transformación sería $log(Pr[1|x_1, x_2, ..., x_j])/(1 - Pr[1|x_1, x_2, ..., x_j])$. Esta transformación es denominada transformación logarítmica. El modelo resultante es: $Pr[1|x_1, x_2, ..., x_j] = 1/(1 + exp(-\beta_0 - \beta_1 x_1 - ... - \beta_k x_k))$

Al igual que la regresión lineal múltiples, la regresión logística consiste en obtener aquellos pesos que hagan máximo el logaritmo de la función de verosimilitud. Este método de obtención de los pesos se denomina método de máxima verosimilitud. Se pretende maximizar la siguiente ecuación para todos los individuos i:

$\sum_{i=1}^{n}(1 - x_i)log(1 - Pr[1|x_{i1}, x_{i2}, ..., x_{ik}) + x_i log(Pr[1|x_{i1}, x_{i2}, ..., x_{ik})]$

Redes neuronales

Este método está diseñado para simular la arquitectura del cerebro humano para crear una inteligencia artificial que permita clasificar individuos según el valor de una variable respuesta categórica. Esta arquitectura se basa en unidades procesadoras que simulan a las neuronas y conexiones entres las unidades procesadoras que tienen un peso asociado.

Una neurona artificial toma un conjunto de entradas y produce una salida de la siguiente forma. Primero calcula el valor potencial postsináptico de la neurona, generalmente con la siguiente función lineal: $net_i = \sum_{j=0} w_{ij} x_j$. A continuación aplica una función de transferencia a la suma ponderada de las entradas y obtiene la salida.

La arquitectura de una red neuronal se basa en un conjunto de uueronas que se agrupan en lo que se denominan capas. Un conjunto de una o más capas forma la red neuronal. Existen tres tipos de capas: - Capa de entrada: Aquella que recibe los datos del entorno - Capa de salida: Aquella que proporciona la respuesta - Capa oculta: Aquellas capas intermedias.

La fase de entrenamiento o aprendizaje de la red se trata de el proceso por el que se produce un ajuste de los parámetros libres de la red a partir de un proceso externo. El proceso de aprendizaje es generalmente iterativo, actualizándose los pesos, continuamente, hasta que la red alcance un rendimiento deseado o un número máximo de iteraciones. La red aprende examinando individuo a individuo, generando una predicción para cada uno y realizando los ajustes necesarios a los pesos cuando la predicción es incorrecta.

Máquinas de vectores de soporte

Una máquina de vectores de soporte (SVM) aprende una función de clasificación de una variable respuesta categórica de dos categorías resolviendo un problema de programación cuadrática.

Una SVM es un modelo que representa a los individuos en el espacio, separando las clases de la variable respuesta a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Según donde se coloquen los nuevos individuos introducidos se les clasificará en una clase o en otra, es decir, según su posición respecto a los planos.

El objetivo consiste en contruir un hiperplano o un conjunto de hiperplanos en un espacio de dimensionalidad alta que puedan ser utilizados como clasificadores.La manera más simple de realizar la separación es mediante una línea, plano o hiperplano recto pero debido a las limitaciones computacionales de las máquinas de

aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel proyecta la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquinas de aprendizaje lineal solucionando el problema.

Naive Bayes

El método de naive bayes para clasificar una variable categórica se basa en el teorema de Bayes que establece la siguiente igualdad: $P(y/x_i, ..., x_k) = P(y)P(x_i, ..., x_k|y)/P(x_i, ..., x_k)$. Debido a la suposición de indepencia se puede simplificar de la siguiente manera: $P(y/x_i, ..., x_k) = P(y)\prod_{i=1}^{k} P(x_i|y)/P(x_i, ..., x_k)$

Como $P(x_i, ..., x_k)$ es constante de para una entrada concreta, se usa la siguente regla de clasificación usando estimaciones a posteriori para $P(y)$ y $P(x_i/y)$: $\hat{y} = arg\ max_y P(y)\prod_{i=1}^{k} P(x_i|y)$

Árboles de clasificación

Un árbol de clasificación predice la clasificación de un individuo respecto a una variable respuesta según el valor de sus atributos. Estos árboles son gráficos con nodos. Al principio del árbol están presentes todos los individuos sin clasificar pero a medida que se recorre el árbol hacia abajo se van separando en dos grupos disjuntos cada vez según sus valores en determinados atributos hasta acabar todos los individuos clasificados.

La división de los individuos en grupos se puede realizar según alguno de los siguientes criterios: Índice de Gini, Chi-cuadrado (CHAID), ganancia en la información o reducción de la varianza.

El método para obtener un buen árbol de clasificación es obtener primero uno demasiado grande e ir podándolo, es decir, quitando ramas reduciendo así su complejidad.

Resultados

Pre-procesamiento de los datos

Primero se deben cargar los datos y pre-procesarlos para que puedan ser utilizados adecuadamente en un modelo.

```
notas_m <- read.csv("student-mat.csv")
notas_p <- read.csv("student-por.csv")
notas_m_corr <- notas_m ## Uso posterior explicado
notas_p_corr <- notas_p ## Uso posterior explicado
```

Antes de limpiarlos, se unen ambos archivos para realizar la limpieza de las variables únicamente en un archivo y no en dos. Para diferenciar las notas se añade a ambos archivos una columna adicional que indicará a que asignatura pertenecen los datos de esa fila. Esta variable se eliminará al terminar la limpieza del archivo al volver a separar las asignaturas.

```
notas_m$asignatura<-"M"
notas_p$asignatura<-"P"
notas<-rbind(notas_p,notas_m)
head(notas)
```

```
##   school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4  at_home  teacher     course
## 2     GP   F  17       U     GT3       T    1    1  at_home    other     course
## 3     GP   F  15       U     LE3       T    1    1  at_home    other      other
## 4     GP   F  15       U     GT3       T    4    2   health services       home
## 5     GP   F  16       U     GT3       T    3    3    other    other       home
## 6     GP   M  16       U     LE3       T    4    3 services    other reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0       yes     no   no         no
```

```
## 2   father          1         2        0        no    yes   no         no
## 3   mother          1         2        0       yes     no   no         no
## 4   mother          1         3        0        no    yes   no        yes
## 5   father          1         2        0        no    yes   no         no
## 6   mother          1         2        0        no    yes   no        yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1     yes    yes       no       no      4        3     4    1    1      3
## 2      no    yes      yes       no      5        3     3    1    1      3
## 3     yes    yes      yes       no      4        3     2    2    3      3
## 4     yes    yes      yes      yes      3        2     2    1    1      5
## 5     yes    yes       no       no      4        3     2    1    2      5
## 6     yes    yes      yes       no      5        4     2    1    2      5
##   absences G1 G2 G3 asignatura
## 1        4  0 11 11          P
## 2        2  9 11 11          P
## 3        6 12 13 12          P
## 4        0 14 14 14          P
## 5        0 11 13 13          P
## 6        6 12 12 13          P
```

Se obtiene a continuación los tipos de las variables que han sido asignados automáticamente al cargar los archivos y se corrigen los que haga falta.

```
sapply(notas,class)
```

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

Todas las variables excepto la edad, las ausencias y las notas deben ser factores por lo que se procede a su cambio. Además se le da nombre a los niveles que estaban representados por números y se cambia el tipo de las variables de tipo integer por el tipo numeric.

```
notas$school<-factor(notas$school)

notas$sex<-factor(notas$sex)
levels(notas$sex)<-c("mujer","hombre")

notas$address<-factor(notas$address)
levels(notas$address)<-c("Urban","Rural")

notas$famsize<-factor(notas$famsize)

notas$Pstatus<-factor(notas$Pstatus)
levels(notas$Pstatus)<-c("juntos","separados")

notas$Medu<-factor(notas$Medu)
```

```
levels(notas$Medu)<-c("ninguna", "<=4ºEP", "5ºEP-3ºESO", "4ºESO-2ºBachiller", "estudios superiores")

notas$Fedu<-factor(notas$Fedu)
levels(notas$Fedu)<-c("ninguna", "<=4ºEP", "5ºEP-3ºESO", "4ºESO-2ºBachiller", "estudios superiores")

notas$Mjob<-factor(notas$Mjob)
notas$Fjob<-factor(notas$Fjob)
notas$reason<-factor(notas$reason)
notas$guardian<-factor(notas$guardian)

notas$traveltime<-factor(notas$traveltime)
levels(notas$traveltime)<-c("<15 min", "15-30 min", "30 min.-1 hora", ">1 hora")

notas$studytime<-factor(notas$studytime)
levels(notas$studytime)<-c("<2 horas", "2-5 horas", "5-10 horas", ">10 horas")

notas$failures<-factor(notas$failures)
levels(notas$failures)<-c("0","1","2",">=3")

notas$schoolsup<-factor(notas$schoolsup)
notas$famsup<-factor(notas$famsup)
notas$paid<-factor(notas$paid)
notas$activities<-factor(notas$activities)
notas$nursery<-factor(notas$nursery)
notas$higher<-factor(notas$higher)
notas$internet<-factor(notas$internet)
notas$romantic<-factor(notas$romantic)

notas$famrel<-factor(notas$famrel)
levels(notas$famrel)<-c("muy mal", "mal", "regular", "bien", "muy bien")

notas$freetime<-factor(notas$freetime)
levels(notas$freetime)<-c("nada", "poco", "algo", "suficiente", "mucho")

notas$goout<-factor(notas$goout)
levels(notas$goout)<-c("nada", "poco", "algo", "suficiente", "mucho")

notas$Dalc<-factor(notas$Dalc)
levels(notas$Dalc)<-c("nada", "poco", "algo", "suficiente", "mucho")

notas$Walc<-factor(notas$Walc)
levels(notas$Walc)<-c("nada", "poco", "algo", "suficiente", "mucho")

notas$health<-factor(notas$health)
levels(notas$health)<-c("muy mal", "mal", "regular", "bien", "muy bien")

notas$asignatura<-factor(notas$asignatura)
levels(notas$asignatura)<-c("Matemáticas","Portugués")

Classes=sapply(notas,class)
for (i in 1:ncol(notas))
if (Classes[i]=='integer') notas[[i]]=as.numeric(notas[[i]])
```

```
head(notas)
```

```
##   school    sex age address famsize   Pstatus                Medu
## 1     GP  mujer  18   Rural     GT3    juntos estudios superiores
## 2     GP  mujer  17   Rural     GT3 separados              <=4ºEP
## 3     GP  mujer  15   Rural     LE3 separados              <=4ºEP
## 4     GP  mujer  15   Rural     GT3 separados estudios superiores
## 5     GP  mujer  16   Rural     GT3 separados   4ºESO-2ºBachiller
## 6     GP hombre  16   Rural     LE3 separados estudios superiores
##                  Fedu     Mjob     Fjob     reason guardian traveltime
## 1 estudios superiores  at_home  teacher     course   mother  15-30 min
## 2             <=4ºEP  at_home    other     course   father    <15 min
## 3             <=4ºEP  at_home    other      other   mother    <15 min
## 4          5ºEP-3ºESO   health services       home   mother    <15 min
## 5   4ºESO-2ºBachiller    other    other       home   father    <15 min
## 6   4ºESO-2ºBachiller services    other reputation   mother    <15 min
##    studytime failures schoolsup famsup paid activities nursery higher internet
## 1  2-5 horas        0       yes     no   no         no     yes    yes       no
## 2  2-5 horas        0        no    yes   no         no      no    yes      yes
## 3  2-5 horas        0       yes     no   no         no     yes    yes      yes
## 4 5-10 horas        0        no    yes   no        yes     yes    yes      yes
## 5  2-5 horas        0        no    yes   no         no     yes    yes       no
## 6  2-5 horas        0        no    yes   no        yes     yes    yes      yes
##   romantic   famrel   freetime      goout Dalc Walc   health absences G1 G2 G3
## 1       no     bien       algo suficiente nada nada  regular        4  0 11 11
## 2       no muy bien       algo       algo nada nada  regular        2  9 11 11
## 3       no     bien       algo       poco poco algo  regular        6 12 13 12
## 4      yes  regular       poco       poco nada nada muy bien        0 14 14 14
## 5       no     bien       algo       poco nada poco muy bien        0 11 13 13
## 6       no muy bien suficiente       poco nada poco muy bien        6 12 12 13
##   asignatura
## 1  Portugués
## 2  Portugués
## 3  Portugués
## 4  Portugués
## 5  Portugués
## 6  Portugués
```

```
sapply(notas,class)
```

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

A continuación, se comprueba si existen valores nulos.

```
sapply(notas, function(x) sum(is.na(x)))
```

```
##     school        sex        age    address    famsize    Pstatus       Medu
```

```
##          0          0          0          0          0          0          0
##       Fedu       Mjob       Fjob     reason   guardian traveltime  studytime
##          0          0          0          0          0          0          0
##   failures  schoolsup     famsup       paid activities    nursery     higher
##          0          0          0          0          0          0          0
##   internet   romantic     famrel   freetime      goout       Dalc       Walc
##          0          0          0          0          0          0          0
##     health   absences         G1         G2         G3 asignatura
##          0          0          0          0          0          0
```

No existen valores nulos por lo que con esto se termina la limpieza. Se procede ahora a volver a separar la matriz en dos distinguiendo por la asignatura y a eliminar la variable auxiliar asignatura. Las primeras 649 filas corresponden a la asignatura de portugués y las restantes a la asignatura de matemáticas.

```
notas_p<-notas[1:649,]
notas_m<-notas[650:1044,]
notas_p<-notas_p[,!(names(notas) %in% "asignatura")]
notas_m<-notas_m[,!(names(notas) %in% "asignatura")]
```

Se crea para ambas asignaturas una nueva variable que corresponda a si un alumno ha aprobado o suspendido para posteriormente también poder clasificar a los alumnos por esta variable.

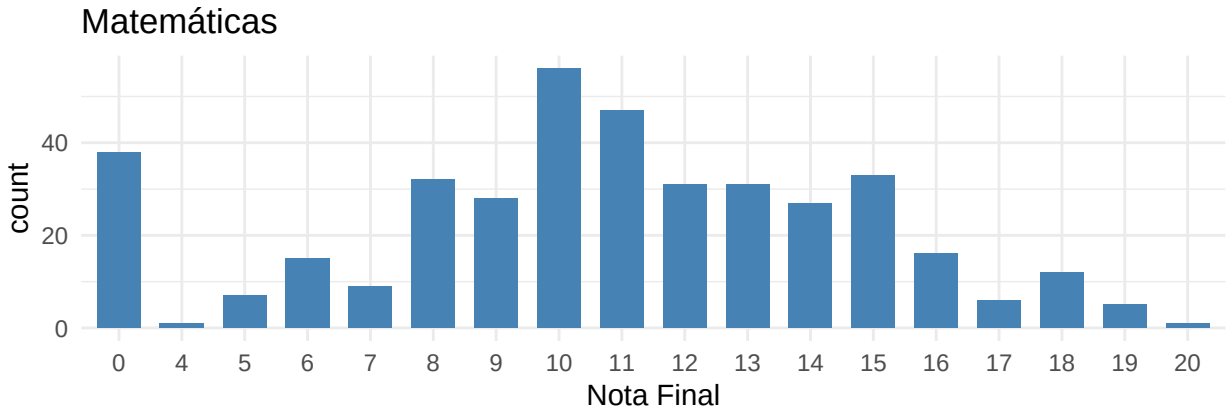
```
notas_p$calificacion<-ifelse(notas_p$G3 < 10, "suspenso", "aprobado")
notas_p$calificacion<-factor(notas_p$calificacion)
notas_m$calificacion<-ifelse(notas_m$G3 < 10, "suspenso", "aprobado")
notas_m$calificacion<-factor(notas_m$calificacion)
```

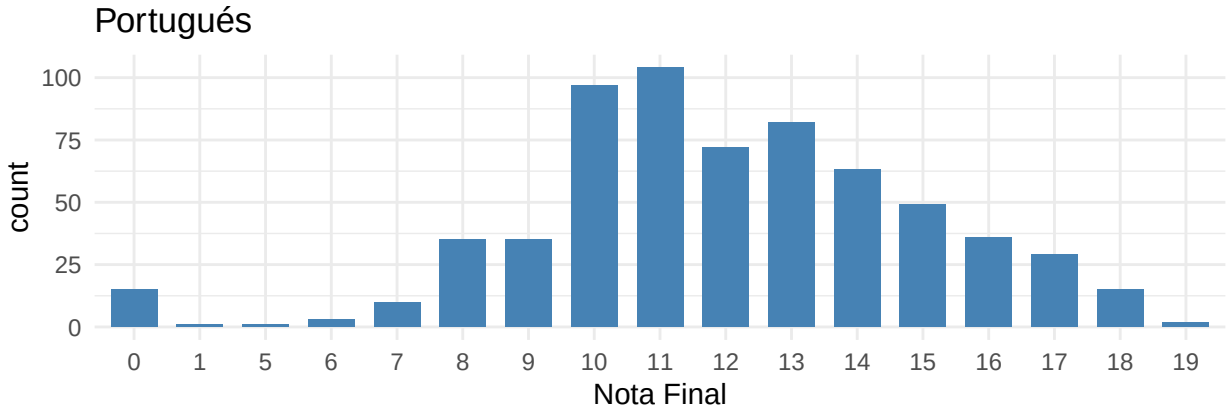
Por último, se comprueba la distribución de G3 en ambas asignaturas para detectar si existe alguna anomalía.

```
library(ggplot2)
q1 = ggplot(notas_m, aes(x=as.factor(G3))) +
  geom_bar(stat="count", width=0.7, fill="steelblue") + labs(x="Nota Final", title="Matemáticas")+
  theme_minimal()
q2 = ggplot(notas_p, aes(x=as.factor(G3))) +
  geom_bar(stat="count", width=0.7, fill="steelblue") + labs(x="Nota Final", title="Portugués")+
  theme_minimal()
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.0.5
```

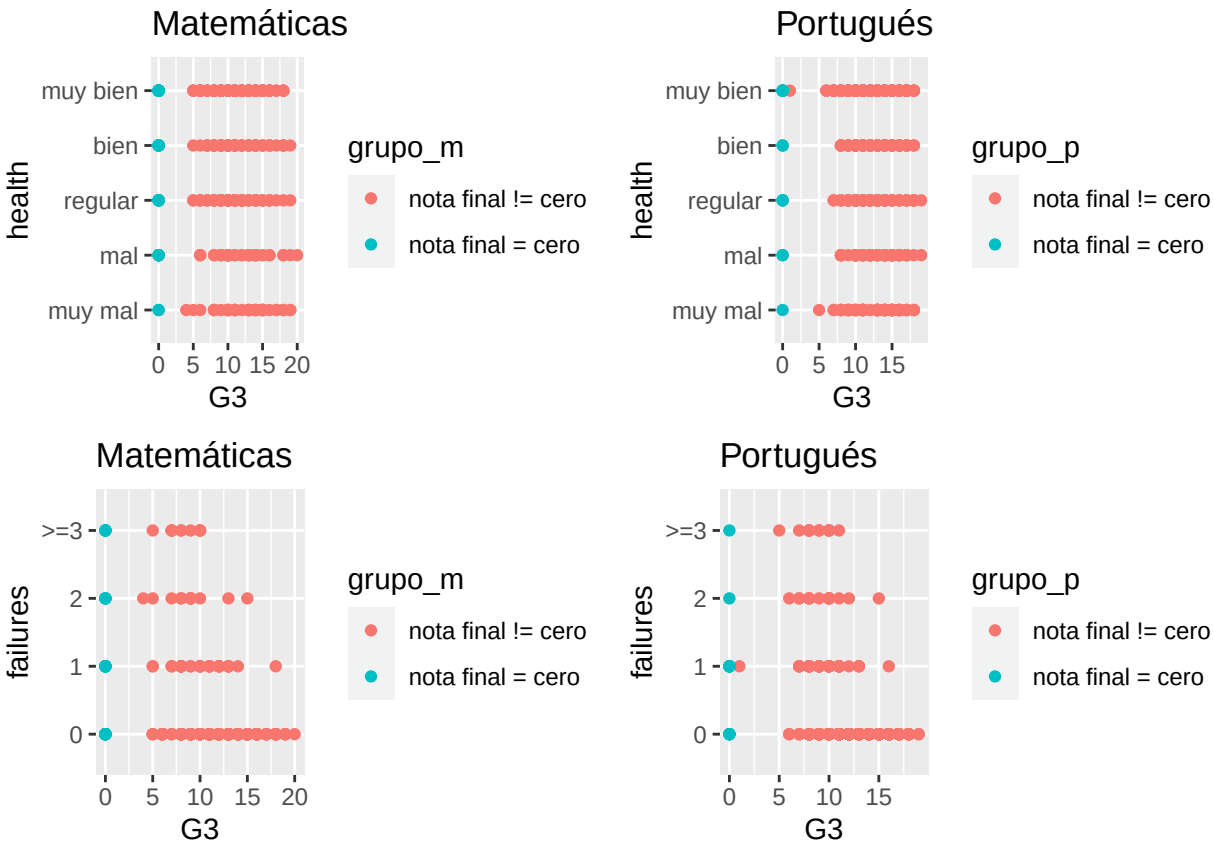
```
grid.arrange(q1,q2, nrow = 2, ncol=1)
```



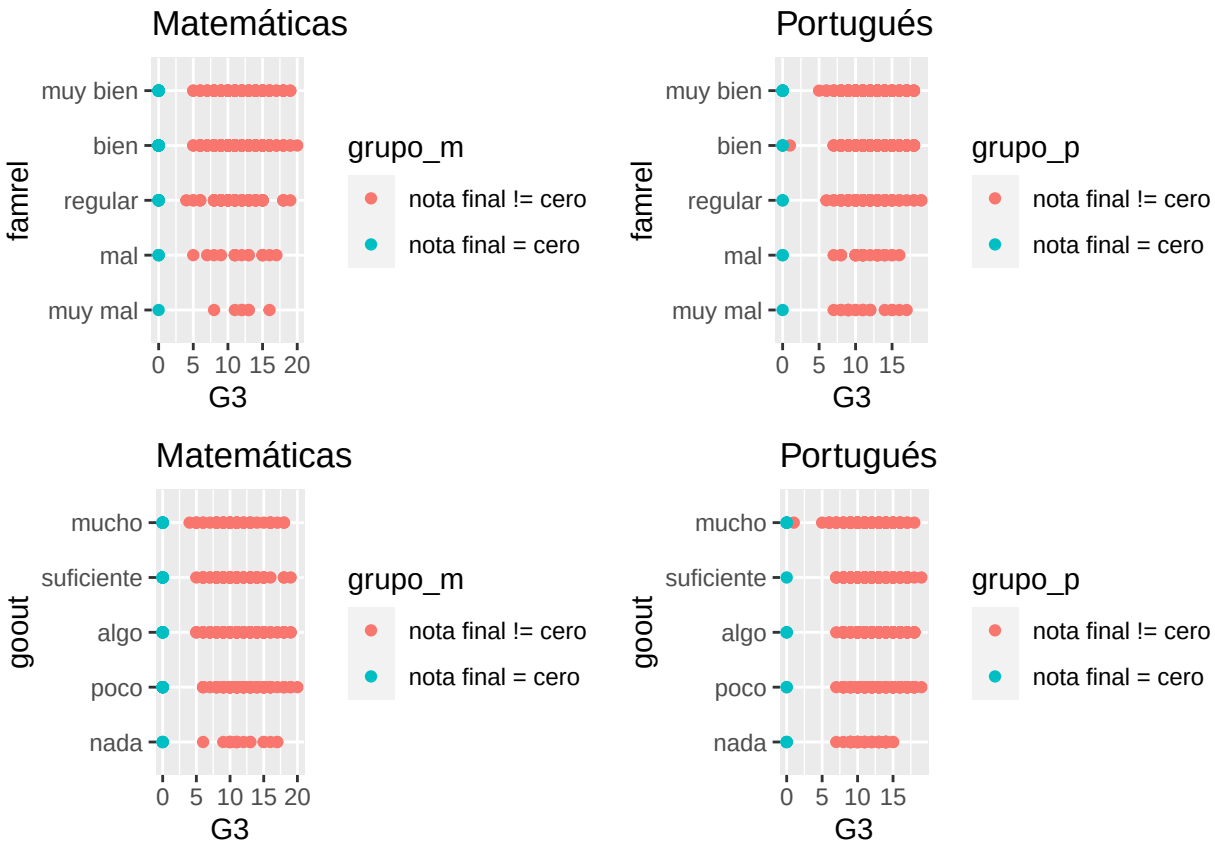


La nota final, en ambas asignaturas, a simple vista sigue una distribución normal exceptuando la frequencia anómala de la nota 0 que destaca notablemente más en matemáticas que en portugués. Esto significa que hay un error ya que no debe haber tantos alumnos con un cero de nota. Esta alta frequencia del cero se puede deber a múltiples cosas: a lo mejor los valores nulos se han sustituido en la base de datos con un cero, o los no presentados también se han calificado con un cero o existe alguna explicación relacionada con el resto de variables. Se dejan a continuación algunos diagramas de dispersión para comprobar este último caso .

```
grupo_m <- as.factor(ifelse(notas_m$G3 > 0, "nota final != cero", "nota final = cero"))
grupo_p <- as.factor(ifelse(notas_p$G3 > 0, "nota final != cero", "nota final = cero"))
q1<-qplot(G3, health, data = notas_m, colour = grupo_m) + labs(title="Matemáticas")
q2<-qplot(G3, health, data = notas_p, colour = grupo_p) + labs(title="Portugués")
q3<-qplot(G3, failures, data = notas_m, colour = grupo_m) + labs(title="Matemáticas")
q4<-qplot(G3, failures, data = notas_p, colour = grupo_p) + labs(title="Portugués")
grid.arrange(q1,q2,q3,q4, nrow = 2, ncol=2)
```




```
q1<-qplot(G3, famrel, data = notas_m, colour = grupo_m) + labs(title="Matemáticas")
q2<-qplot(G3, famrel, data = notas_p, colour = grupo_p) + labs(title="Portugués")
q3<-qplot(G3, goout, data = notas_m, colour = grupo_m) + labs(title="Matemáticas")
q4<-qplot(G3, goout, data = notas_p, colour = grupo_p) + labs(title="Portugués")
grid.arrange(q1,q2,q3,q4, nrow = 2, ncol=2)
```

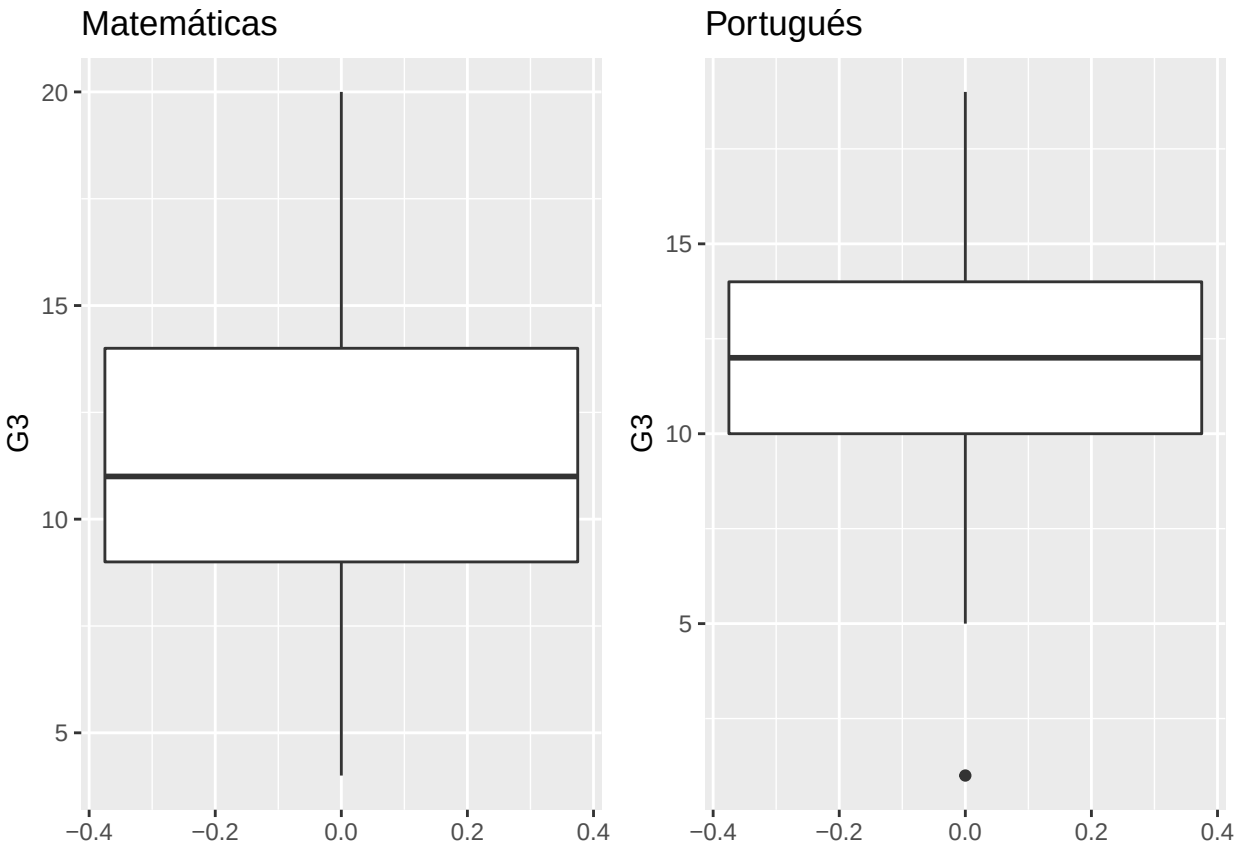


Se observa como los alumnos tiene un cero de nota final están presentes en todos los grupos de las varibles sin separarse en ninguno. Por ello, para que se cumpla la normalidad de los datos se procede a eliminar los datos de estos alumnos para no tenerlos en cuenta.

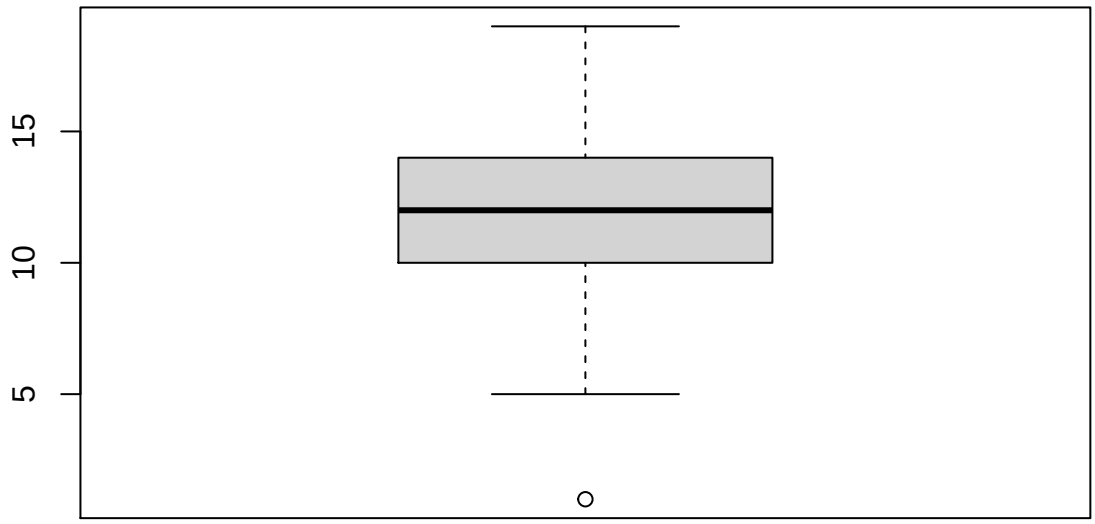
```
notas_m<-notas_m[notas_m$G3>0, ]
notas_p<-notas_p[notas_p$G3>0, ]
notas_m_corr <- notas_m_corr[notas_m_corr$G3>0, ] ## Uso posterior explicado
notas_p_corr <- notas_p_corr[notas_p_corr$G3>0, ] ## Uso posterior explicado
```

Se comprueba por último la existencia de outliers representado el diagrama de cajas.

```
p1 <- ggplot(notas_m, aes(y=G3)) +
geom_boxplot() + labs(title="Matemáticas")
p2 <- ggplot(notas_p, aes(y=G3)) +
geom_boxplot() + labs(title="Portugués")
grid.arrange(p1,p2, nrow = 1, ncol=2)
```




```
boxplot(notas_p$G3)$out
```




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

Existe un dato anómalo según el criterio elegido en la asignatura de portugués por lo que se procede a su eliminación.

```
notas_p<-notas_p[notas_p$G3!=1, ]
```

Descriptiva básica y visualización de los datos

Se realiza primero una descriptiva básica y una visualización del archivo de datos para conocer y familiarizarse con los datos a analizar.

La función summary proporcionará la descriptiva básica. De las variables númericas calculará el mínimo, máximo, media aritmética y los percentiles 25, 50 y 75. De las variables que son factores proporcionará las frecuencias absolutas de los distintos niveles de los factores.

```
summary(notas_m)
```

```
##  school       sex           age         address    famsize        Pstatus
##  GP:315   mujer :185   Min.   :15.00   Urban: 78   GT3:250   juntos   : 39
##  MS: 42   hombre:172   1st Qu.:16.00   Rural:279   LE3:107   separados:318
##                        Median :17.00
##                        Mean   :16.66
##                        3rd Qu.:18.00
##                        Max.   :22.00
##                   Medu                       Fedu           Mjob
##  ninguna            :  3   ninguna            :  2   at_home : 50
##  <=4ºEP             : 50   <=4ºEP             : 71   health  : 32
```

```
##  5ºEP-3ºESO         : 89   5ºEP-3ºESO         :102   other   :127
##  4ºESO-2ºBachiller  : 90   4ºESO-2ºBachiller  : 94   services: 94
##  estudios superiores:125   estudios superiores: 88   teacher : 54
##
##       Fjob            reason      guardian            traveltime
##  at_home : 17   course    :126   father: 82   <15 min       :236
##  health  : 18   home      : 97   mother:248   15-30 min     : 95
##  other   :196   other     : 35   other : 27   30 min.-1 hora: 19
##  services:100   reputation: 99                >1 hora       :  7
##  teacher : 26
##
##      studytime   failures  schoolsup famsup     paid     activities nursery
##  <2 horas  : 92   0  :294   no :307   no :138   no :184   no :177    no : 71
##  2-5 horas :182   1  : 40   yes: 50   yes:219   yes:173   yes:180    yes:286
##  5-10 horas: 59   2  : 12
##  >10 horas : 24   >=3: 11
##
##
##  higher    internet  romantic       famrel          freetime          goout
##  no : 14   no : 58   no :245   muy mal :  7   nada      : 17   nada      : 19
##  yes:343   yes:299   yes:112   mal     : 15   poco      : 60   poco      : 94
##                                regular : 61   algo      :136   algo      :122
##                                bien    :178   suficiente:106   suficiente: 77
##                                muy bien: 96   mucho     : 38   mucho     : 45
##
##         Dalc             Walc          health       absences
##  nada      :250   nada      :133   muy mal : 45   Min.   : 0.000
##  poco      : 64   poco      : 73   mal     : 38   1st Qu.: 2.000
##  algo      : 25   algo      : 77   regular : 83   Median : 4.000
##  suficiente:  9   suficiente: 48   bien    : 58   Mean   : 6.317
##  mucho     :  9   mucho     : 26   muy bien:133   3rd Qu.: 8.000
##                                                  Max.   :75.000
##       G1              G2              G3          calificacion
##  Min.   : 3.00   Min.   : 5.00   Min.   : 4.00   aprobado:265
##  1st Qu.: 9.00   1st Qu.: 9.00   1st Qu.: 9.00   suspenso: 92
##  Median :11.00   Median :11.00   Median :11.00
##  Mean   :11.27   Mean   :11.36   Mean   :11.52
##  3rd Qu.:14.00   3rd Qu.:14.00   3rd Qu.:14.00
##  Max.   :19.00   Max.   :19.00   Max.   :20.00
```

```
summary(notas_p)
```

```
##  school       sex           age         address    famsize        Pstatus
##  GP:421   mujer :376   Min.   :15.00   Urban:187   GT3:443   juntos   : 78
##  MS:212   hombre:257   1st Qu.:16.00   Rural:446   LE3:190   separados:555
##                        Median :17.00
##                        Mean   :16.72
##                        3rd Qu.:18.00
##                        Max.   :22.00
##                  Medu                       Fedu           Mjob
##  ninguna            :  6   ninguna            :  7   at_home :131
##  <=4ºEP             :138   <=4ºEP             :167   health  : 48
##  5ºEP-3ºESO         :181   5ºEP-3ºESO         :202   other   :249
##  4ºESO-2ºBachiller  :137   4ºESO-2ºBachiller  :130   services:135
##  estudios superiores:171   estudios superiores:127   teacher : 70
```

```
##
##       Fjob            reason      guardian            traveltime
##  at_home : 41   course    :279   father:149   <15 min       :360
##  health  : 23   home      :146   mother:444   15-30 min     :205
##  other   :359   other     : 67   other : 40   30 min.-1 hora: 52
##  services:175   reputation:141                >1 hora       : 16
##  teacher : 35
##
##      studytime   failures  schoolsup famsup     paid     activities nursery
##  <2 horas  :204   0  :543   no :566   no :240   no :595   no :326    no :126
##  2-5 horas :297   1  : 62   yes: 67   yes:393   yes: 38   yes:307    yes:507
##  5-10 horas: 97   2  : 15
##  >10 horas : 35   >=3: 13
##
##
##  higher    internet  romantic       famrel          freetime          goout
##  no : 64   no :144   no :404   muy mal : 21   nada      : 45   nada      : 44
##  yes:569   yes:489   yes:229   mal     : 27   poco      :104   poco      :143
##                                regular : 99   algo      :246   algo      :202
##                                bien    :313   suficiente:175   suficiente:140
##                                muy bien:173   mucho     : 63   mucho     :104
##
##          Dalc             Walc          health       absences
##  nada      :443   nada      :243   muy mal : 89   Min.   : 0.000
##  poco      :117   poco      :148   mal     : 76   1st Qu.: 0.000
##  algo      : 43   algo      :114   regular :121   Median : 2.000
##  suficiente: 13   suficiente: 85   bien    :106   Mean   : 3.752
##  mucho     : 17   mucho     : 43   muy bien:241   3rd Qu.: 6.000
##                                                   Max.   :32.000
##        G1              G2              G3          calificacion
##  Min.   : 0.0   Min.   : 5.00   Min.   : 5.00   aprobado:549
##  1st Qu.:10.0   1st Qu.:10.00   1st Qu.:10.00   suspenso: 84
##  Median :11.0   Median :12.00   Median :12.00
##  Mean   :11.5   Mean   :11.76   Mean   :12.21
##  3rd Qu.:13.0   3rd Qu.:13.00   3rd Qu.:14.00
##  Max.   :19.0   Max.   :19.00   Max.   :19.00
```

Para la visualización de los datos se utilizarán los paquetes corrplot, ggplot2, gridExtra (ya añadidaa anteriormente las dos últimas) y ggmosaic que ofrecen multitud de gráficos y posibilidades. Se plantearán distintas representaciones gráficas para las variables.

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggmosaic)
```

```
## Warning: package 'ggmosaic' was built under R version 4.0.5
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.0.5
```

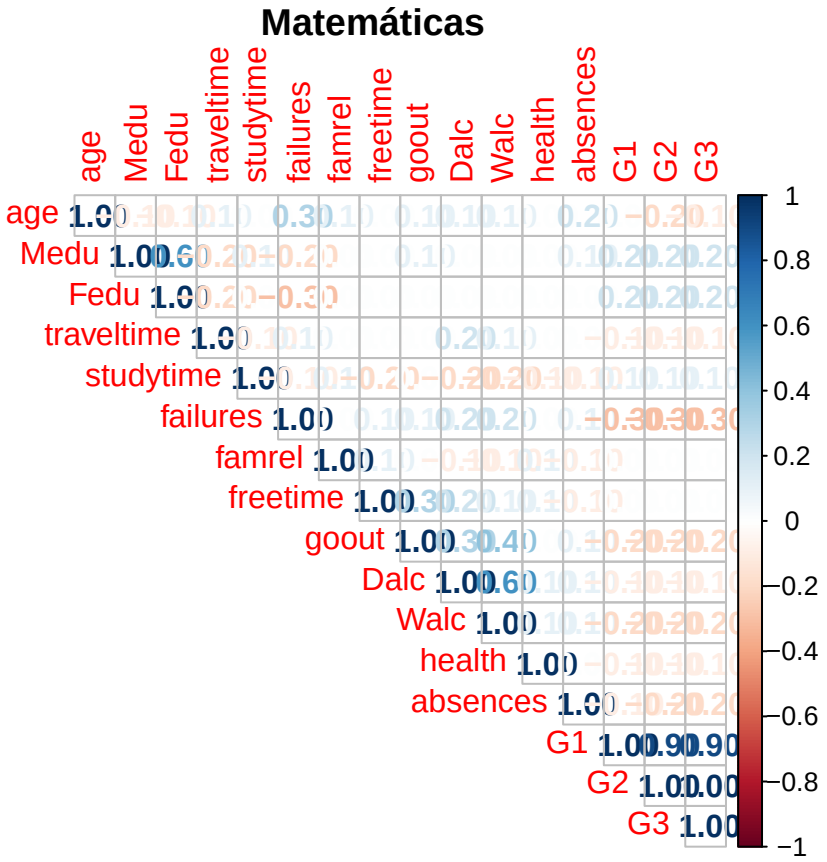
```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggmap':
##
```

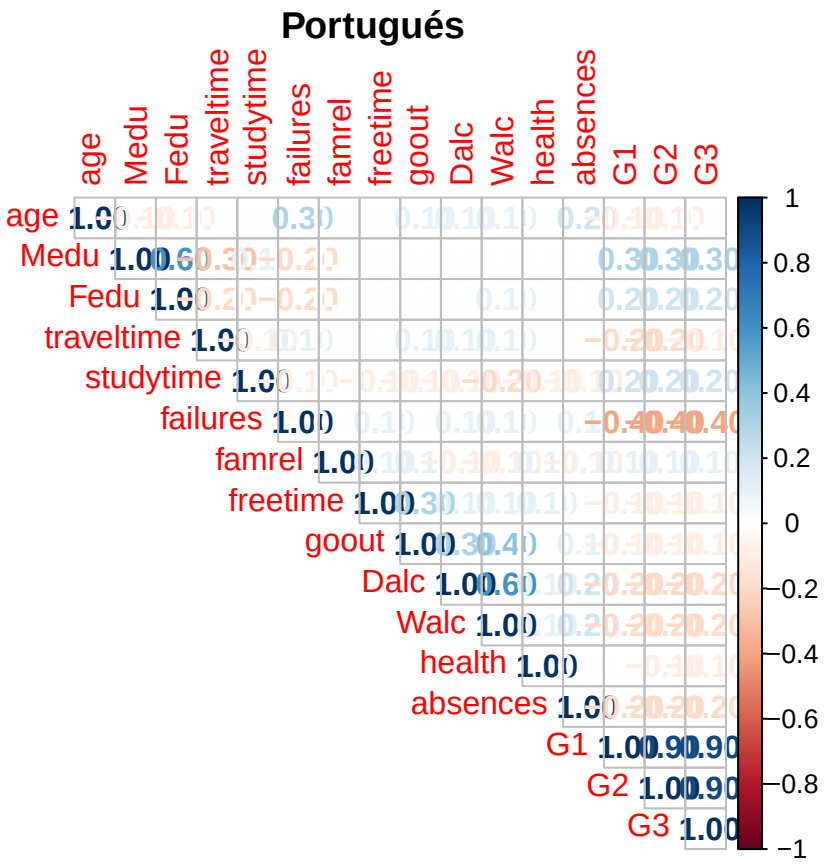
```
##     theme_nothing
```

Primero se obtiene la matriz de correlaciones entre todas las variables inicialmente númericas. Al inicio, al cargar los datos se guardaron en una matrices auxiliares los datos con las clases asignadas automáticas a las variables. En la matriz a continuación se analizará la correlación entre las variables numéricas de cada matriz correspondiente a una asignatura.

```
var_numericas <- Filter(is.numeric, notas_m_corr)
correlacion<-round(cor(var_numericas), 1)
corrplot(correlacion, method="number", type="upper",title="Matemáticas", mar=c(0,0,1,0))
```



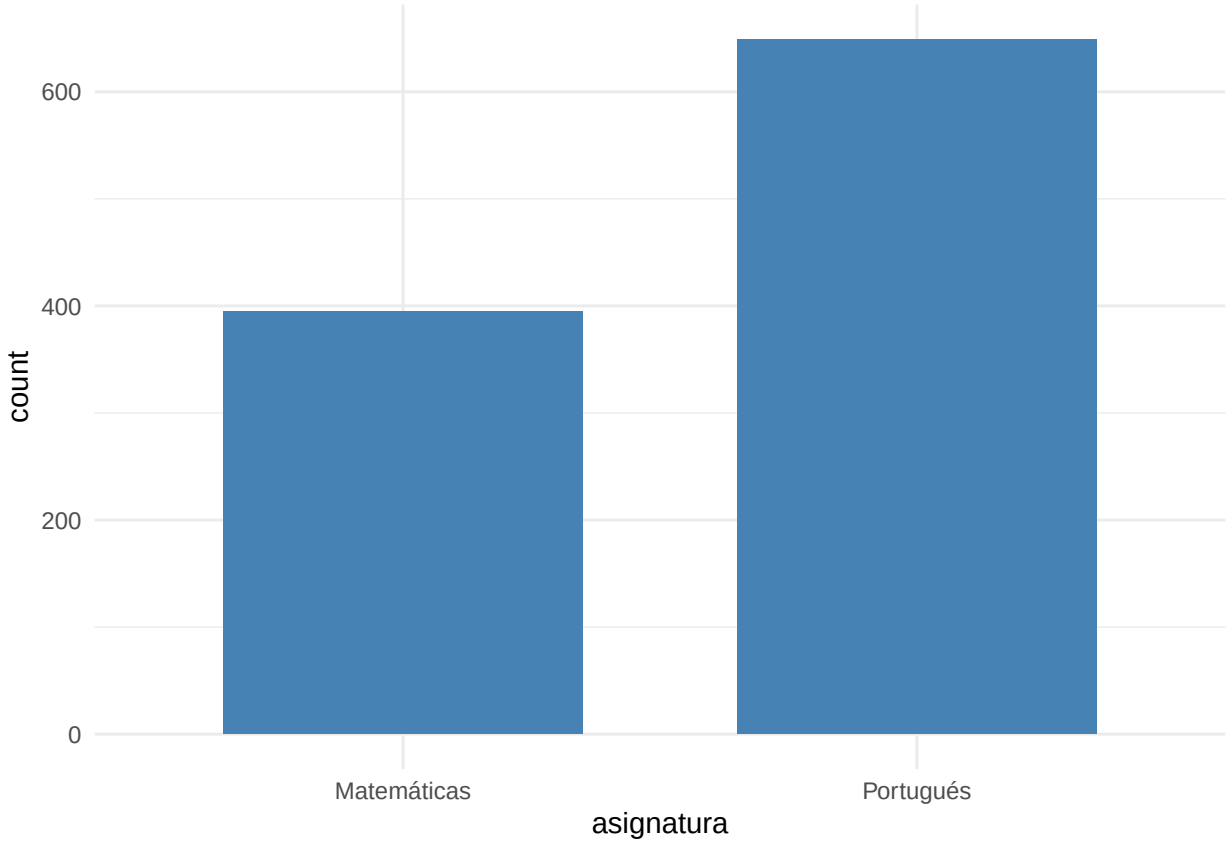

```
var_numericas <- Filter(is.numeric, notas_p_corr)
correlacion<-round(cor(var_numericas), 1)
corrplot(correlacion, method="number", type="upper",title="Portugués", mar=c(0,0,1,0))
```



Aquellas variables que presentan números de colores mas fuertes, ya sea azul o naranja, se dice que están correlacionadas. En ambas asignaturas, como es de esperar, las tres notas están altamente correlacionadas directamente. Cuando se realice el análisis de la nota final, la variable G3, se realizarán tres casos: considerando que no se tiene ninguna nota previa, considerando que solo se tienen la nota del primer trimestre, la variable G1, y finalmente considerando que se tienen las notas de los dos trimestres previos, las variables G1 y G2. En cuanto al resto de variables, se observa una baja correlación directa entre la educación del padre y la eduación de la madre, es decir, ambos padres suelen haber estudiado lo mismo; y también una baja correlación inversa entre el número de suspensos y las distintas notas, es decir, cuanto más aumenta el número de suspenso más descienden las notas.

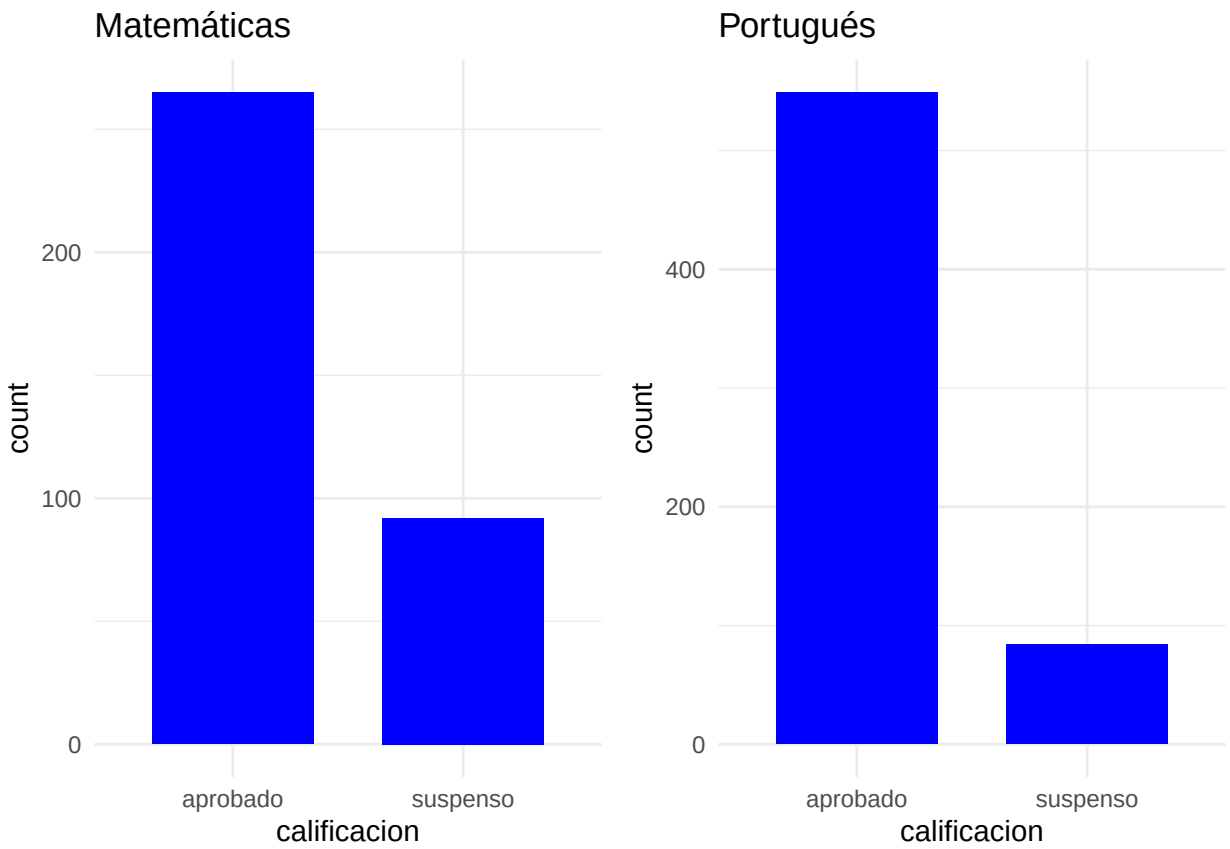
A continuación, se va a estudiar las variables nominales mediante diagramas de barras diferenciando por asignatura.

```
q = ggplot(notas, aes(x=asignatura)) +
  geom_bar(stat="count", width=0.7, fill="steelblue") +
  theme_minimal()
plot(q)
```



Como ya se ha mencionado antes, se observa que el número de datos recogidos para al asignatura de matemáticas es menor que para la asignatura de portugués.

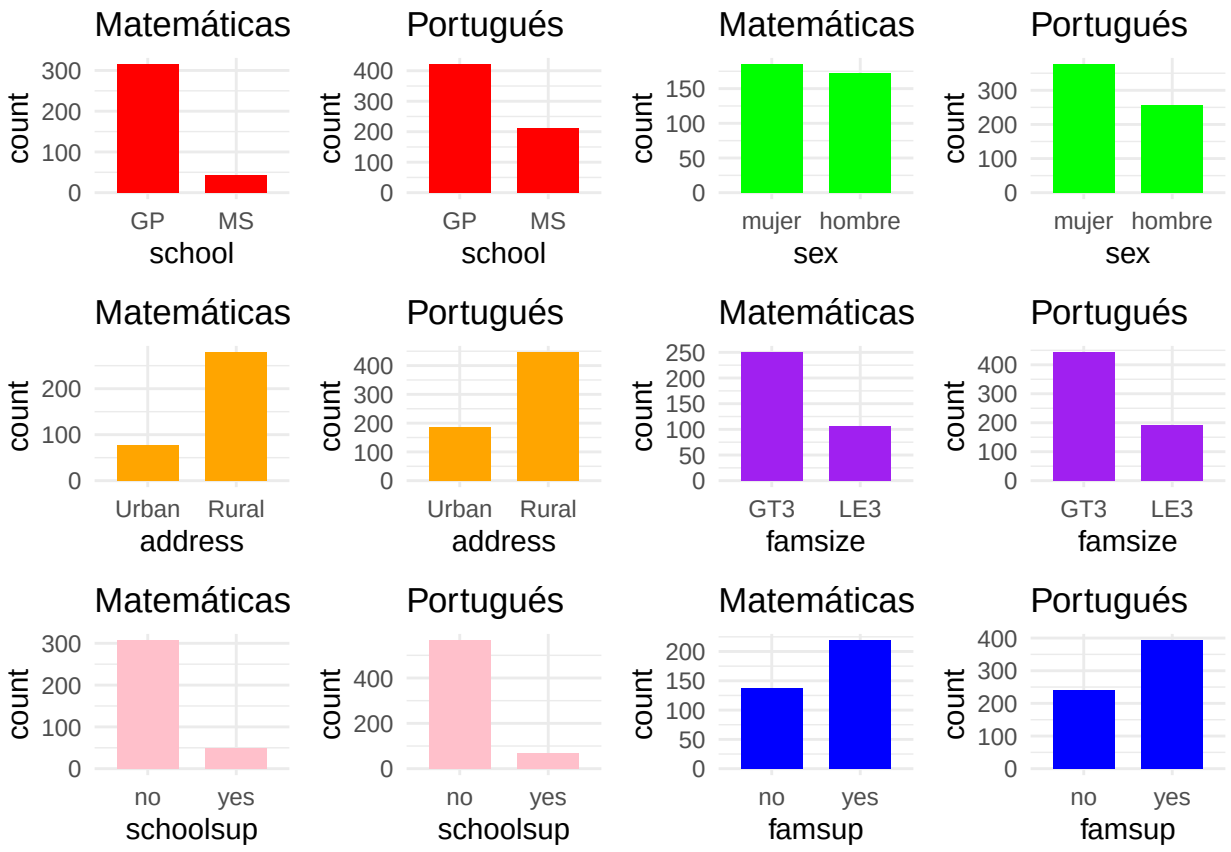
```
q1 = ggplot(notas_m, aes(x=calificacion)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Matemáticas") +
  theme_minimal()
q2 = ggplot(notas_p, aes(x=calificacion)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Portugués") +
  theme_minimal()
grid.arrange(q1, q2,nrow = 1, ncol=2)
```



La proporción de suspensos en la asignatura de matemáticas es mayor que en el asignatura de portugués.

```
g1 = ggplot(notas_m, aes(x=school)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Matemáticas") +
  theme_minimal()
g2 = ggplot(notas_p, aes(x=school)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Portugués") +
  theme_minimal()
g3 = ggplot(notas_m, aes(x=sex)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Matemáticas") +
  theme_minimal()
g4 = ggplot(notas_p, aes(x=sex)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Portugués") +
  theme_minimal()
g5 = ggplot(notas_m, aes(x=address)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Matemáticas") +
  theme_minimal()
g6 = ggplot(notas_p, aes(x=address)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Portugués") +
  theme_minimal()
g7 = ggplot(notas_m, aes(x=famsize)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Matemáticas") +
  theme_minimal()
g8 = ggplot(notas_p, aes(x=famsize)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Portugués") +
  theme_minimal()
g29 = ggplot(notas_m, aes(x=schoolsup)) +
```

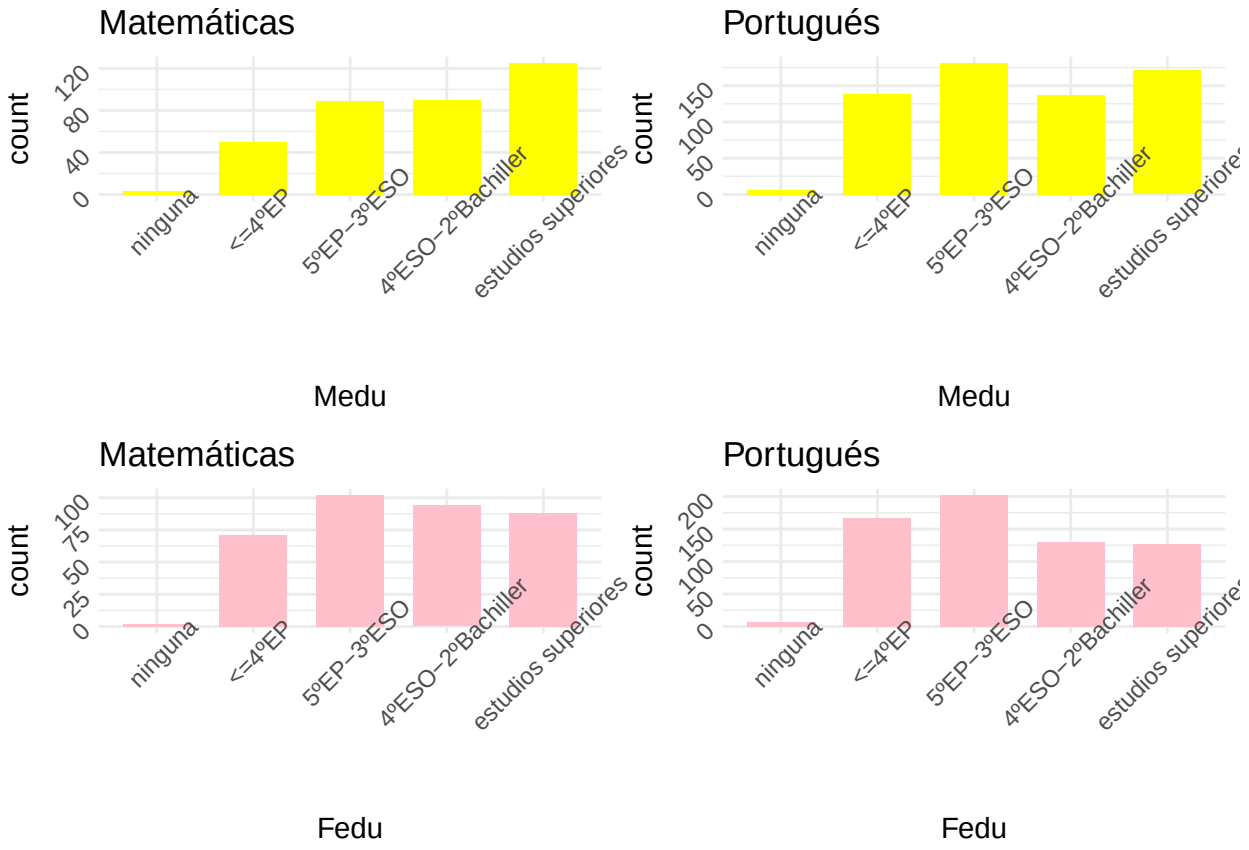
```
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Matemáticas") +
  theme_minimal()
g30 = ggplot(notas_p, aes(x=schoolsup)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Portugués") +
  theme_minimal()
g31 = ggplot(notas_m, aes(x=famsup)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Matemáticas") +
  theme_minimal()
g32 = ggplot(notas_p, aes(x=famsup)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Portugués") +
  theme_minimal()
grid.arrange(g1, g2, g3, g4, g5, g6, g7, g8, g29, g30, g31, g32, nrow = 3, ncol=4)
```



Destacar como hay notablemente más datos para el colegio Gabriel Pereira que para Mousinho da Silveira, hay ligermante más mujeres que hombre y como la mayoría de alumnos viven en un núcleo rural en vez de urbano. En cuánto al apoyo adicional, contrasta la diferencia entre el apoyo proporcionado por el colegio y por las familias. Mientras la mayoría de familias proporcionan apoyo a sus hijos el colegio no proporciona apoyo a casi ningún alumno.

```
g9 = ggplot(notas_m, aes(x=Medu)) +
  geom_bar(stat="count", width=0.7, fill="yellow") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g10 = ggplot(notas_p, aes(x=Medu)) +
  geom_bar(stat="count", width=0.7, fill="yellow") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g11 = ggplot(notas_m, aes(x=Fedu)) +
```

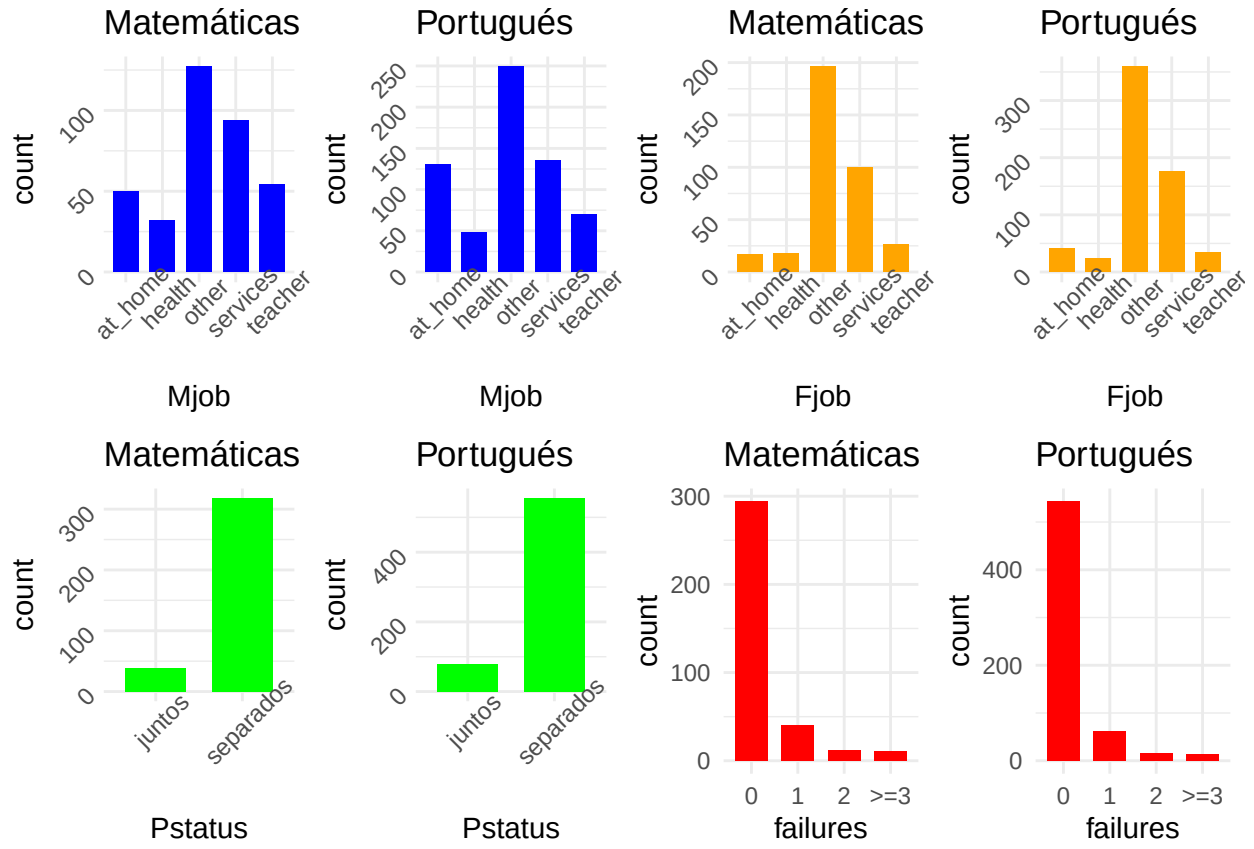
```
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g12 = ggplot(notas_p, aes(x=Fedu)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
grid.arrange(g9, g10, g11,g12, nrow = 2, ncol=2)
```



Casi todos los padres y madres de los alumnos tienen algún tipo de educación.

```
g13 = ggplot(notas_m, aes(x=Mjob)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g14 = ggplot(notas_p, aes(x=Mjob)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g15 = ggplot(notas_m, aes(x=Fjob)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g16 = ggplot(notas_p, aes(x=Fjob)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g17 = ggplot(notas_m, aes(x=Pstatus)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g18 = ggplot(notas_p, aes(x=Pstatus)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
```

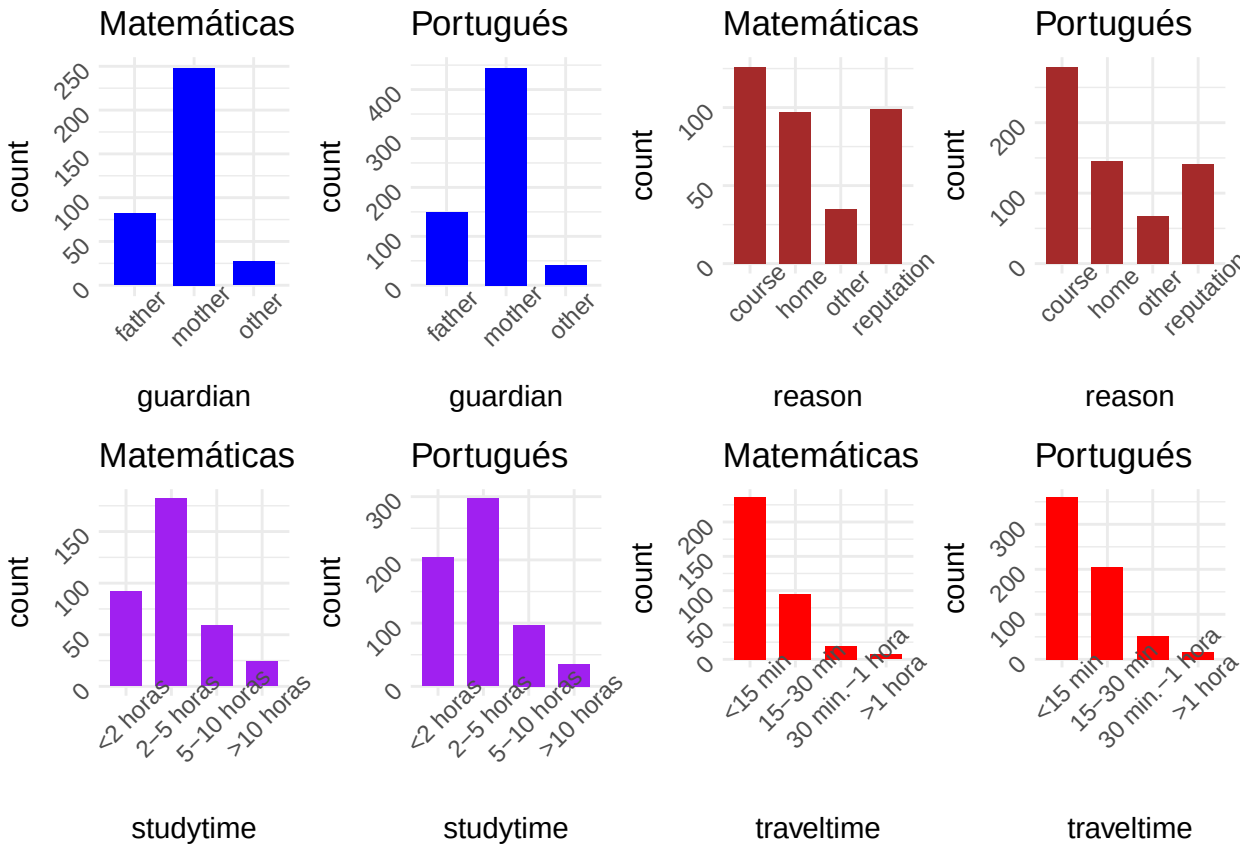
```
g27 = ggplot(notas_m, aes(x=failures)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Matemáticas") +
  theme_minimal()
g28 = ggplot(notas_p, aes(x=failures)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Portugués") +
  theme_minimal()
grid.arrange(g13, g14, g15, g16, g17, g18, g27, g28, nrow = 2, ncol=4)
```



De entre los trabajos propuestos sin contar otro tipo de trabajo, el más populares entre los padres es el de servicios. Para las madres también es el de servicios pero también destacan el de profesora y ama de casa. Mencionar como la gran mayoría de padres y madres de los alumnos además viven separados.

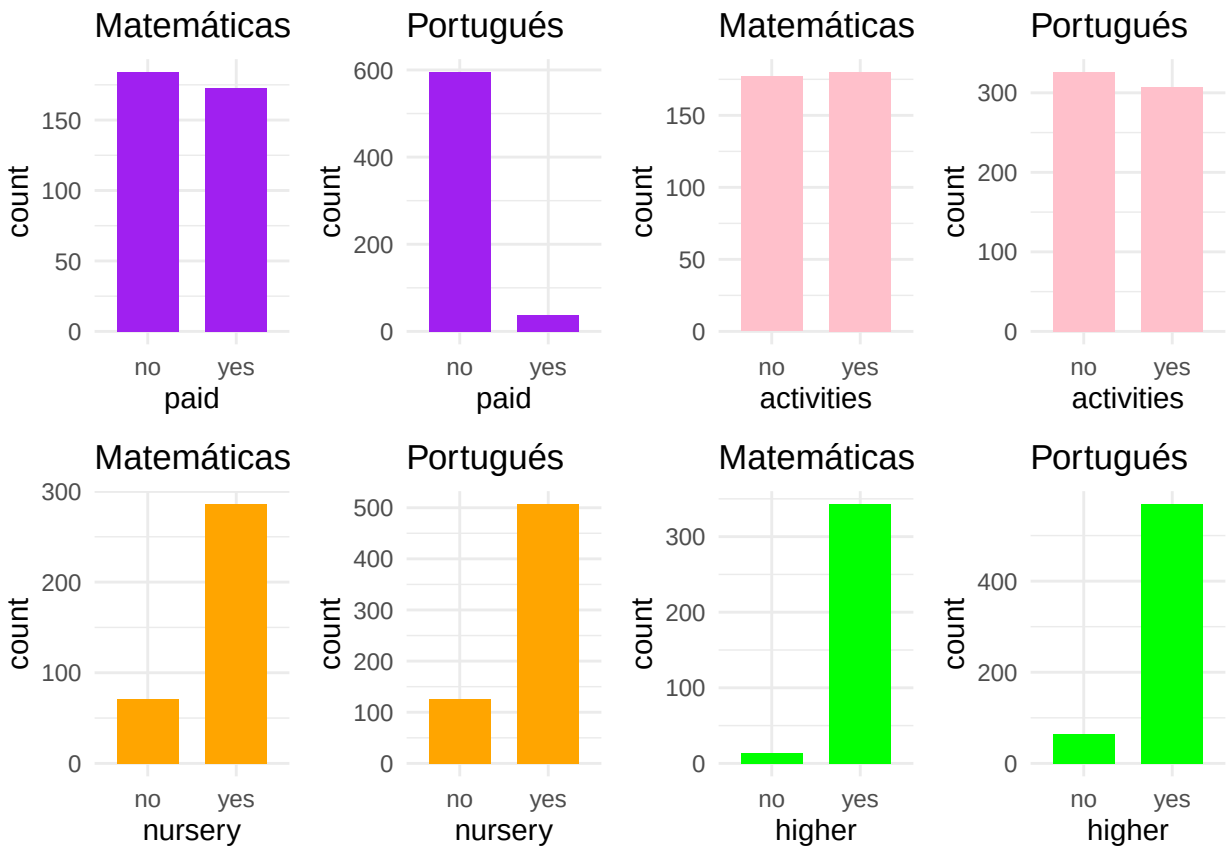
```
g19 = ggplot(notas_m, aes(x=reason)) +
  geom_bar(stat="count", width=0.7, fill="brown") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g20 = ggplot(notas_p, aes(x=reason)) +
  geom_bar(stat="count", width=0.7, fill="brown") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g21 = ggplot(notas_m, aes(x=guardian)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g22 = ggplot(notas_p, aes(x=guardian)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g23 = ggplot(notas_m, aes(x=traveltime)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
```

```
g24 = ggplot(notas_p, aes(x=traveltime)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g25 = ggplot(notas_m, aes(x=studytime)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g26 = ggplot(notas_p, aes(x=studytime)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
grid.arrange(g21, g22, g19, g20, g25, g26, g23, g24, nrow = 2, ncol=4)
```




```
g33 = ggplot(notas_m, aes(x=paid)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Matemáticas") +
  theme_minimal()
g34 = ggplot(notas_p, aes(x=paid)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Portugués") +
  theme_minimal()
g35 = ggplot(notas_m, aes(x=activities)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Matemáticas") +
  theme_minimal()
g36 = ggplot(notas_p, aes(x=activities)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Portugués") +
  theme_minimal()
g37 = ggplot(notas_m, aes(x=nursery)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Matemáticas") +
  theme_minimal()
```

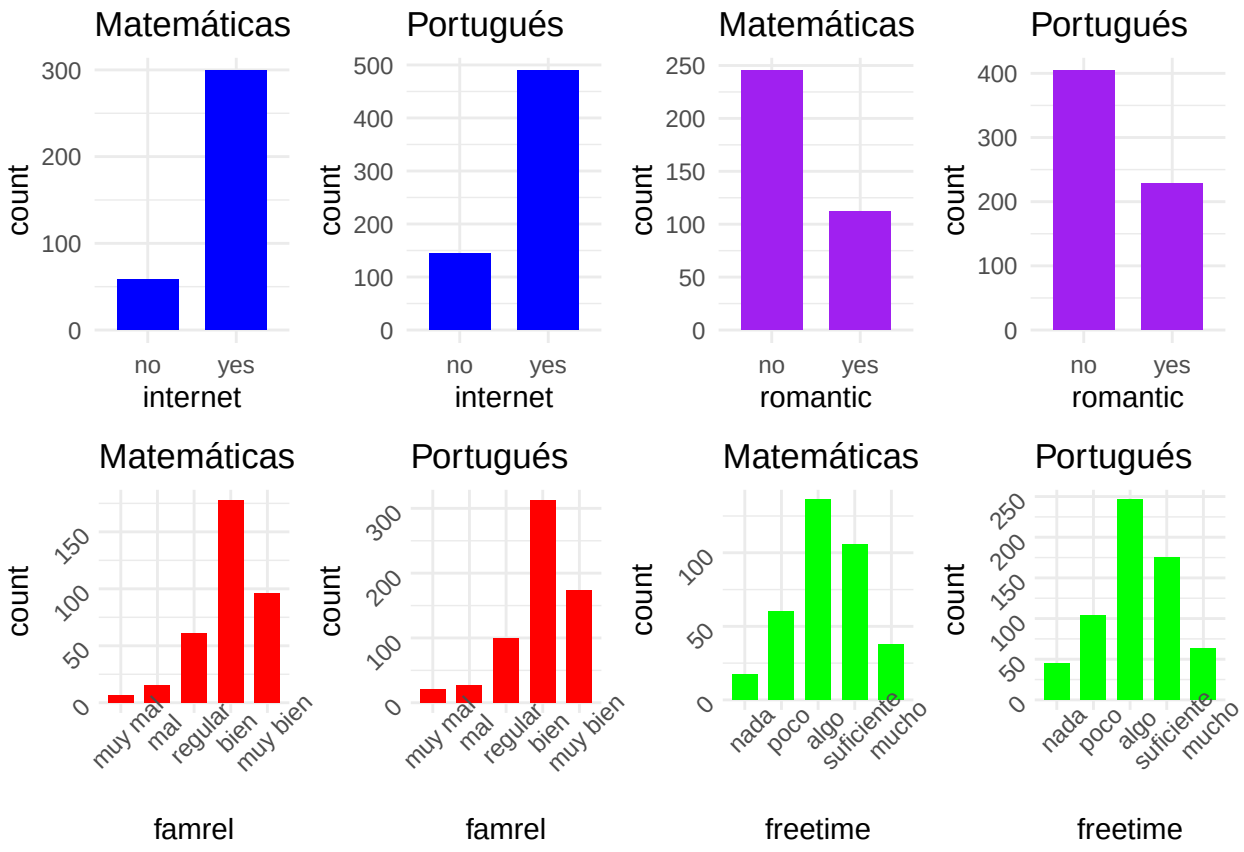
```
g38 = ggplot(notas_p, aes(x=nursery)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Portugués") +
  theme_minimal()
g39 = ggplot(notas_m, aes(x=higher)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Matemáticas") +
  theme_minimal()
g40 = ggplot(notas_p, aes(x=higher)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Portugués") +
  theme_minimal()
grid.arrange(g33, g34, g35, g36, g37, g38, g39, g40, nrow = 2, ncol=4)
```



Existe una notable diferencia entre las asignaturas en cuanto a las clases extras pagadas. La asignatura de matemáticas tiene una notablemente mayor proporción de alumnos que pagan clases que la asignatura de portugués.

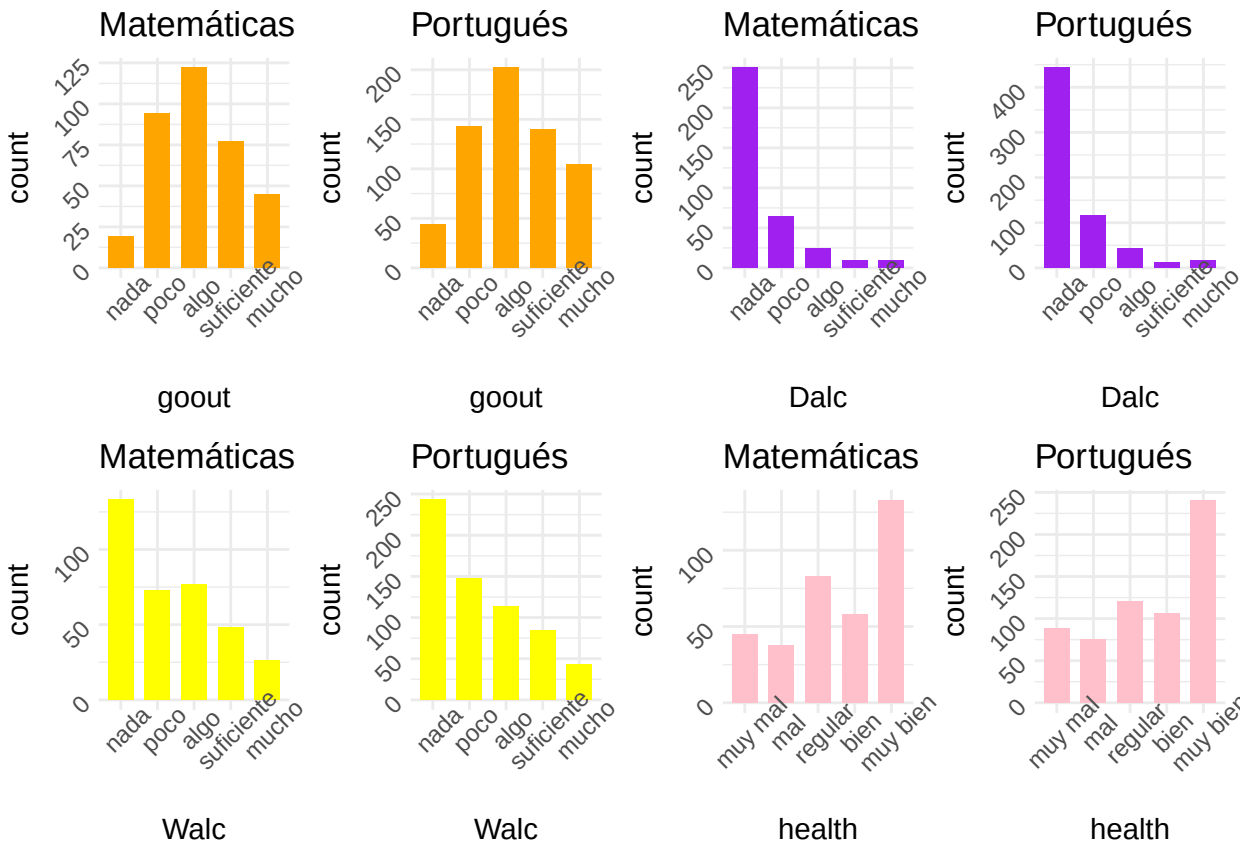
```
g41 = ggplot(notas_m, aes(x=internet)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Matemáticas") +
  theme_minimal()
g42 = ggplot(notas_p, aes(x=internet)) +
  geom_bar(stat="count", width=0.7, fill="blue") + labs(title="Portugués") +
  theme_minimal()
g43 = ggplot(notas_m, aes(x=romantic)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Matemáticas") +
  theme_minimal()
g44 = ggplot(notas_p, aes(x=romantic)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Portugués") +
  theme_minimal()
```

```
g45 = ggplot(notas_m, aes(x=famrel)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g46 = ggplot(notas_p, aes(x=famrel)) +
  geom_bar(stat="count", width=0.7, fill="red") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g47 = ggplot(notas_m, aes(x=freetime)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g48 = ggplot(notas_p, aes(x=freetime)) +
  geom_bar(stat="count", width=0.7, fill="green") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
grid.arrange(g41, g42, g43, g44, g45, g46, g47, g48, nrow = 2, ncol=4)
```




```
g49 = ggplot(notas_m, aes(x=goout)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g50 = ggplot(notas_p, aes(x=goout)) +
  geom_bar(stat="count", width=0.7, fill="orange") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g51 = ggplot(notas_m, aes(x=Dalc)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g52 = ggplot(notas_p, aes(x=Dalc)) +
  geom_bar(stat="count", width=0.7, fill="purple") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
```

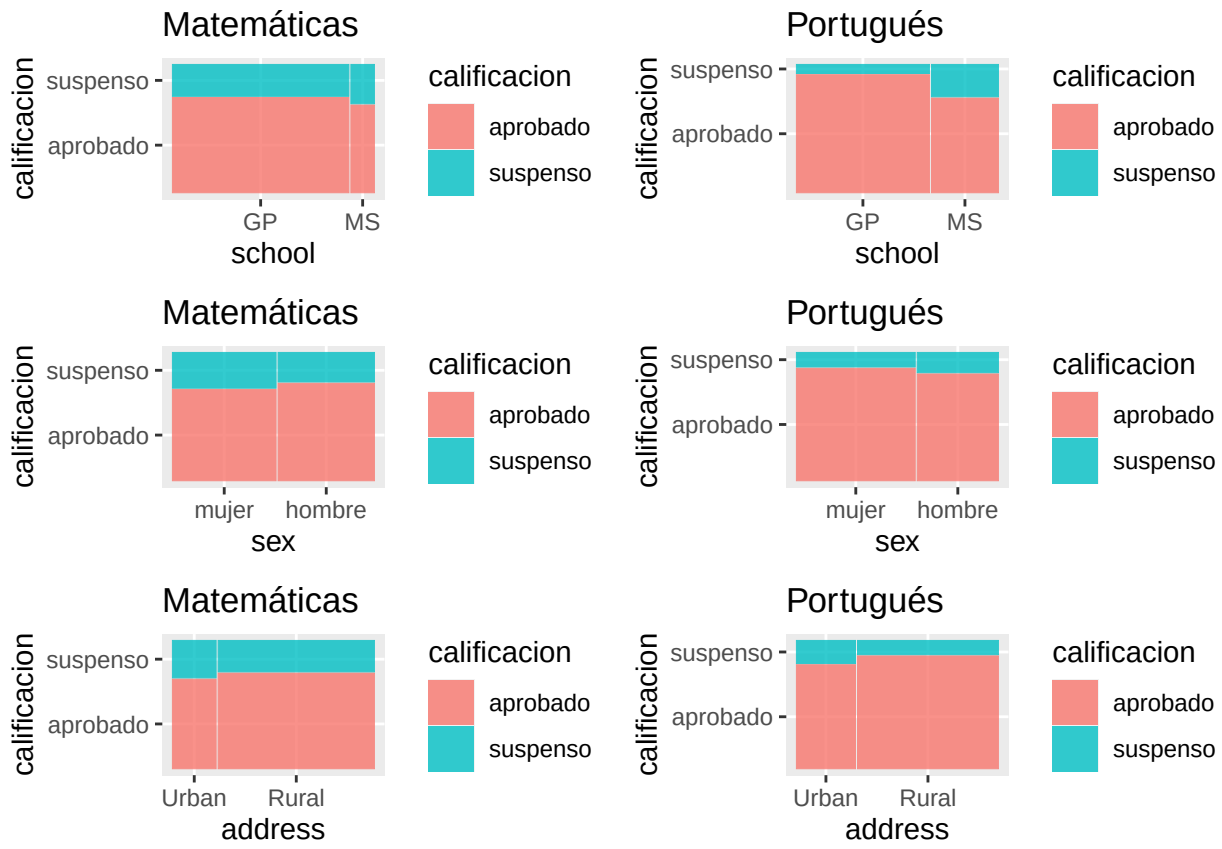
```
g53 = ggplot(notas_m, aes(x=Walc)) +
  geom_bar(stat="count", width=0.7, fill="yellow") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g54 = ggplot(notas_p, aes(x=Walc)) +
  geom_bar(stat="count", width=0.7, fill="yellow") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g55 = ggplot(notas_m, aes(x=health)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Matemáticas") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
g56 = ggplot(notas_p, aes(x=health)) +
  geom_bar(stat="count", width=0.7, fill="pink") + labs(title="Portugués") +
  theme_minimal() + theme(axis.text = element_text(angle = 45))
grid.arrange(g49, g50, g51, g52, g53, g54, g55, g56, nrow = 2, ncol=4)
```



Para finalizar con las varibles nominales, se analiza visualmente su relación con la variable calificación.

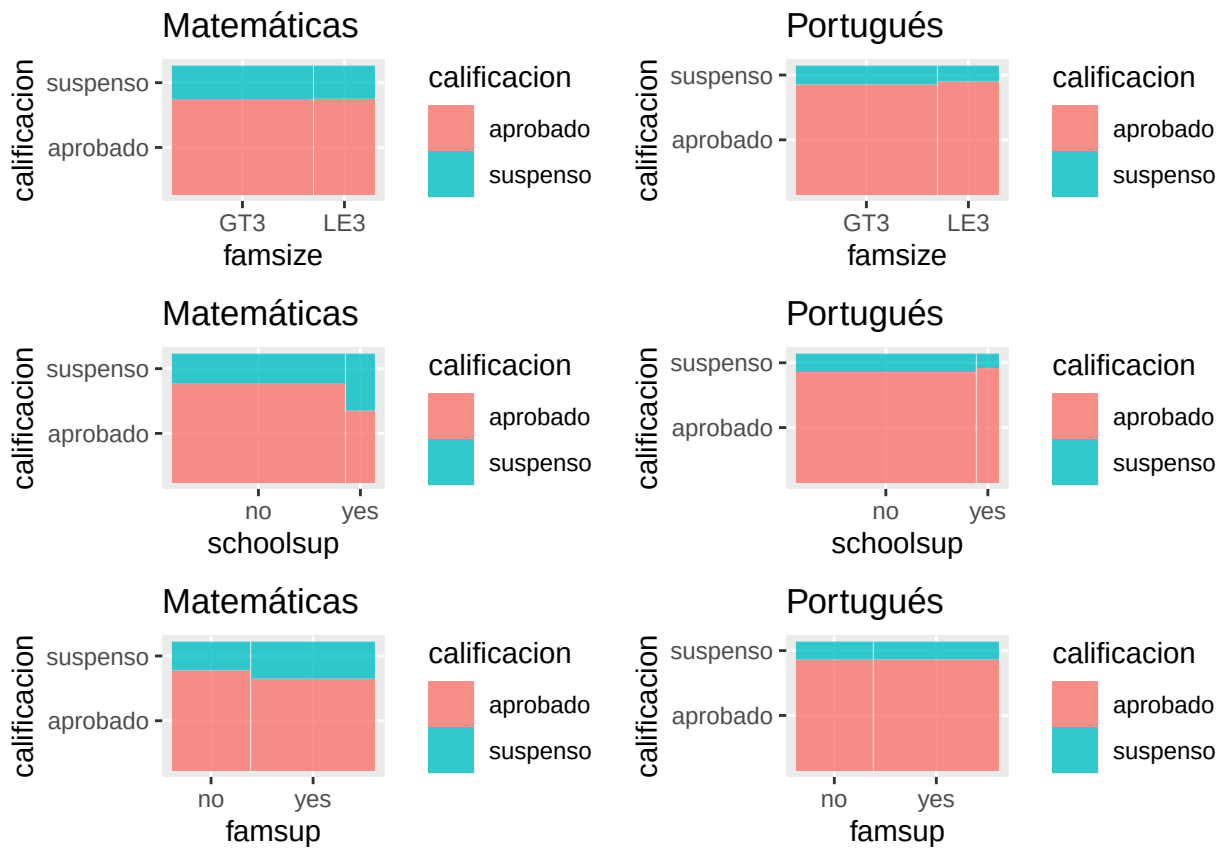
```
q1=ggplot(data = notas_m) +
  geom_mosaic(aes(x = product(calificacion, school), fill=calificacion)) + labs(title='Matemáticas')
q2=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, school), fill=calificacion)) +
  labs(title='Portugués')
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, sex), fill=calificacion))  + labs
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, sex), fill=calificacion)) +
  labs(title='Portugués')
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, address), fill=calificacion)) +
  labs(title='Matemáticas')
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, address),
```

```
                                        fill=calificacion)) +
  labs(title='Portugués')
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



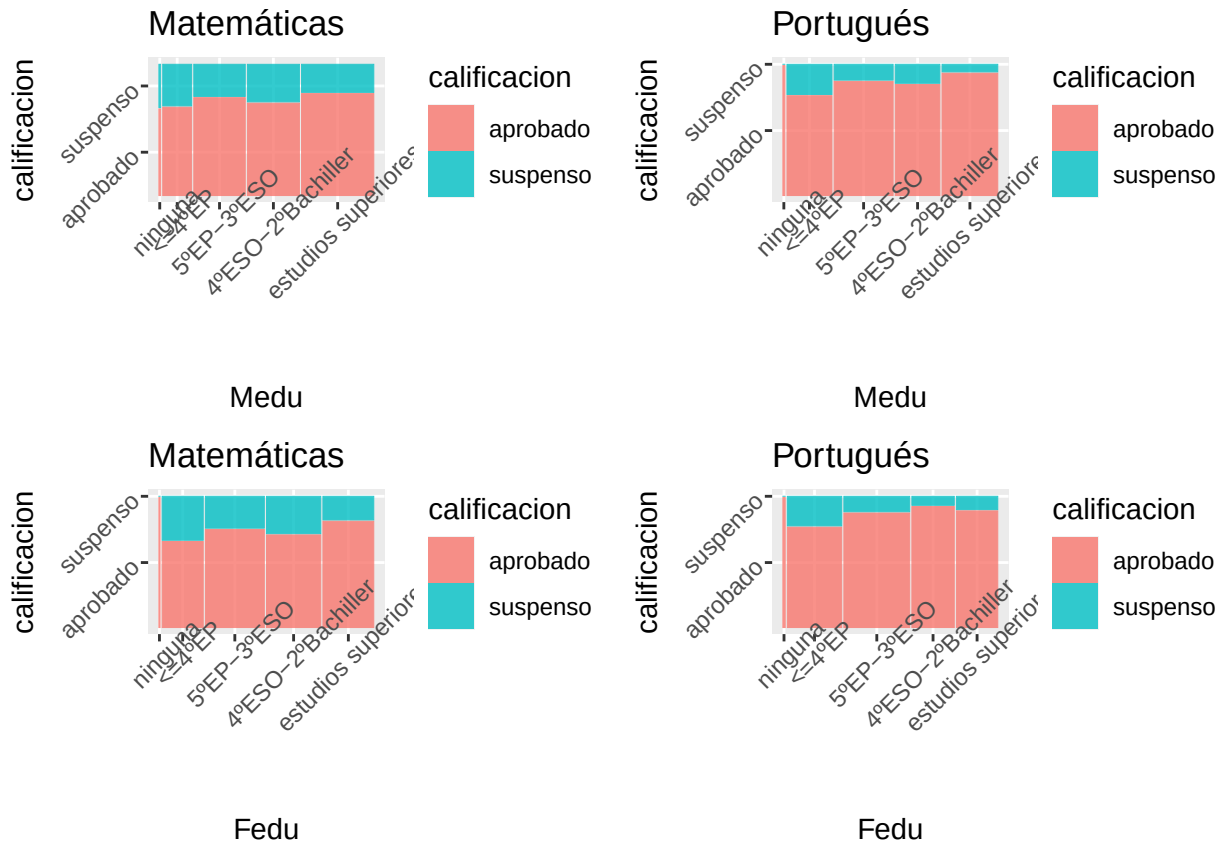
En ambas asignaturas, el porcentaje de suspensos y aprobados no tiene diferencias notables en las distintas categorías dentro de una variable. La única pequeña diferencia que se podría destacar es que el colegio Mousinho da Silveira tiene menos aprobados que el colegio Gabriel Pereira.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, famsize),
                                        fill=calificacion)) +
  labs(title='Matemáticas')
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, famsize), fill=calificacion)) + labs(title='Portugués')
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, schoolsup), fill=calificacion)) +
  labs(title='Matemáticas')
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, schoolsup), fill=calificacion)) +
  labs(title='Portugués')
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, famsup), fill=calificacion)) +
  labs(title='Matemáticas')
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, famsup),
                                        fill=calificacion)) +
  labs(title='Portugués')
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



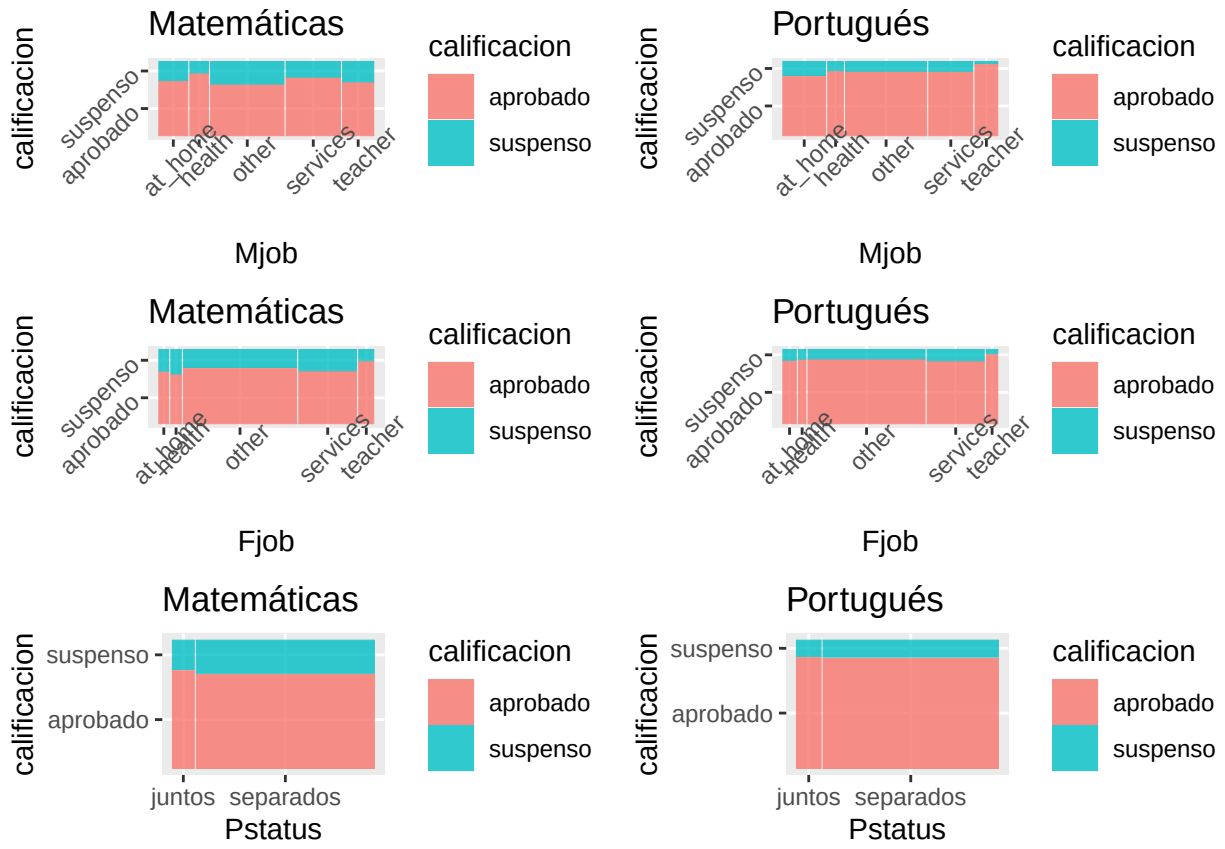
Al igual que en los gráficos anteriores, el porcentaje de suspensos y aprobados no tiene diferencias notables en las distintas categorías dentro de una variable. Una pequeña diferencia se observa entre los distintos niveles de la variable schoolsup en la asignatura de matemáticas: aquellos alumnos que no tienen apoyo del colegio tienen mayor proporción de aprobados. Esto a lo mejor se debe a que aquellos alumnos que si reciben apoyo del colegio son aquellos que más lo necesitan y que peor llevan la asignatura.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Medu),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, Medu), fill=calificacion)) + labs(title='Portugués') + theme
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Fedu), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, Fedu), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
plot_grid(q1,q2,q3,q4,nrow = 2)
```



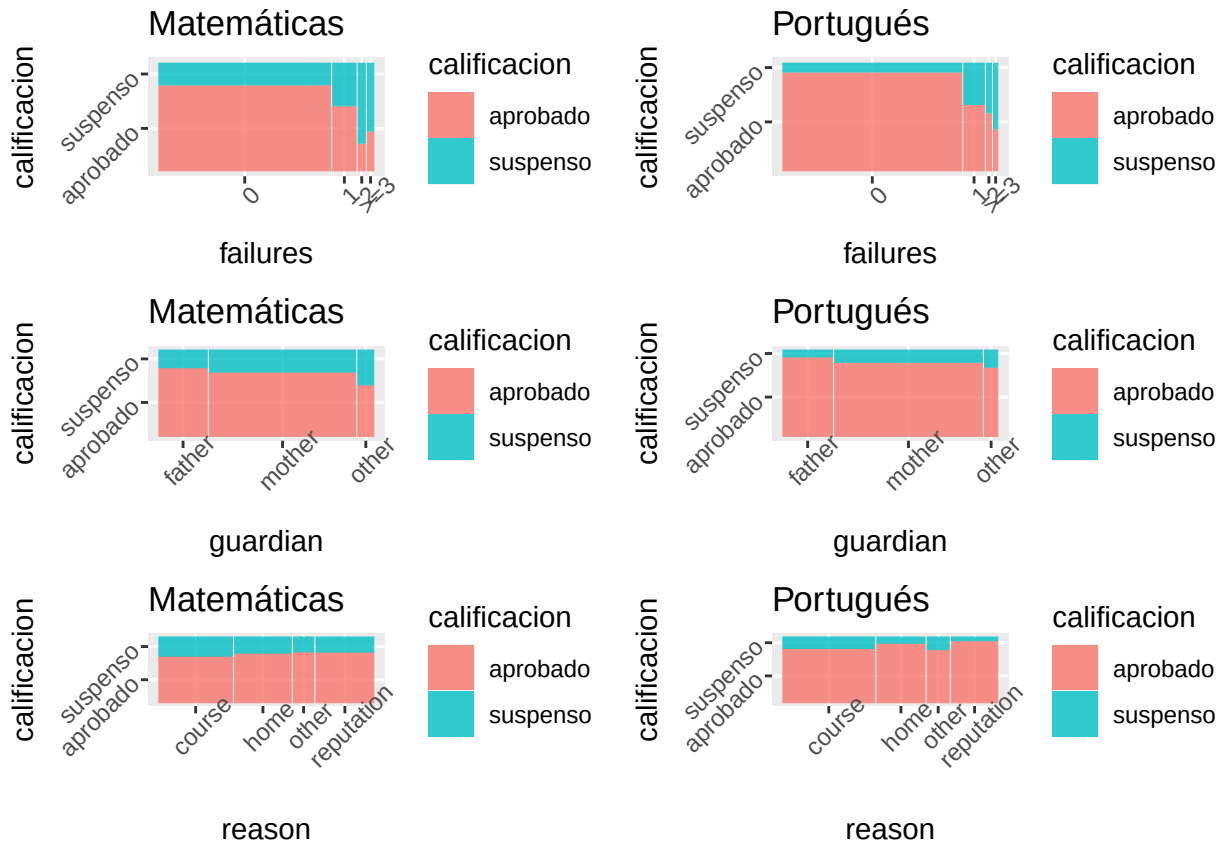
Las variables se comportan igual en ambas asignaturas al igual que en los gráficos anteriores. Sin embargo, entre los distintos niveles de estas variables se encuentran ligeras diferencias. Aquellos alumnos cuyos padres no tienen estudio han aprobado todos y para el resto de niveles la proporción de aprobados aumenta según aumenta la educación de los padres y madres.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Mjob),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, Mjob), fill=calificacion)) + labs(title='Portugués') + theme
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Fjob), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, Fjob), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Pstatus), fill=calificacion)) +
  labs(title='Matemáticas')
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, Pstatus),
                                            fill=calificacion)) +
  labs(title='Portugués')
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



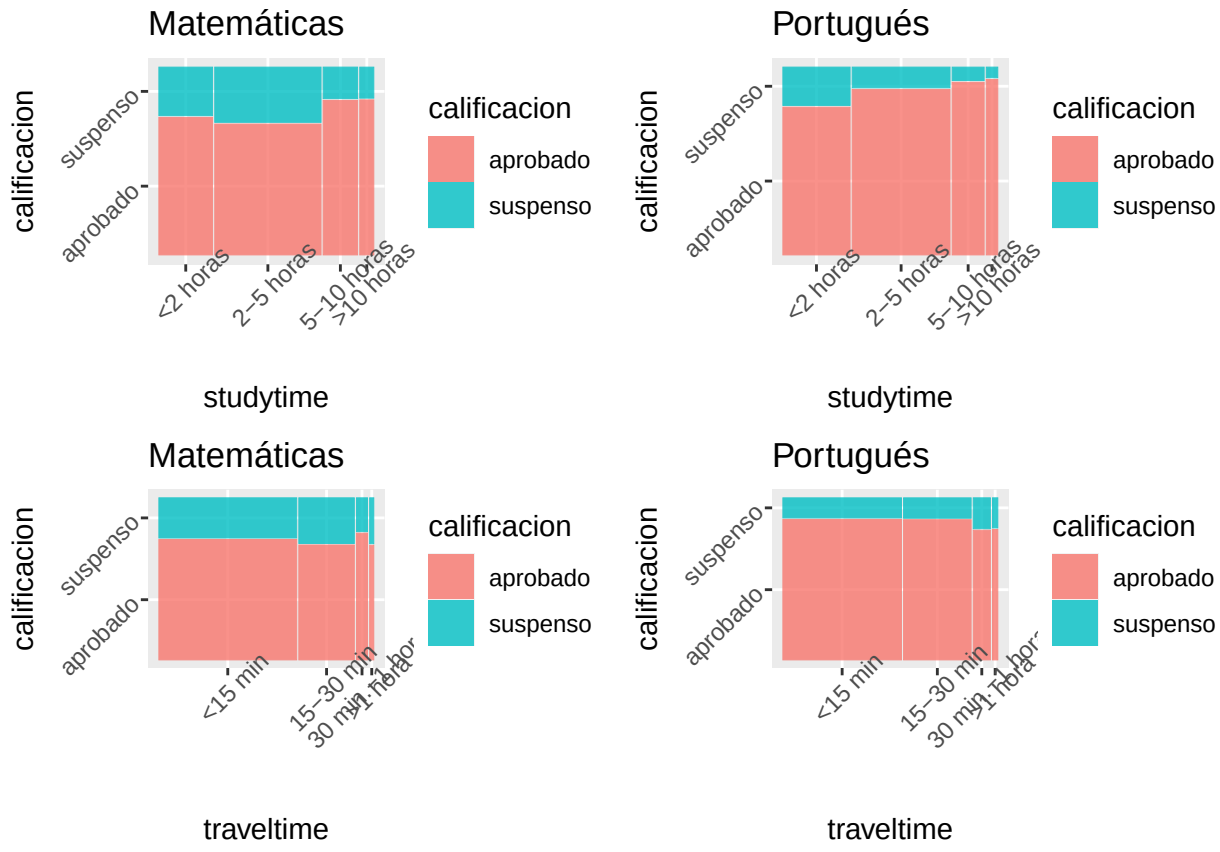
Estas variables también se comportan igual en ambas asignaturas. La única pequeña diferencia que se podría destacar de estas variables es que aquellos alumnos cuyo padre o madre son profesores tienen mayor porcentaje de aprobados que los otros niveles de la misma variable.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, failures),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, failures), fill=calificacion)) + labs(title='Portugués') + t
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, guardian), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, guardian), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, reason), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, reason),
                                            fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



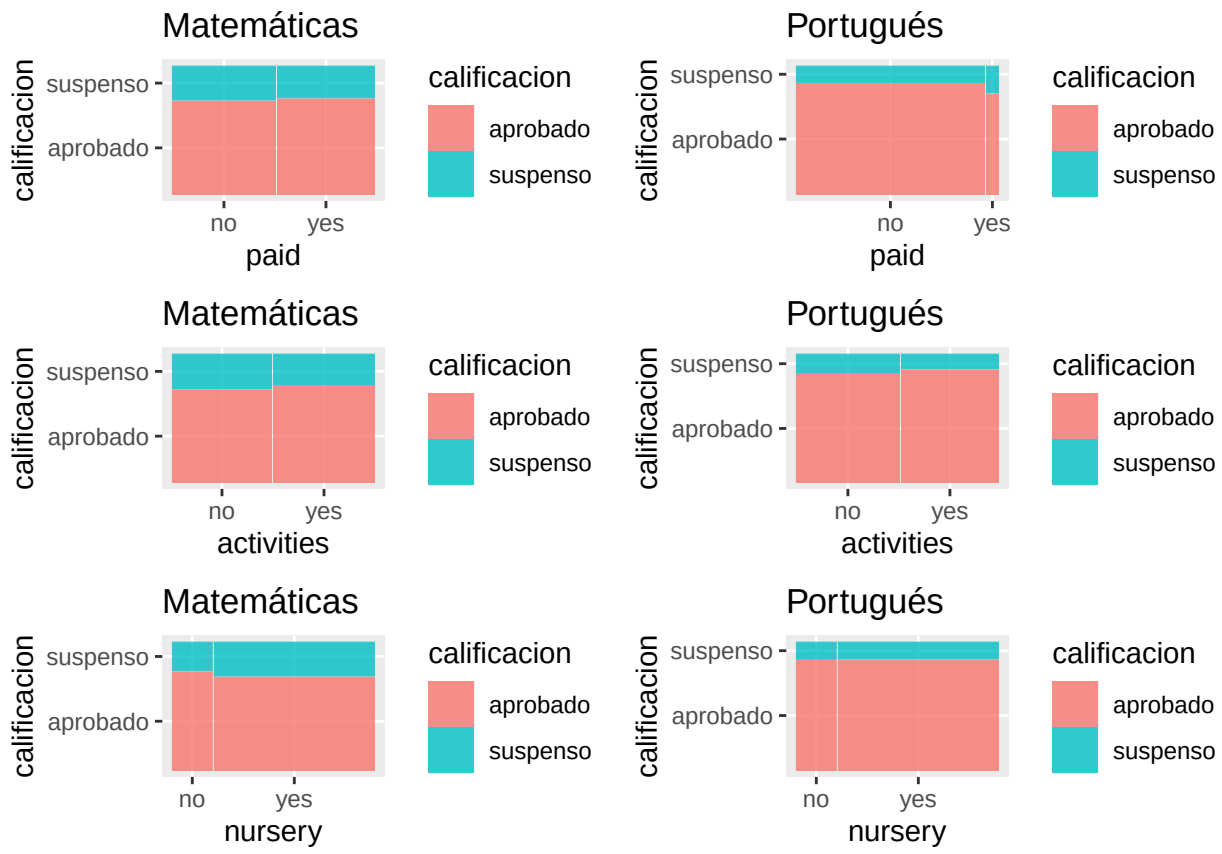
Aquellos alumnos con asignaturas pasadas suspensas tienen un mayor porcentaje de suspensos en las asignaturas estudiadas. Mencionar también que aquellos alumnos cuyo tutor legal no es ni su padre ni su madre tienen un ligero mayor porcentaje de suspensos que las otras categoría y que aquellos alumnos que eligieron el colegio por su reputación tienen el porcentaje de aprobados más alto de entre las otras categorías de la variables.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, studytime),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, studytime), fill=calificacion)) + labs(title='Portugués') +
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, traveltime), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, traveltime), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
plot_grid(q1,q2,q3,q4, nrow = 2)
```



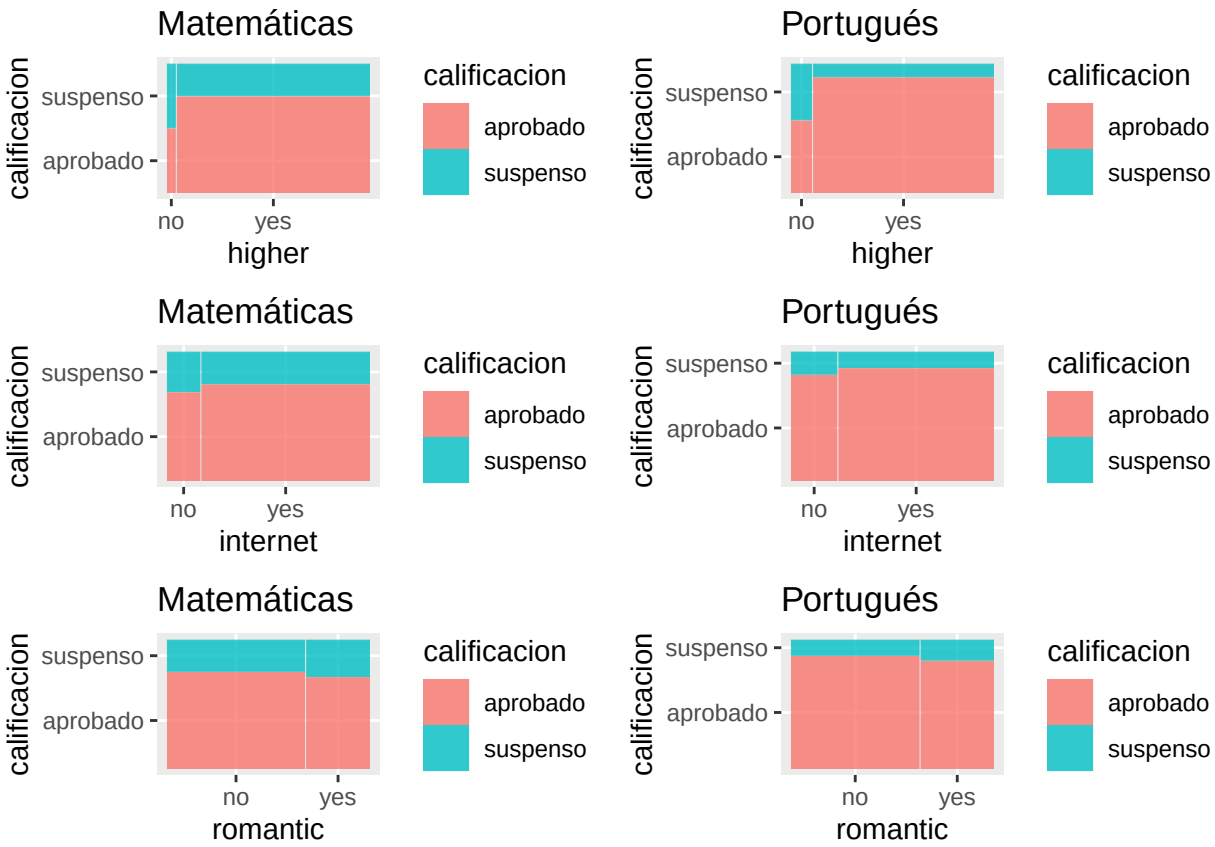
Como es evidente, la proporción de aprobados aumenta cuanto mayor es el tiempo de estudio dedicado a la asignatura pero tampoco en gran proporción. En cuanto a la variable de tiempo entre el colegio y la casa del estudiente no se destaca ninguna diferencia entre los distintos niveles.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, paid),
                                            fill=calificacion)) +
  labs(title='Matemáticas')
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, paid), fill=calificacion)) + labs(title='Portugués')
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, activities), fill=calificacion)) +
  labs(title='Matemáticas')
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, activities), fill=calificacion)) +
  labs(title='Portugués')
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, nursery), fill=calificacion)) +
  labs(title='Matemáticas')
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, nursery),
                                            fill=calificacion)) +
  labs(title='Portugués')
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



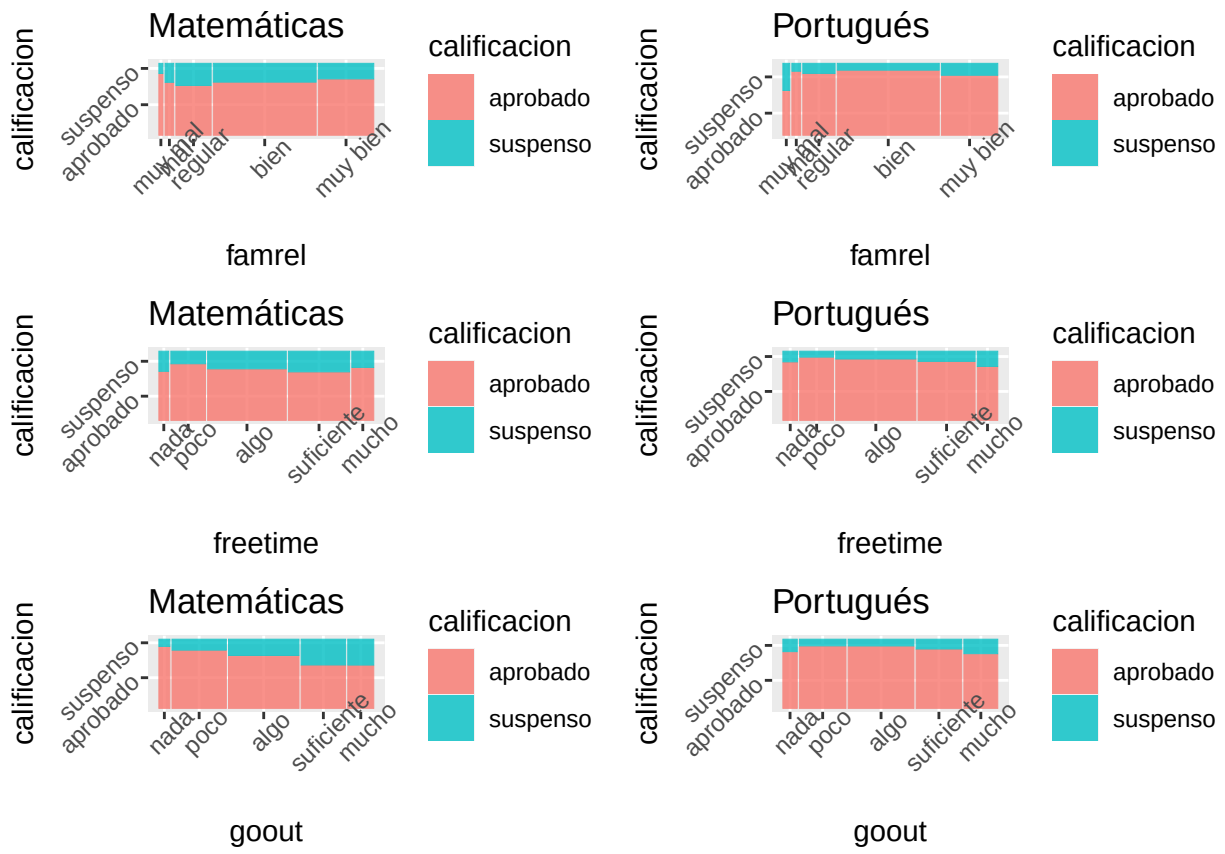
En estas variables no se observa ninguna diferencia notable.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, higher),
                                            fill=calificacion)) +
  labs(title='Matemáticas')
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, higher), fill=calificacion)) + labs(title='Portugués')
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, internet), fill=calificacion)) +
  labs(title='Matemáticas')
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, internet), fill=calificacion)) +
  labs(title='Portugués')
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, romantic), fill=calificacion)) +
  labs(title='Matemáticas')
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, romantic),
                                            fill=calificacion)) +
  labs(title='Portugués')
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



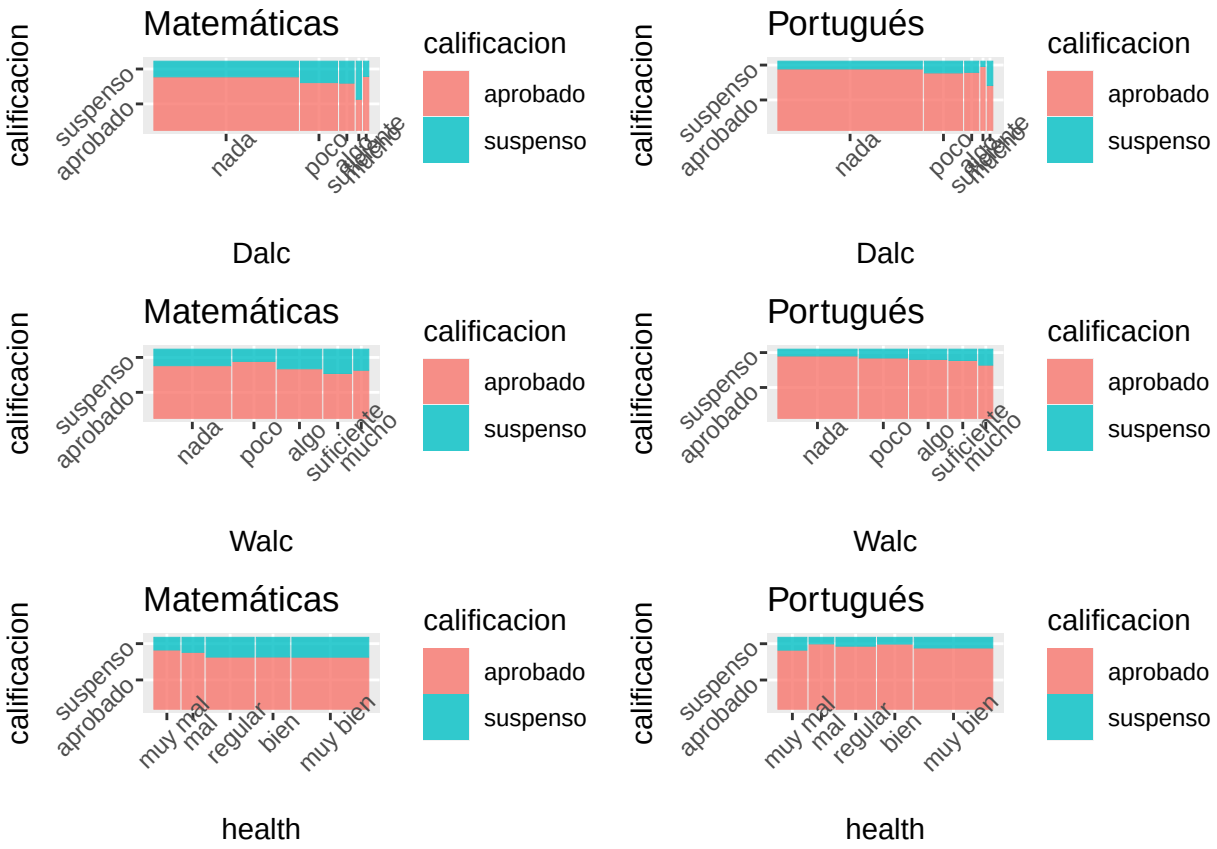
Se observa una clara diferencia entre los distintos niveles de la variable higher en ambas asignaturas: aquellos alumnos que desean continuar con su educación al terminar el instituto tienen una mayor proporción de aprobados que aquellos que no. En los niveles de las otras dos variables se nota una pequeña diferencia pero tampoco de gran importancia. Los alumnos que no están en una relación romántica presenta un ligero mayor porcentaje de aprobados, esto podría a lo mejor deberse a que al no tener que dedicarle tiempo a una relación tienen más tiempo disponible y no están tan distraidos. Y sobre la variable internet, los alumnos que tienen acceso a internet también presentan una ligera mayor proporción de aprobados que los que no. Esto podría deberse a que estos alumno con internet tienen más recursos para estudiar y así sacar mejores notas y aprobar.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, famrel),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, famrel), fill=calificacion)) + labs(title='Portugués') + th
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, freetime), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, freetime), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, goout), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, goout),
                                            fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



En la asignatura de matemáticas los alumnos que muy mala relación familiar son los que mayor proporción de aprobados tienen al contrario que en la asigntura de portugués. Para el resto de niveles de famrel, a medida que mejora la relación familiar mejor ligeramente la proporción de aprobados pero no de forma muy notable. Mencionar también que en amba asignaturas para la variable free time los alumnos que tienen poco tiempo libre aprueban más que los otros niveles. Y de forma similar a la variable freetime, la variable goout que indica cuanto salen los alumnos muestra como, en ambas asignaturas, el porcentaje de aprobados aumenta ligeramente cuanto menos sales a excepción de en el nivel nada.

```
q1=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Dalc),
                                            fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q2=ggplot(data = notas_p) +
  geom_mosaic(aes(x = product(calificacion, Dalc), fill=calificacion)) + labs(title='Portugués') + theme
q3=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, Walc), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q4=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, Walc), fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
q5=ggplot(data = notas_m) + geom_mosaic(aes(x = product(calificacion, health), fill=calificacion)) +
  labs(title='Matemáticas') + theme(axis.text = element_text(angle = 45))
q6=ggplot(data = notas_p) + geom_mosaic(aes(x = product(calificacion, health),
                                            fill=calificacion)) +
  labs(title='Portugués') + theme(axis.text = element_text(angle = 45))
plot_grid(q1,q2,q3,q4,q5,q6, nrow = 3)
```



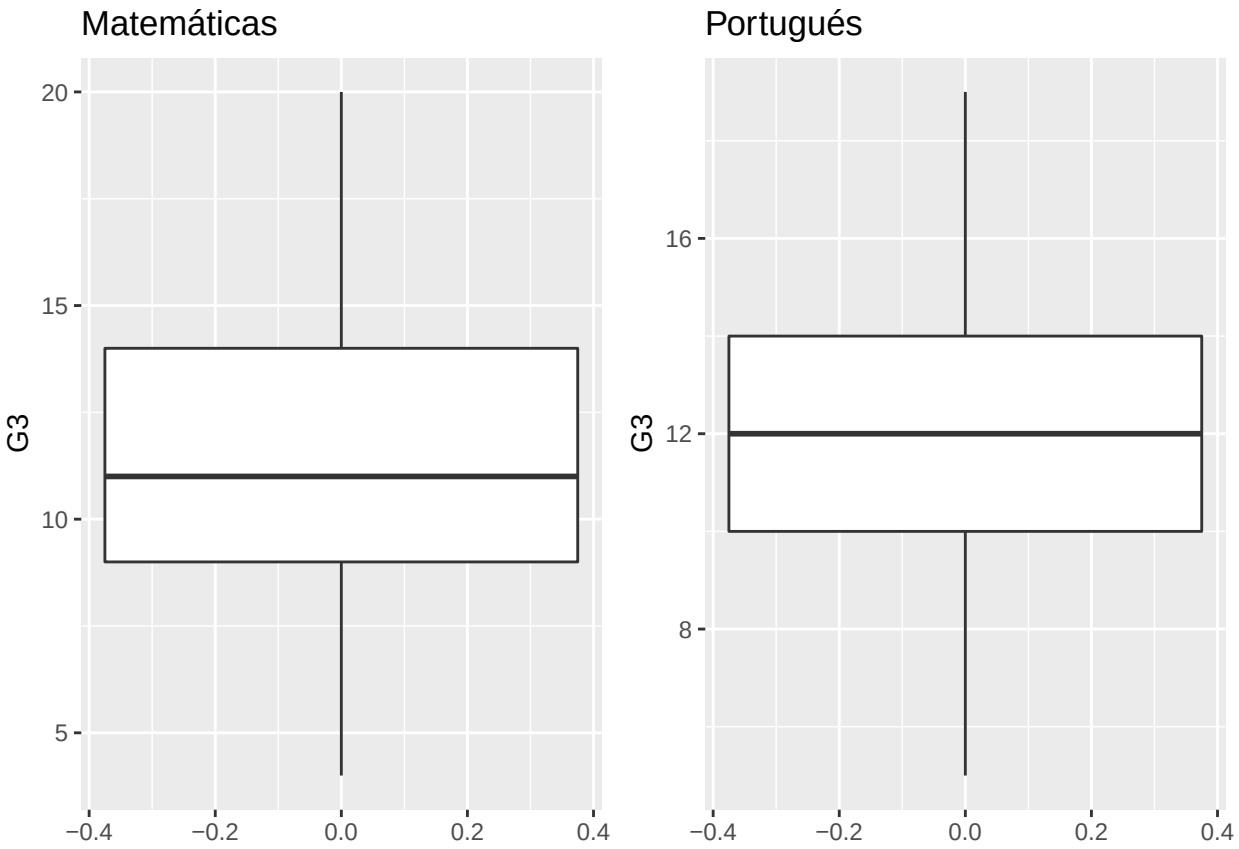
Por último, el porcentaje de aprobados en ambas variables dismuye a medida que aumenta el alcohol consumido. Siendo la excepción, la asignatura de matemáticas en la variable Dalc ya que aquellos alumnos que consumen bastante alcohol en un día lectivo son los que sorprendente mayor porcentaje de aprobados tienen de entre el resto de niveles de la variable. En cuánto a al estado de salud, las diferencias son mínimas.

Con esto, se termina el análisis visual de las variables nominales.

A continuación se grafican y estudian las variables númericas. Aquellos estudiantes cuyos padres no tienen educación han aprobado. Para las otros niveles, el porcentaje de aprobados aumenta según aumenta la educación de los padres.

La nota final de la asignatura, la variable G3, se puede representar mediante un diagrama de cajas.

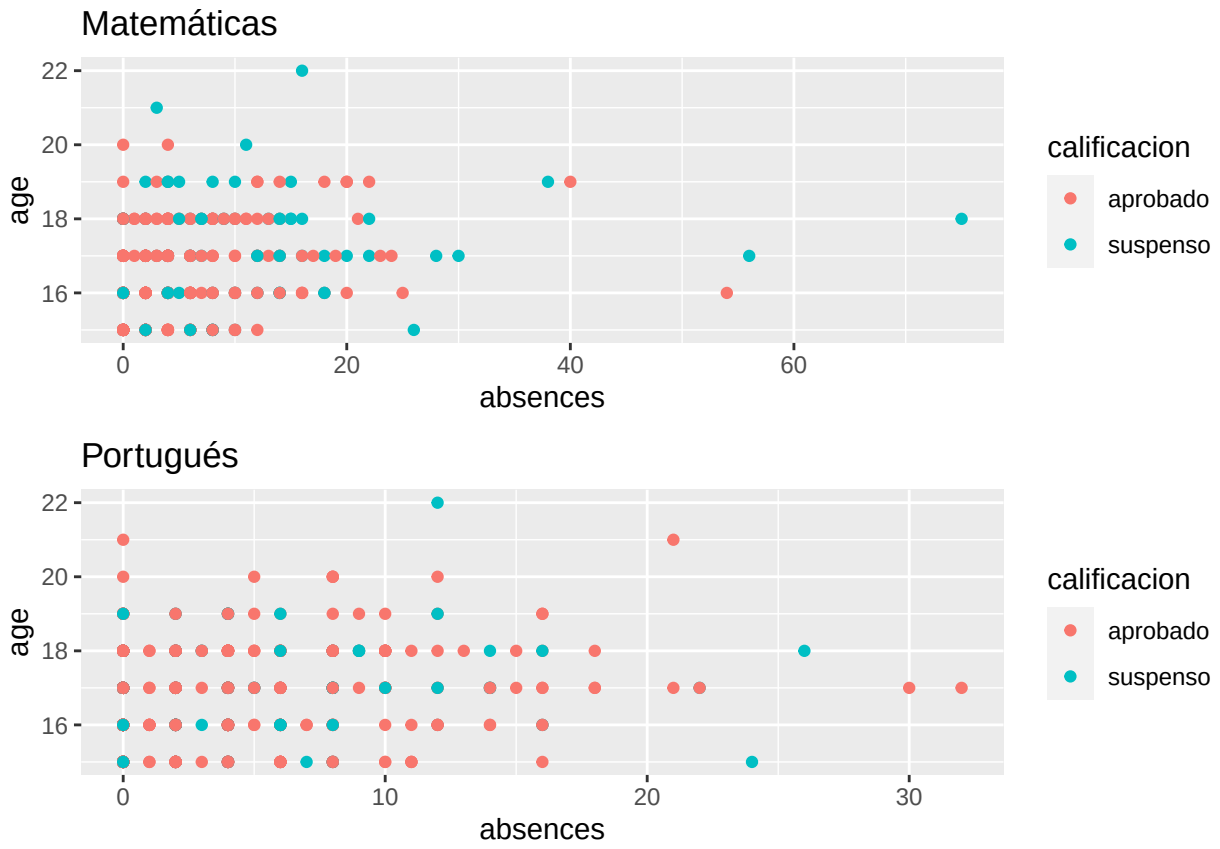
```
p1 <- ggplot(notas_m, aes(y=G3)) +
geom_boxplot() + labs(title="Matemáticas")
p2 <- ggplot(notas_p, aes(y=G3)) +
geom_boxplot() + labs(title="Portugués")
grid.arrange(p1,p2, nrow = 1, ncol=2)
```



A continuación, realizamos, por ejemplo, varios diagramas de dispersión coloreando los resultados de acuerdo con la variable calificación para observar como se comporta esta variable respecto a las variables númericas.

Diagramas de dispersión de la edad y ausencias.

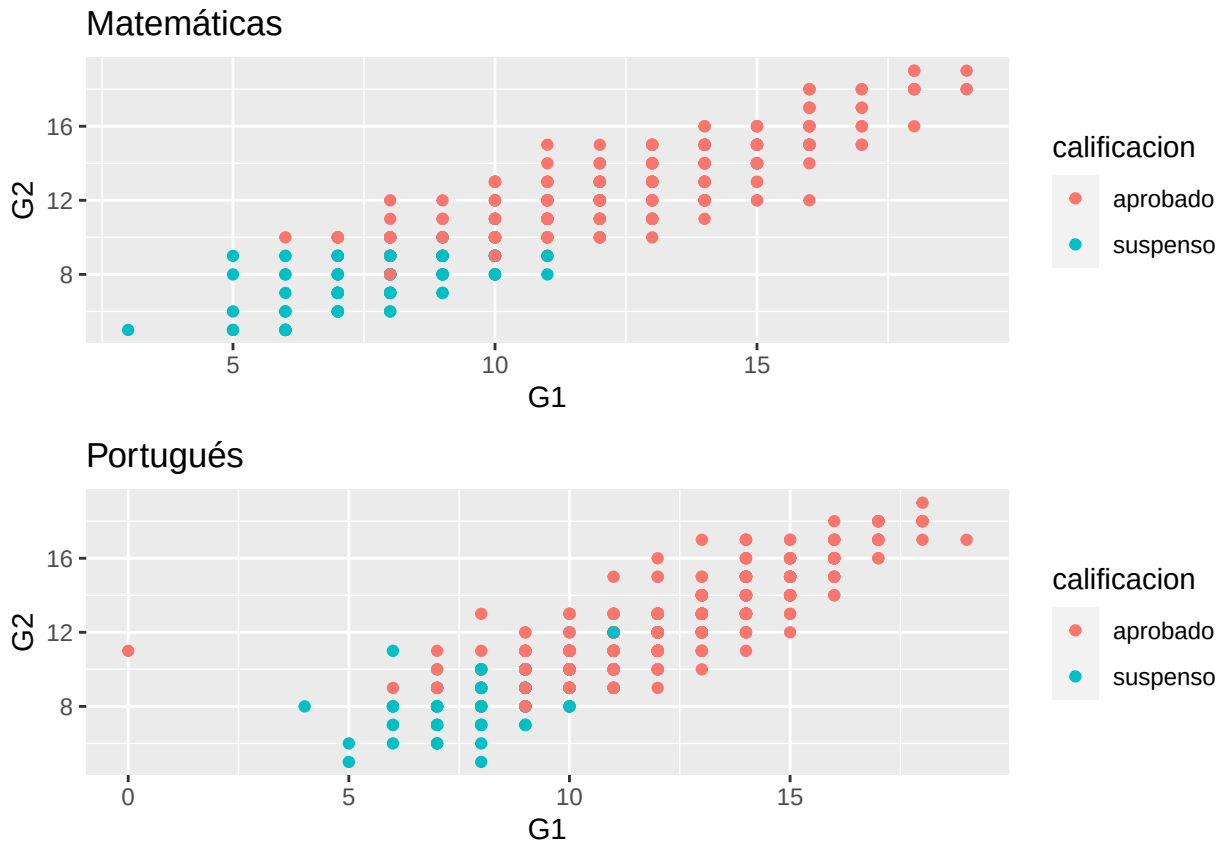
```
q1<-qplot(absences, age, data = notas_m, colour = calificacion) + labs(title="Matemáticas")
q2<-qplot(absences, age, data = notas_p, colour = calificacion) + labs(title="Portugués")
grid.arrange(q1,q2, nrow = 2, ncol=1)
```



No se observa una clara separación entre los supensos y aprobados según los valores de la edad y las ausencias. Mencionar que la mayoría alumnos están entre los 15 y 19 años como es normal en un instituto. Aquellos de mayor edad puede ser repetidores.

Diagramas de dispersión de la nota del primer trimestre y la nota del segundo trimestre

```
q1<-qplot(G1, G2, data = notas_m, colour = calificacion) + labs(title="Matemáticas")
q2<-qplot(G1, G2, data = notas_p, colour = calificacion) + labs(title="Portugués")
grid.arrange(q1,q2, nrow = 2, ncol=1)
```

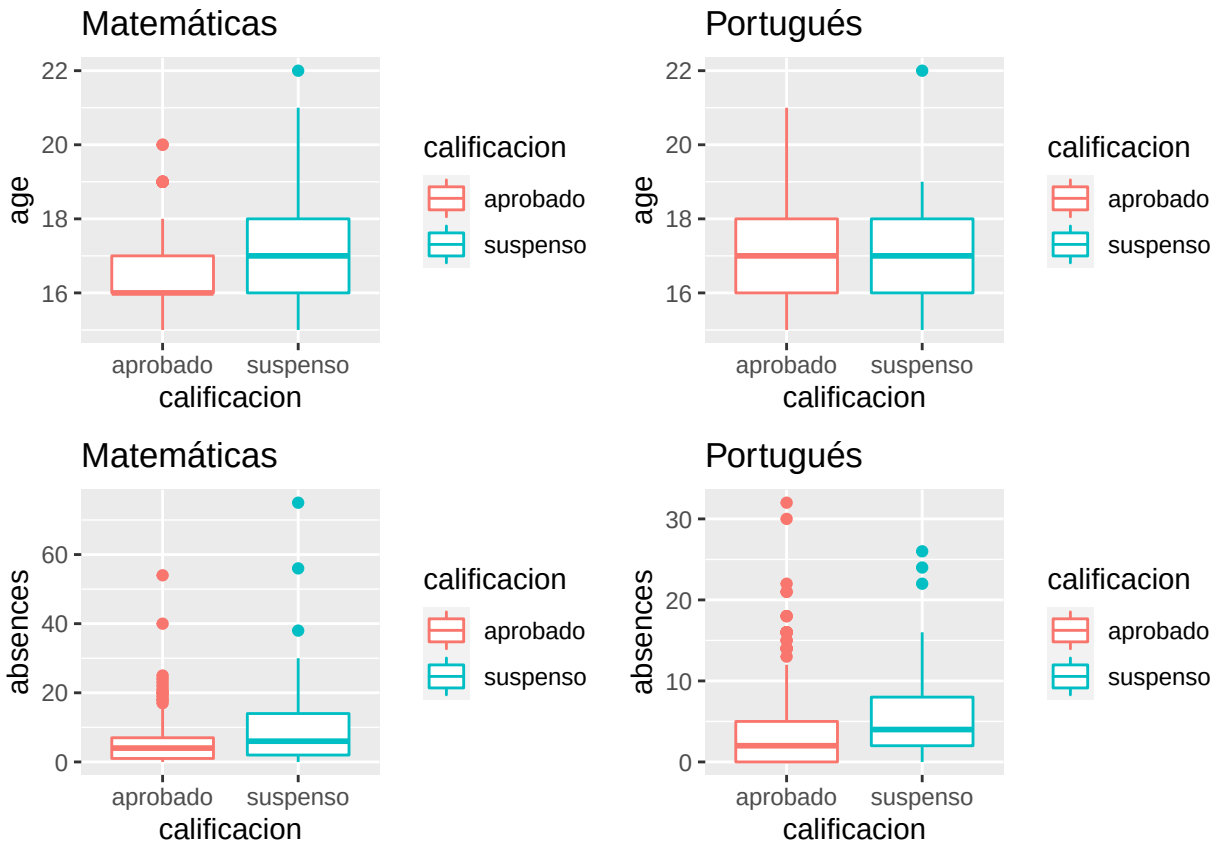


Se observa una clara separación en ambas asignaturas entre los puntos de color azul, que corresponden con los supensos, y los puntos de color naranja, que corresponden con los aprobados. Aquellos que obtuvieron buenas notas en ambos trimestres aprobaron, mientras los que tuvieron malas notas suspendieron.

Además, se observa, también en ambas asignaturas, que la mayoría de los alumnos obtuvieron más o menos la misma nota en el segundo trimestre que en el primer trimestre, a excepción de unos pocos que obtuvieron una mejor nota en el primer trimestre que en el segundo.

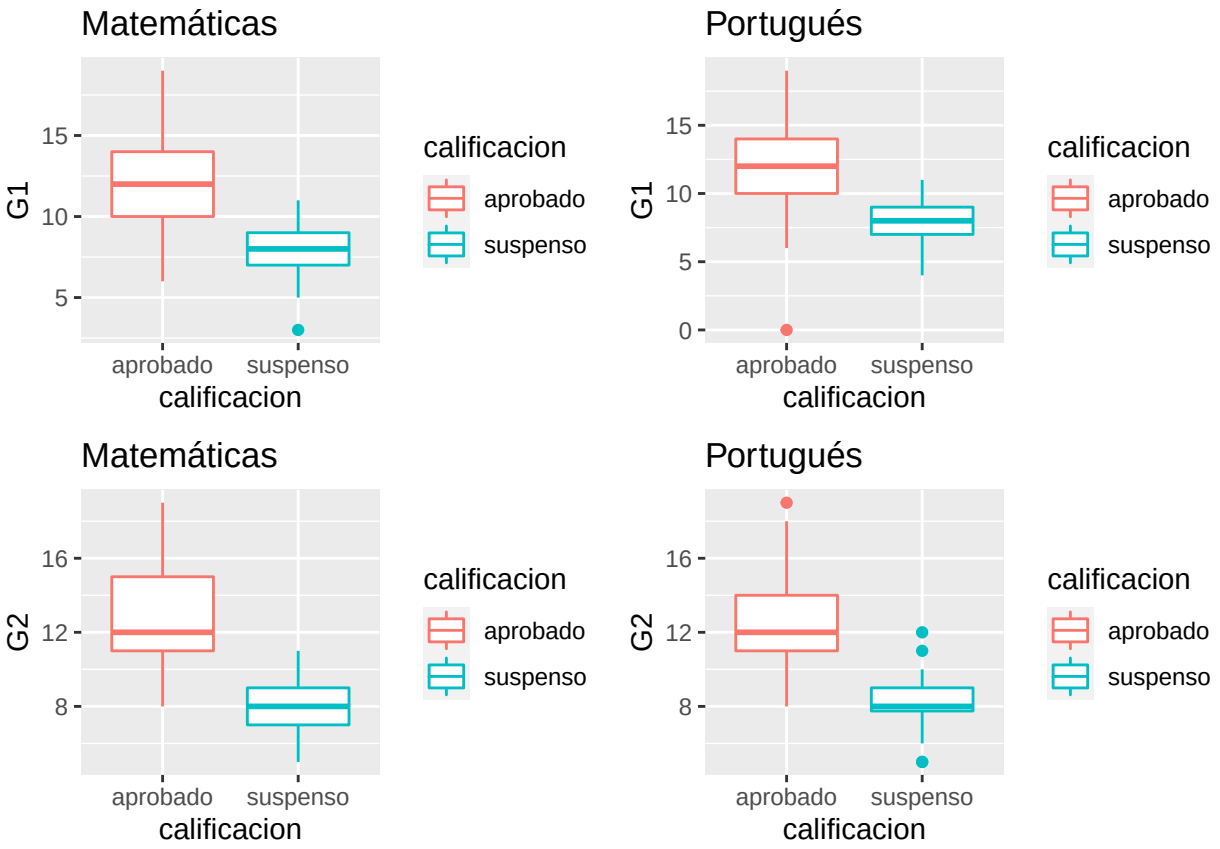
Se representan a continuación las variables numéricas según la calificación, suspenso o aprobado, mediante diagramas de cajas.

```
p1 <- ggplot(notas_m, aes(x=calificacion, y=age, color=calificacion)) +
geom_boxplot() + labs(title="Matemáticas")
p2 <- ggplot(notas_p, aes(x=calificacion, y=age, color=calificacion)) +
geom_boxplot() + labs(title="Portugués")
p3 <- ggplot(notas_m, aes(x=calificacion, y=absences, color=calificacion)) +
geom_boxplot() + labs(title="Matemáticas")
p4 <- ggplot(notas_p, aes(x=calificacion, y=absences, color=calificacion)) +
geom_boxplot() + labs(title="Portugués")
grid.arrange(p1, p2, p3, p4, nrow = 2, ncol=2)
```



La media de edad es notablemente menor en los aprobados que en los suspensos. En el resto de estas variables se comportan igual ambos grupos.

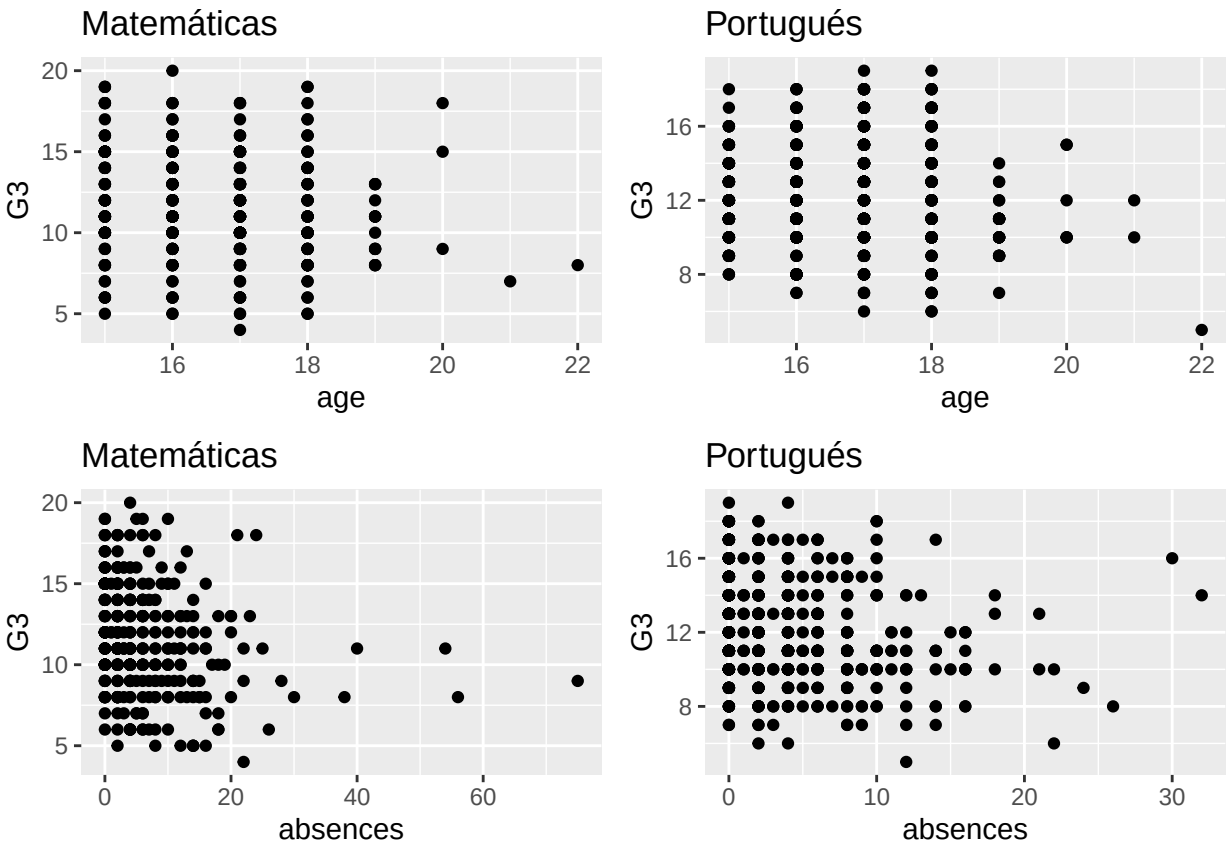
```
p5 <- ggplot(notas_m, aes(x=calificacion, y=G1, color=calificacion)) +
geom_boxplot() + labs(title="Matemáticas")
p6 <- ggplot(notas_p, aes(x=calificacion, y=G1, color=calificacion)) +
geom_boxplot() + labs(title="Portugués")
p7 <- ggplot(notas_m, aes(x=calificacion, y=G2, color=calificacion)) +
geom_boxplot() + labs(title="Matemáticas")
p8 <- ggplot(notas_p, aes(x=calificacion, y=G2, color=calificacion)) +
geom_boxplot() + labs(title="Portugués")
grid.arrange(p5, p6, p7, p8, nrow = 2, ncol=2)
```



Como ya se ha mencionado antes y al estar relacionadas G1 y G2 con G3, aquellos alumnos que finalmente aprueban se encuentran en el rango superior en G1 y G2.

Se realizan a continuación varios diagramas de dispersión de la nota final frente las variables númericas de los datos (sin tener en cuenta G1 y G2) para detectar si existe alguna tendencia simple.

```
p1 <- qplot(age, G3, data = notas_m) + labs(title="Matemáticas")
p2 <- qplot(age, G3, data = notas_p) + labs(title="Portugués")
p3 <- qplot(absences, G3, data = notas_m) + labs(title="Matemáticas")
p4 <- qplot(absences, G3, data = notas_p) + labs(title="Portugués")
grid.arrange(p1, p2, p3, p4, nrow = 2, ncol=2)
```



En cuanto a las distintas edades, no se nota ninguna diferencia notable entre ellas. Sin embargo, en la representación de G3 frente a las ausencias, se observa como aquellos alumnos que tienen de las mejores notas finales no tienen casi ausencias.

Con esto se termina la descriptiva básica y visualización de los datos.

Predicción de la nota final de forma numéricamente

Se realizarán a continuación los análisis considerando como se lleva haciendo las asignaturas separados. De esta forma luego también se podrá comparar los resultados obtenidos para cada una y ver que en que son similiares y en que difieren.

Como ya se mencióno anteriormente, se analizará la nota final en tres escenarios distintos: considerando que no se tiene ninguna nota previa, considerando que solo se tienen la nota del primer trimestre, la variable G1, y finalmente considerando que se tienen las notas de los dos trimestres previos, las variables G1 y G2.

Se utilizarán modelos de regresión lineal múltiple.

Escenario 1: sin G1 y G2

Asignatura: portugués Se pone a continuación la nota final de la asigntura de portugués en función del resto de variables exceptuando G1 y G2 mediante un modelo lineal.

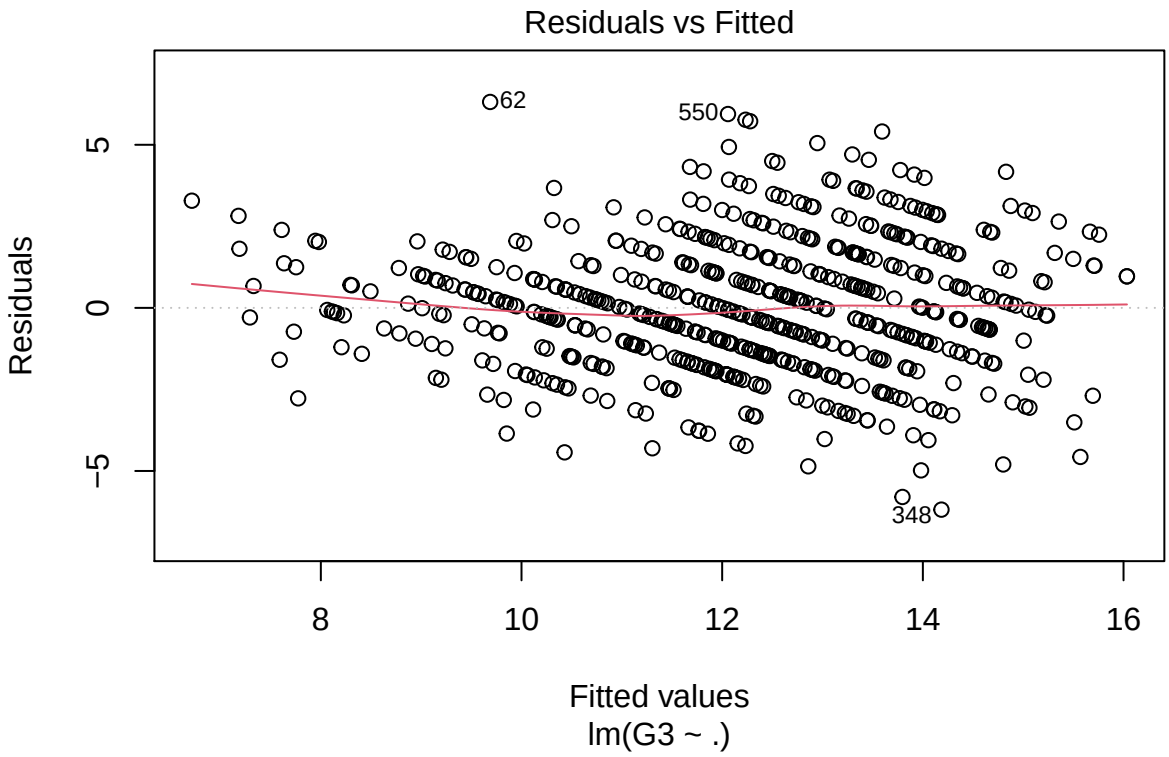
```
fit1 <- lm(G3 ~ ., data=notas_p[,!(names(notas_p) %in% c("G1","G2", "calificacion"))])
summary(fit1)
```

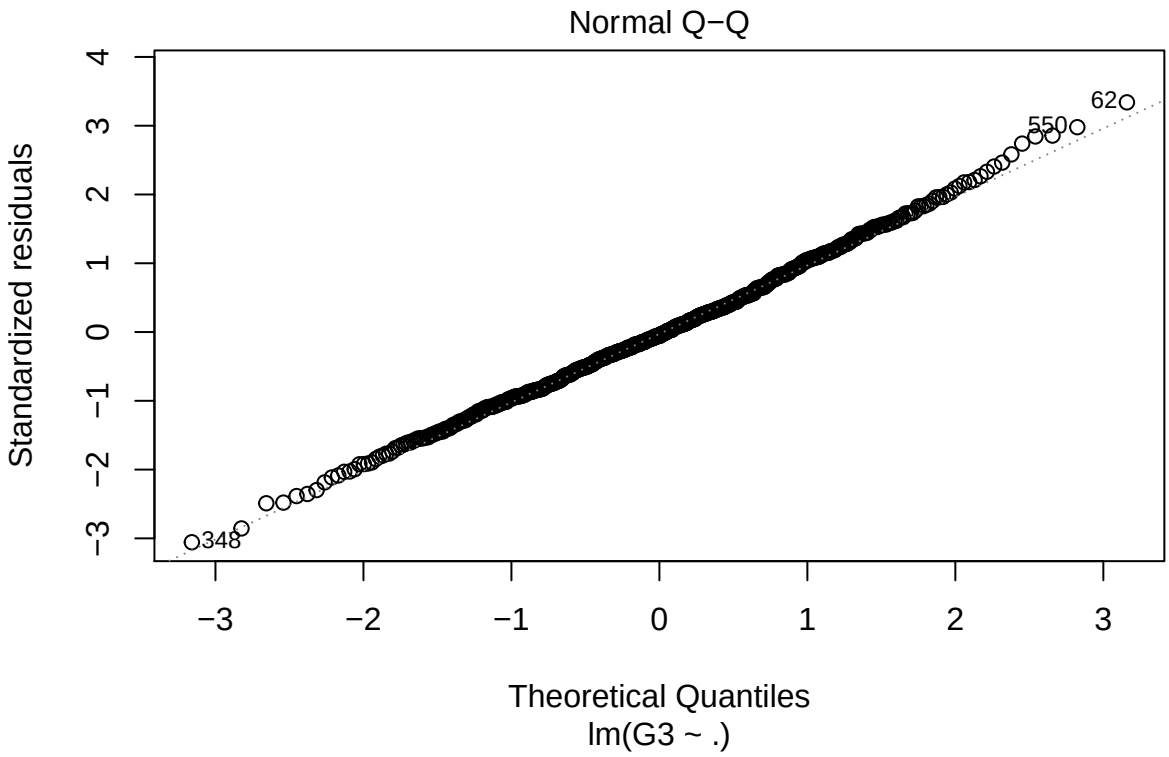
```
##
## Call:
## lm(formula = G3 ~ ., data = notas_p[, !(names(notas_p) %in% c("G1",
```

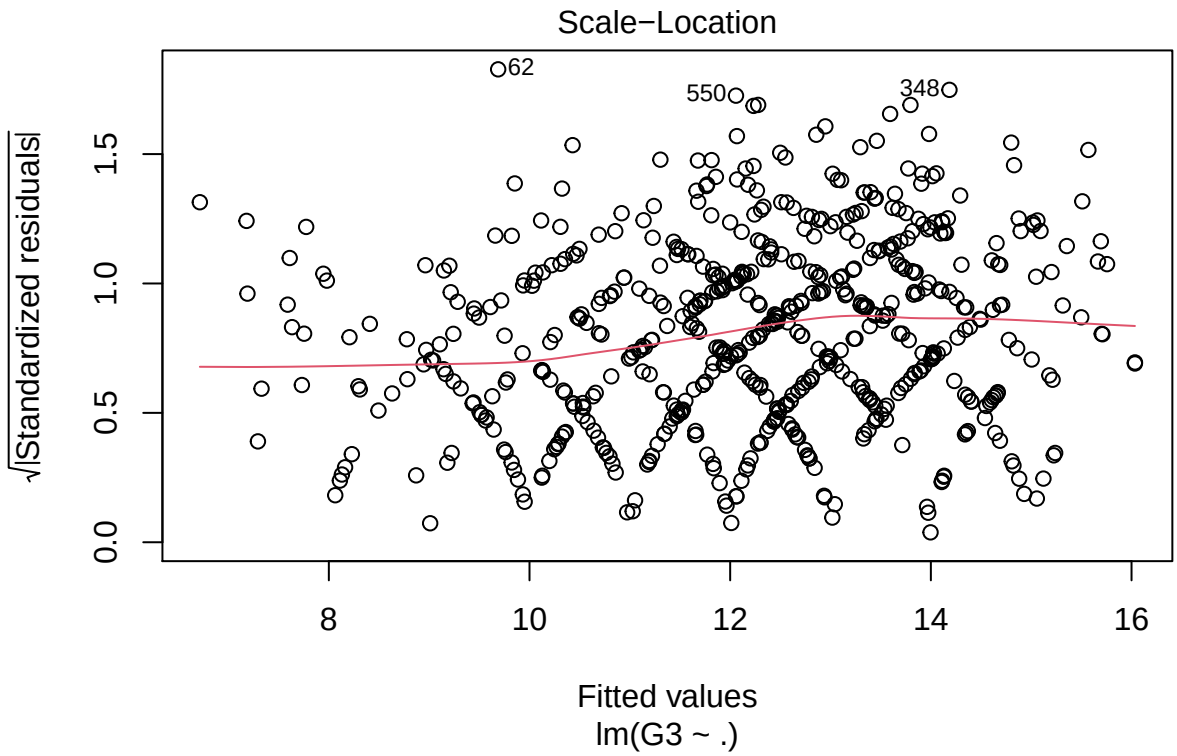
```
##     "G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1853 -1.3893 -0.1102  1.2831  6.3124
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.719562   2.090136   2.258 0.024327 *
## schoolMS                  -0.726846   0.225010  -3.230 0.001309 **
## sexhombre                 -0.627664   0.205338  -3.057 0.002344 **
## age                        0.321211   0.084569   3.798 0.000162 ***
## addressRural               0.153562   0.215826   0.712 0.477064
## famsizeLE3                 0.025745   0.199805   0.129 0.897523
## Pstatusseparados          -0.076771   0.286771  -0.268 0.789023
## Medu<=4ºEP                 0.365641   0.934600   0.391 0.695777
## Medu5ºEP-3ºESO             0.362738   0.940337   0.386 0.699825
## Medu4ºESO-2ºBachiller      0.538489   0.951451   0.566 0.571642
## Meduestudios superiores    1.102204   0.981156   1.123 0.261758
## Fedu<=4ºEP                -0.517962   0.871310  -0.594 0.552441
## Fedu5ºEP-3ºESO            -0.032762   0.878492  -0.037 0.970264
## Fedu4ºESO-2ºBachiller     -0.378286   0.894088  -0.423 0.672386
## Feduestudios superiores   -0.187171   0.919891  -0.203 0.838841
## Mjobhealth                 0.288021   0.444756   0.648 0.517512
## Mjobother                  0.139149   0.248673   0.560 0.575998
## Mjobservices               0.294045   0.304515   0.966 0.334651
## Mjobteacher                0.231314   0.430012   0.538 0.590842
## Fjobhealth                -0.419994   0.614863  -0.683 0.494844
## Fjobother                 -0.073107   0.377177  -0.194 0.846381
## Fjobservices              -0.340666   0.396268  -0.860 0.390329
## Fjobteacher                0.881793   0.565086   1.560 0.119214
## reasonhome                 0.274353   0.232878   1.178 0.239255
## reasonother               -0.148261   0.302584  -0.490 0.624338
## reasonreputation           0.336905   0.241980   1.392 0.164387
## guardianmother            -0.406743   0.217681  -1.869 0.062207 .
## guardianother             -0.239918   0.443779  -0.541 0.588979
## traveltime15-30 min        0.124912   0.203513   0.614 0.539608
## traveltime30 min.-1 hora   0.531700   0.351633   1.512 0.131073
## traveltime>1 hora         -0.725129   0.588260  -1.233 0.218214
## studytime2-5 horas         0.345712   0.219710   1.573 0.116166
## studytime5-10 horas        0.637351   0.298383   2.136 0.033109 *
## studytime>10 horas         0.834551   0.422283   1.976 0.048610 *
## failures1                 -1.788704   0.322318  -5.549 4.42e-08 ***
## failures2                 -2.506291   0.615321  -4.073 5.30e-05 ***
## failures>=3               -2.802290   0.642452  -4.362 1.53e-05 ***
## schoolsupyes              -1.142388   0.299172  -3.818 0.000149 ***
## famsupyes                 -0.241963   0.187375  -1.291 0.197118
## paidyes                   -0.376843   0.378009  -0.997 0.319233
## activitiesyes              0.258863   0.183182   1.413 0.158165
## nurseryyes                -0.063513   0.223011  -0.285 0.775902
## higheryes                  1.626136   0.319893   5.083 5.05e-07 ***
## internetyes                0.185264   0.229281   0.808 0.419421
## romanticyes               -0.219320   0.189279  -1.159 0.247064
## famrelmal                  0.824033   0.657619   1.253 0.210706
```

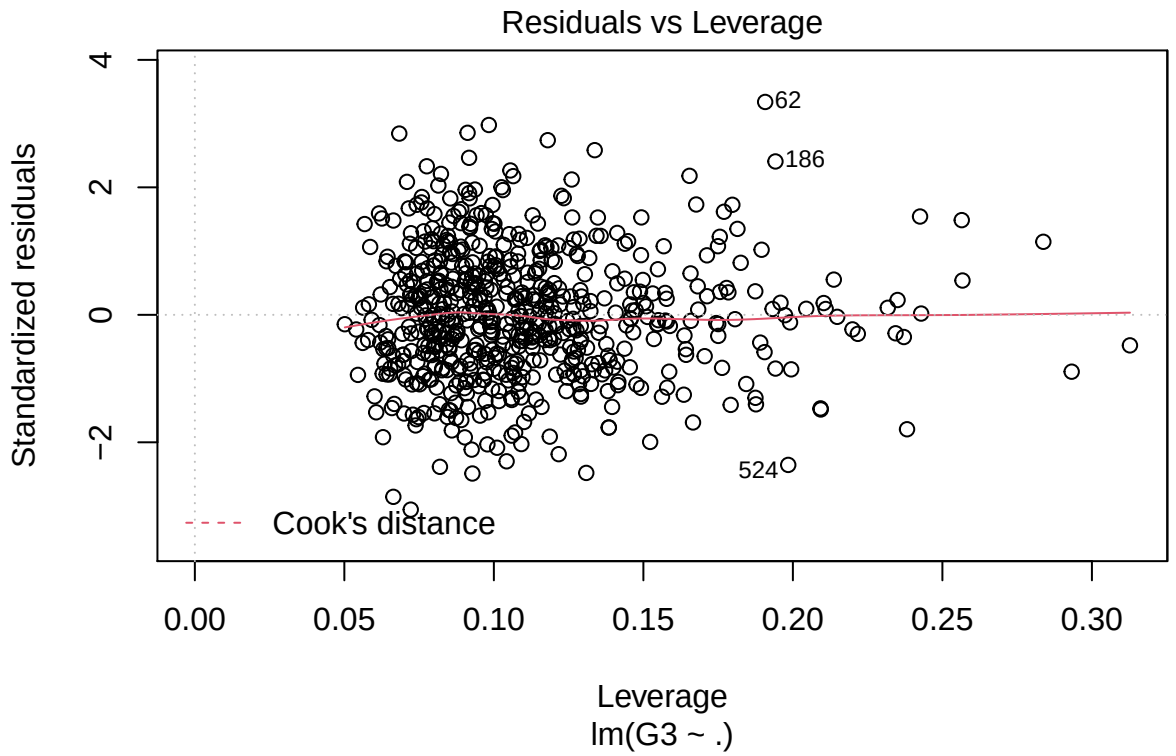
```
## famrelregular             0.695629   0.551229   1.262 0.207486
## famrelbien                1.212590   0.516364   2.348 0.019203 *
## famrelmuy bien            1.041771   0.528461   1.971 0.049175 *
## freetimepoco              0.968316   0.398069   2.433 0.015304 *
## freetimealgo              0.245493   0.365473   0.672 0.502042
## freetimesuficiente        0.340451   0.390293   0.872 0.383419
## freetimemucho             0.413652   0.452520   0.914 0.361052
## gooutpoco                 0.955784   0.382271   2.500 0.012693 *
## gooutalgo                 0.449487   0.375581   1.197 0.231897
## gooutsuficiente           0.164281   0.398709   0.412 0.680473
## gooutmucho                0.001865   0.423852   0.004 0.996491
## Dalcpoco                 -0.271293   0.266213  -1.019 0.308603
## Dalcalgo                 -0.164038   0.422211  -0.389 0.697777
## Dalcsuficiente            0.119785   0.668638   0.179 0.857887
## Dalcmucho                -0.862488   0.692067  -1.246 0.213191
## Walcpoco                 -0.145076   0.243199  -0.597 0.551060
## Walcalgo                  0.113892   0.281926   0.404 0.686381
## Walcsuficiente           -0.406303   0.349959  -1.161 0.246133
## Walcmucho                 0.309860   0.530380   0.584 0.559304
## healthmal                -0.242905   0.347424  -0.699 0.484742
## healthregular            -0.638367   0.311592  -2.049 0.040952 *
## healthbien               -0.222650   0.320719  -0.694 0.487830
## healthmuy bien           -0.695279   0.284967  -2.440 0.015000 *
## absences                 -0.072928   0.020429  -3.570 0.000388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.101 on 563 degrees of freedom
## Multiple R-squared:  0.4429, Adjusted R-squared:  0.3746
## F-statistic: 6.487 on 69 and 563 DF,  p-value: < 2.2e-16
```

```
plot(fit1)
```


Residuals vs Fitted
62
550
348
Residuals
5
0
−5
8
10
12
14
16
Fitted values
lm(G3 ~ .)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(G3 ~ .)
62
550
348
−3
−2
−1
0
1
2
3
4


Scale−Location
62
550
348
√|Standardized residuals|
0.0
0.5
1.0
1.5
8
10
12
14
16
Fitted values
lm(G3 ~ .)



En este modelo ajustado únicamente resultan significativas para un nivel de significación del 5% las variables school (MS), sex (hombre), age, studytime (5-10 horas y >10 horas), failures (todos los niveles), schoolsup (yes), higher (si), famrel (bien y muy bien), freetime(poco), goout(poco), health(regular y muy bien) y absences.

El modelo estimado obtenido aproximando a dos decimales y teniendo únicamente en cuenta las variables significativas es:

$G3 = 4.72 - 0.63 * schoolMS - 0.66 * sexhombre + 0.32 * age + 0.64 * studytime5 - 10horas + 0.83 * studytime > 10horas - 1.79 * failures1 - 2.51 * failures2 - 2.80 * failures >= 3 - 1.14 * schoolsupyes + 1.63 * higheryes + 1.21 * famrelbien + 1.04 * famrelmuybien + 0.97 * freetimepoco + 0.96 * gooutpoco - 0.64 * healthregular - 0.70 * healthmuybien - 0.07 * absences$

En el gráfico de residuos frente a los valores predichos por el modelo se trata de una nube de puntos aleatoria de dispersión regular. Es cierto que a la izquierda no hay tantos puntos, pero el cambio no es drástico por lo que se asume homocedasticidad. Del gráfico Normal Q-Q se puede asumir la normalidad de los residuos al aproximarse todos los puntos a la recta diagonal. Se puede considerar el modelo como válido.

El coeficiente de determinación obtenido es 0.44. Este valor es bajo, más del 50% de la variabilidad de la nota final no es explicada por el modelo. Esto se puede deber a que las variables no estén altamente relacionadas con la nota final al tratarse de aspectos sociales y no sobre la eduación o inteligencia de un alumno.

```
R_Cuadrado_p<-c(0.4429)
```

La tabla anova correspondiente a este ajuste es la siguiente:

```
anova(fit1)
```

```
## Analysis of Variance Table
##
```

```
## Response: G3
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## school        1  231.17 231.167 52.3671 1.515e-12 ***
## sex           1   95.91  95.908 21.7264 3.930e-06 ***
## age           1    4.99   4.986  1.1296 0.2883149
## address       1   13.19  13.190  2.9881 0.0844292 .
## famsize       1    5.52   5.518  1.2500 0.2640372
## Pstatus       1    3.78   3.780  0.8563 0.3551799
## Medu          4  297.38  74.344 16.8414 4.736e-13 ***
## Fedu          4   28.08   7.019  1.5901 0.1754301
## Mjob          4   20.33   5.082  1.1513 0.3314910
## Fjob          4   34.73   8.683  1.9670 0.0981057 .
## reason        3   89.48  29.828  6.7571 0.0001753 ***
## guardian      2   50.80  25.399  5.7537 0.0033605 **
## traveltime    3    7.15   2.384  0.5400 0.6550775
## studytime     3  140.88  46.961 10.6382 8.216e-07 ***
## failures      3  425.97 141.990 32.1655 < 2.2e-16 ***
## schoolsup     1   62.58  62.578 14.1761 0.0001840 ***
## famsup        1    9.16   9.164  2.0759 0.1501971
## paid          1    1.64   1.636  0.3706 0.5428975
## activities    1    5.66   5.656  1.2813 0.2581435
## nursery       1    0.05   0.054  0.0122 0.9121402
## higher        1  143.91 143.913 32.6011 1.833e-08 ***
## internet      1    3.10   3.101  0.7024 0.4023440
## romantic      1   10.15  10.149  2.2990 0.1300158
## famrel        4   32.75   8.187  1.8546 0.1169802
## freetime      4   47.66  11.914  2.6989 0.0299951 *
## goout         4   73.90  18.475  4.1851 0.0023892 **
## Dalc          4   18.88   4.721  1.0695 0.3707765
## Walc          4   20.18   5.044  1.1426 0.3354989
## health        4   40.80  10.200  2.3106 0.0566725 .
## absences      1   56.26  56.258 12.7444 0.0003876 ***
## Residuals   563 2485.28   4.414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ajustando ahora el modelo a solo las variables significativas quedaría lo siguiente:

```
fit12 <- lm(G3 ~ school + sex + age + studytime + failures + schoolsup + higher + famrel + freetime + g
summary(fit12)
```

```
##
## Call:
## lm(formula = G3 ~ school + sex + age + studytime + failures +
##     schoolsup + higher + famrel + freetime + goout + health +
##     absences, data = notas_p[, !(names(notas_p) %in% c("G1",
##     "G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8752 -1.5703  0.0038  1.3577  6.6799
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.48095    1.54151   3.556 0.000407 ***
```

```
## schoolMS            -1.07643    0.19670  -5.472 6.52e-08 ***
## sexhombre           -0.45615    0.18783  -2.429 0.015452 *
## age                  0.28880    0.07984   3.617 0.000323 ***
## studytime2-5 horas   0.43527    0.21463   2.028 0.042996 *
## studytime5-10 horas  0.73081    0.29360   2.489 0.013074 *
## studytime>10 horas   0.97935    0.40798   2.400 0.016675 *
## failures1           -1.98545    0.31238  -6.356 4.08e-10 ***
## failures2           -2.72057    0.59189  -4.596 5.24e-06 ***
## failures>=3         -2.90625    0.62934  -4.618 4.74e-06 ***
## schoolsupyes        -1.20282    0.29236  -4.114 4.42e-05 ***
## higheryes            1.88809    0.31514   5.991 3.58e-09 ***
## famrelmal            0.82163    0.64222   1.279 0.201263
## famrelregular        0.66119    0.53939   1.226 0.220744
## famrelbien           1.18436    0.50574   2.342 0.019514 *
## famrelmuy bien       0.95407    0.51684   1.846 0.065385 .
## freetimepoco         0.99283    0.39167   2.535 0.011501 *
## freetimealgo         0.21810    0.35952   0.607 0.544325
## freetimesuficiente   0.36849    0.37900   0.972 0.331312
## freetimemucho        0.50441    0.44555   1.132 0.258038
## gooutpoco            0.74826    0.37818   1.979 0.048319 *
## gooutalgo            0.27366    0.36972   0.740 0.459479
## gooutsuficiente      0.05550    0.38804   0.143 0.886314
## gooutmucho          -0.19990    0.40177  -0.498 0.618988
## healthmal           -0.13414    0.34234  -0.392 0.695317
## healthregular       -0.75250    0.31000  -2.427 0.015499 *
## healthbien          -0.25679    0.31910  -0.805 0.421298
## healthmuy bien      -0.79073    0.27747  -2.850 0.004524 **
## absences            -0.07914    0.01984  -3.989 7.46e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.148 on 604 degrees of freedom
## Multiple R-squared:  0.3753, Adjusted R-squared:  0.3464
## F-statistic: 12.96 on 28 and 604 DF,  p-value: < 2.2e-16
```

Al comparar ambos modelos mediante anova se puede comprobar su igualdad.

```
anova(fit1,fit12)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences
## Model 2: G3 ~ school + sex + age + studytime + failures + schoolsup +
##     higher + famrel + freetime + goout + health + absences
##   Res.Df    RSS  Df Sum of Sq      F  Pr(>F)
## 1    563 2485.3
## 2    604 2786.9 -41   -301.63 1.6666 0.00681 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No se podrían considerar equivalentes.

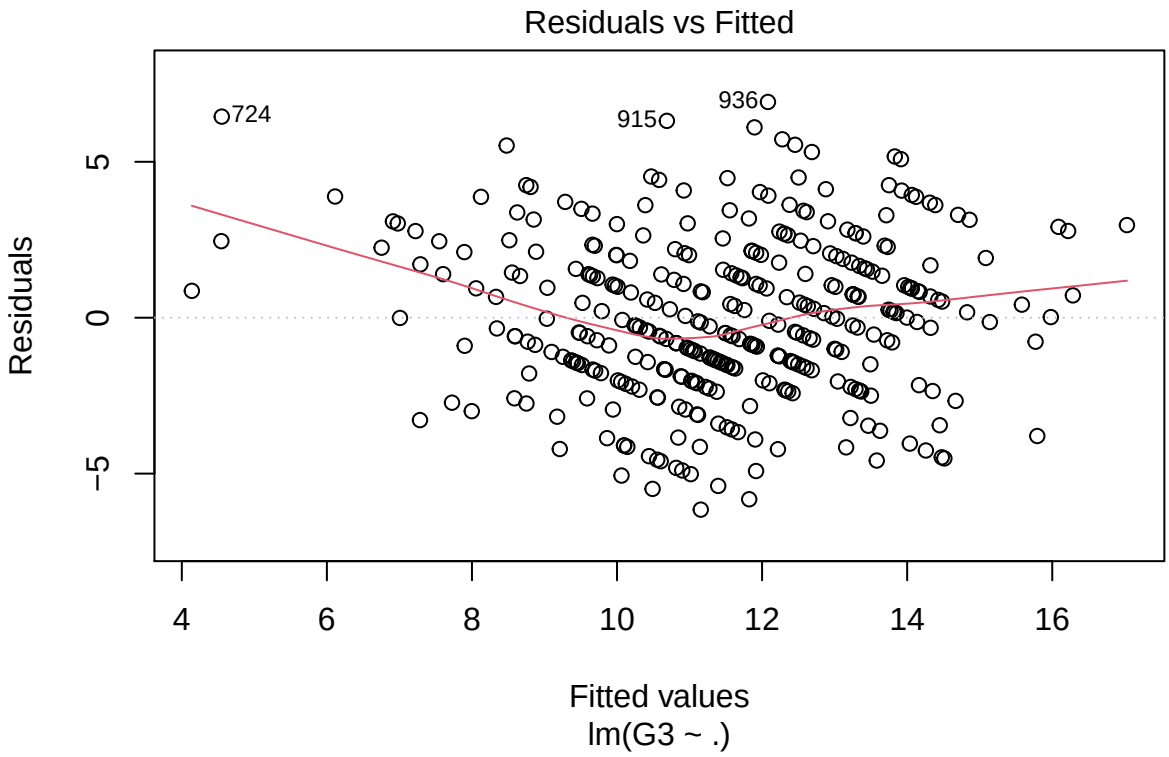
Asignatura: matemáticas Realizo a continuación el mismo proceso pero para la asigntura de matemáticas.

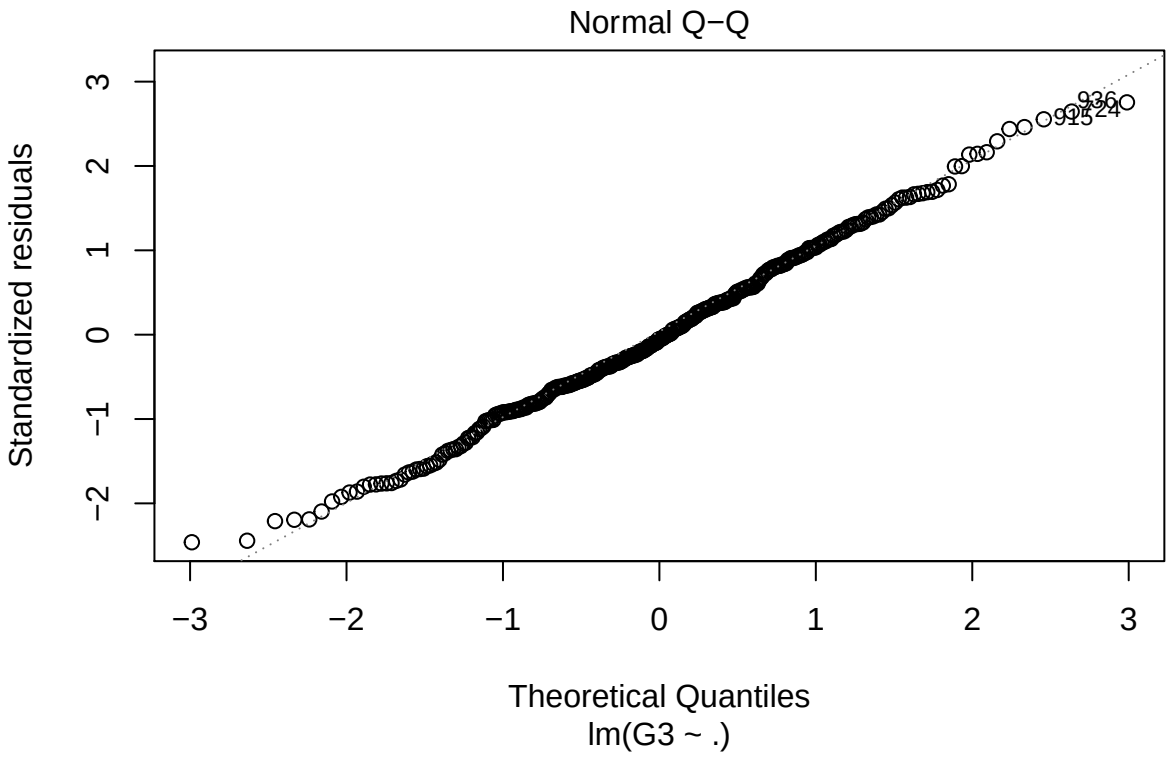
```
fit2 <- lm(G3 ~ ., data=notas_m[,!(names(notas_m) %in% c("G1","G2", "calificacion"))])
summary(fit2)
```

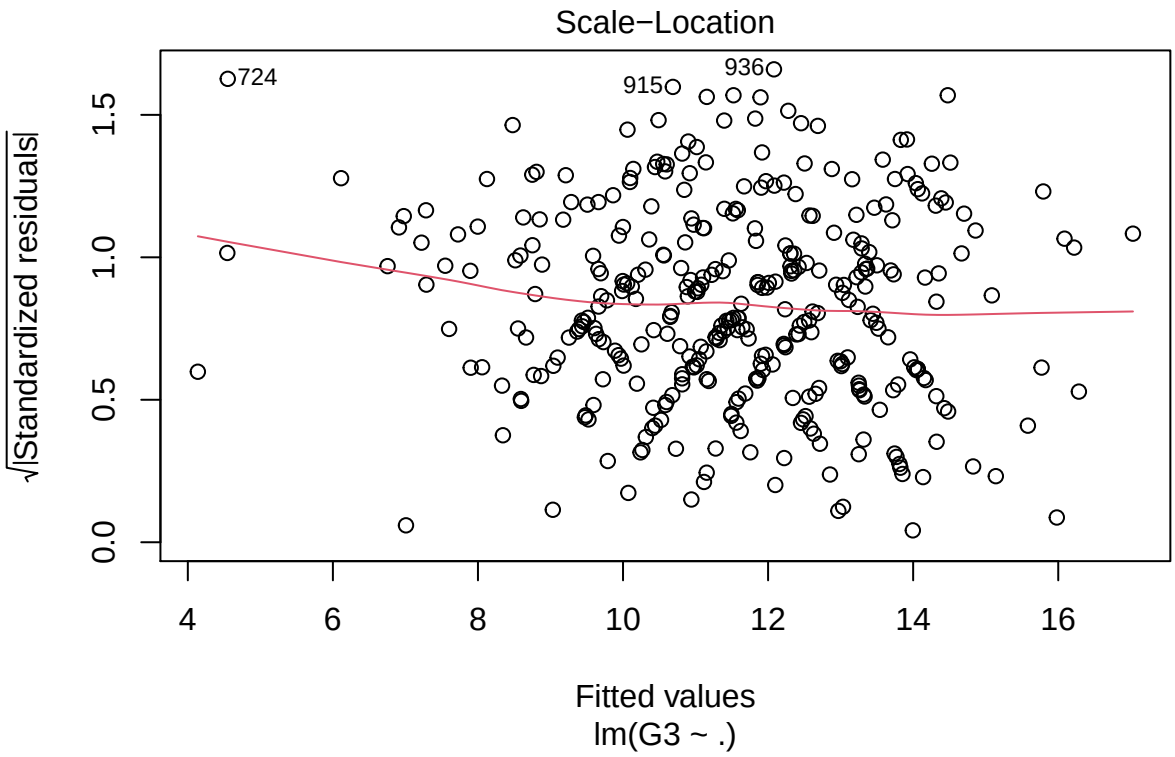
```
##
## Call:
## lm(formula = G3 ~ ., data = notas_m[, !(names(notas_m) %in% c("G1",
##     "G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1548 -1.6512 -0.1395  1.7651  6.9193
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                16.3876275  4.5067410   3.636 0.000328 ***
## schoolMS                   -0.4488923  0.6026190  -0.745 0.456940
## sexhombre                   0.4944555  0.3771191   1.311 0.190859
## age                        -0.1436580  0.1658611  -0.866 0.387140
## addressRural                0.5003329  0.4451124   1.124 0.261927
## famsizeLE3                  0.4258632  0.3664861   1.162 0.246194
## Pstatusseparados           -0.0392257  0.5491269  -0.071 0.943103
## Medu<=4ºEP                 -1.8279289  1.8235797  -1.002 0.317002
## Medu5ºEP-3ºESO             -1.5135216  1.8279059  -0.828 0.408354
## Medu4ºESO-2ºBachiller      -1.3389781  1.8499134  -0.724 0.469775
## Meduestudios superiores    -1.1749559  1.8951532  -0.620 0.535763
## Fedu<=4ºEP                 -0.5873791  2.1500130  -0.273 0.784898
## Fedu5ºEP-3ºESO             -0.2243371  2.1495569  -0.104 0.916953
## Fedu4ºESO-2ºBachiller      -0.4028606  2.1556246  -0.187 0.851880
## Feduestudios superiores     0.1404757  2.1918913   0.064 0.948944
## Mjobhealth                  1.0640817  0.8540409   1.246 0.213803
## Mjobother                  -0.4193655  0.5600286  -0.749 0.454574
## Mjobservices                0.7881497  0.6276407   1.256 0.210234
## Mjobteacher                -0.6412297  0.8030342  -0.799 0.425236
## Fjobhealth                 -0.7710756  1.0719875  -0.719 0.472544
## Fjobother                  -0.3432213  0.7935057  -0.433 0.665675
## Fjobservices               -0.5104490  0.8212349  -0.622 0.534723
## Fjobteacher                 1.2580338  1.0259388   1.226 0.221118
## reasonhome                  0.4063955  0.4229778   0.961 0.337463
## reasonother                -0.0434564  0.5953190  -0.073 0.941860
## reasonreputation            0.2605282  0.4337649   0.601 0.548567
## guardianmother              0.0005368  0.4043898   0.001 0.998942
## guardianother               0.7075242  0.7896999   0.896 0.371035
## traveltime15-30 min        -0.1799017  0.3861484  -0.466 0.641649
## traveltime30 min.-1 hora    0.7526629  0.7666929   0.982 0.327074
## traveltime>1 hora          -0.3564707  1.3131834  -0.271 0.786236
## studytime2-5 horas         -0.0288839  0.4188404  -0.069 0.945068
## studytime5-10 horas         1.1665822  0.5783504   2.017 0.044617 *
## studytime>10 horas          1.3718179  0.7548003   1.817 0.070190 .
## failures1                  -0.9161646  0.5716058  -1.603 0.110081
## failures2                  -2.3498959  0.9675844  -2.429 0.015771 *
## failures>=3                -2.9883473  1.0486520  -2.850 0.004693 **
## schoolsupyes               -2.1464153  0.4898920  -4.381 1.66e-05 ***
## famsupyes                  -0.7001068  0.3539265  -1.978 0.048872 *
```

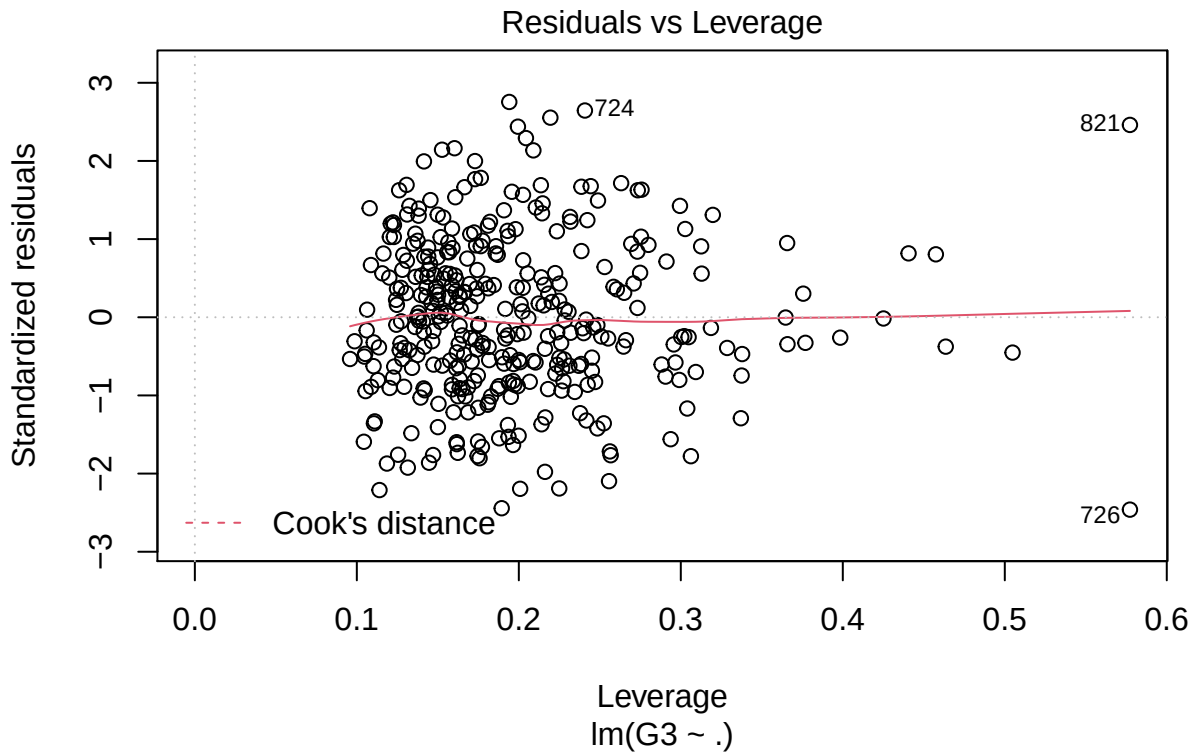
```
## paidyes                 -0.2760804  0.3622072  -0.762 0.446557
## activitiesyes           -0.0709241  0.3399726  -0.209 0.834895
## nurseryyes              -0.2647913  0.4219225  -0.628 0.530776
## higheryes                0.1364896  0.9491572   0.144 0.885759
## internetyes              0.6124148  0.4620859   1.325 0.186117
## romanticyes              0.0045701  0.3698133   0.012 0.990149
## famrelmal                1.4149429  1.4362608   0.985 0.325377
## famrelregular            0.5420714  1.2408954   0.437 0.662557
## famrelbien               0.6541311  1.2016741   0.544 0.586623
## famrelmuy bien           0.9565055  1.2211401   0.783 0.434103
## freetimepoco             0.7413219  0.8618280   0.860 0.390411
## freetimealgo             0.1600978  0.8320449   0.192 0.847553
## freetimesuficiente       0.1761449  0.8597077   0.205 0.837804
## freetimemucho            1.2587545  0.9765529   1.289 0.198444
## gooutpoco                0.4026809  0.7831788   0.514 0.607534
## gooutalgo               -0.4285564  0.7855925  -0.546 0.585820
## gooutsuficiente         -0.8924729  0.8315311  -1.073 0.284043
## gooutmucho              -1.2999997  0.8912443  -1.459 0.145759
## Dalcpoco                -0.0927190  0.4973351  -0.186 0.852238
## Dalcalgo                 0.0414400  0.7658671   0.054 0.956886
## Dalcsuficiente          -0.9338349  1.1630726  -0.803 0.422695
## Dalcmucho               -1.0023291  1.3535834  -0.741 0.459602
## Walcpoco                -0.5710013  0.4677684  -1.221 0.223204
## Walcalgo                -0.2848520  0.5114447  -0.557 0.577992
## Walcsuficiente          -0.9737030  0.6571246  -1.482 0.139500
## Walcmucho                0.7122592  1.0475582   0.680 0.497101
## healthmal               -0.6111764  0.6765174  -0.903 0.367063
## healthregular           -1.5303607  0.5876752  -2.604 0.009691 **
## healthbien              -1.0142070  0.6277501  -1.616 0.107275
## healthmuy bien          -1.2981123  0.5502698  -2.359 0.018992 *
## absences                -0.0558763  0.0219639  -2.544 0.011483 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.798 on 287 degrees of freedom
## Multiple R-squared:  0.3942, Adjusted R-squared:  0.2485
## F-statistic: 2.706 on 69 and 287 DF,  p-value: 3.845e-09
```

```
plot(fit2)
```


Residuals vs Fitted
724
915
936
Residuals
5
0
−5
4
6
8
10
12
14
16
Fitted values
lm(G3 ~ .)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(G3 ~ .)
−3
−2
−1
0
1
2
3
936
915
924


Scale−Location
724
915
936
√|Standardized residuals|
0.0
0.5
1.0
1.5
4
6
8
10
12
14
16
Fitted values
lm(G3 ~ .)



El modelo estimado obtenido aproximando a dos decimales y teniendo únicamente en cuenta las variables significativas es:

$G3 = 16.65 - 2.23 * failures2 - 2.95 * failures >= 3 - 2.33 * schoolsupyes - 0.69 * famsupyes - 1.53 * healthregular - 1.38 * healthmuybien - 0.08 * absences$

El coeficiente de determinación obtenido es 0.4089, el cual también es bajo.

```
R_Cuadrado_m<-c(0.4089)
```

Sin embargo, en esta asignatura a diferencia de en la de portugués pese a cumplir el supuesto de normalidad, se puede considerar que la nube de puntos del gráfico residuals vs fitted sigue un patrón al no haber casi puntos en los laterales por lo que visualmente no se podría asegurar el supuesto de homocedasticidad.

El anova correspondiente es:

```
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: G3
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## school       1   25.93  25.932  3.3122  0.069809 .
## sex          1   38.67  38.670  4.9392  0.027034 *
## age          1   49.05  49.047  6.2646  0.012873 *
## address      1   43.39  43.394  5.5426  0.019233 *
## famsize      1    3.43   3.426  0.4376  0.508812
## Pstatus      1    1.09   1.085  0.1386  0.709916
## Medu         4  124.75  31.187  3.9834  0.003669 **
## Fedu         4   33.21   8.302  1.0604  0.376397
```

```
## Mjob         4   69.00  17.251  2.2035  0.068735 .
## Fjob         4   55.18  13.794  1.7619  0.136596
## reason       3   14.87   4.956  0.6331  0.594215
## guardian     2    0.60   0.301  0.0385  0.962267
## traveltime   3   25.20   8.400  1.0729  0.360881
## studytime    3  195.33  65.110  8.3164 2.540e-05 ***
## failures     3  207.76  69.254  8.8457 1.258e-05 ***
## schoolsup    1  179.61 179.611 22.9413 2.681e-06 ***
## famsup       1   46.51  46.509  5.9405  0.015404 *
## paid         1    7.91   7.913  1.0106  0.315595
## activities   1    0.62   0.624  0.0797  0.777865
## nursery      1    4.95   4.955  0.6329  0.426963
## higher       1    0.95   0.953  0.1217  0.727420
## internet     1    2.86   2.859  0.3652  0.546091
## romantic     1    1.01   1.014  0.1296  0.719134
## famrel       4   11.90   2.975  0.3800  0.822882
## freetime     4   47.18  11.796  1.5067  0.200277
## goout        4   80.50  20.126  2.5706  0.038173 *
## Dalc         4   16.17   4.044  0.5165  0.723687
## Walc         4   54.70  13.676  1.7468  0.139777
## health       4   69.05  17.262  2.2048  0.068589 .
## absences     1   50.67  50.670  6.4719  0.011483 *
## Residuals  287 2246.97   7.829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modelo ajustado únicamente a las variables significativas resultantes:

```
fit21 <- lm(G3 ~ studytime + failures + schoolsup + famsup + health + absences, data=notas_m[,!(names(n
summary(fit21)
```

```
##
## Call:
## lm(formula = G3 ~ studytime + failures + schoolsup + famsup +
##     health + absences, data = notas_m[, !(names(notas_m) %in%
##     c("G1", "G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6683 -1.9188  0.0016  1.9075  7.5977
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         13.57613    0.55158  24.613  < 2e-16 ***
## studytime2-5 horas  -0.37396    0.38288  -0.977 0.329408
## studytime5-10 horas  0.91857    0.50409   1.822 0.069291 .
## studytime>10 horas   0.96150    0.68873   1.396 0.163602
## failures1           -1.09696    0.51458  -2.132 0.033739 *
## failures2           -2.56896    0.86988  -2.953 0.003362 **
## failures>=3         -3.28747    0.90740  -3.623 0.000335 ***
## schoolsupyes        -2.02401    0.45163  -4.482 1.01e-05 ***
## famsupyes           -0.38415    0.32707  -1.175 0.241004
## healthmal           -0.13651    0.64580  -0.211 0.832721
## healthregular       -1.37545    0.54767  -2.511 0.012482 *
## healthbien          -0.96758    0.59045  -1.639 0.102193
```

```
## healthmuy bien      -0.93234    0.51074  -1.825 0.068802 .
## absences            -0.06484    0.01966  -3.298 0.001076 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.919 on 343 degrees of freedom
## Multiple R-squared:  0.212,  Adjusted R-squared:  0.1821
## F-statistic: 7.097 on 13 and 343 DF,  p-value: 3.019e-12
```

Comparación de ambos modelos:

```
anova(fit2,fit21)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences
## Model 2: G3 ~ studytime + failures + schoolsup + famsup + health + absences
##   Res.Df    RSS  Df Sum of Sq      F  Pr(>F)
## 1    287 2247.0
## 2    343 2922.9 -56   -675.91 1.5416 0.01259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Al igual que en el caso de la asignatura de portugués, los dos modelos no pueden considerarse equivalentes al resultar el contraste significativo.

Escenario 2: con G1 y sin G2 En este escenario y en el siguiente existirá colinealidad al estar altamente correlacionadas las notas por lo tanto los modelos no son fiables. Sin embargo, se procederá igual.

Asignatura: portugués Ajuste de la nota final a todas las variables exceptuando G2 y la variable creada calificación.

```
fit3 <- lm(G3 ~ ., data=notas_p[,!(names(notas_p) %in% c("G2", "calificacion"))])
summary(fit3)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = notas_p[, !(names(notas_p) %in% c("G2",
##     "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5050 -0.7557 -0.0615  0.7031  6.4875
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.471e+00  1.230e+00  -1.196 0.232263
## schoolMS                 -2.643e-05  1.327e-01   0.000 0.999841
## sexhombre                -1.634e-01  1.203e-01  -1.358 0.174853
## age                       2.692e-01  4.922e-02   5.470  6.8e-08 ***
## addressRural              5.473e-02  1.256e-01   0.436 0.663174
## famsizeLE3               -1.587e-01  1.164e-01  -1.364 0.173056
```

```
## Pstatusseparados         -1.534e-01  1.668e-01  -0.920 0.358196
## Medu<=4ºEP               -1.237e-02  5.438e-01  -0.023 0.981857
## Medu5ºEP-3ºESO            1.176e-01  5.471e-01   0.215 0.829879
## Medu4ºESO-2ºBachiller     1.644e-01  5.536e-01   0.297 0.766585
## Meduestudios superiores   3.347e-01  5.712e-01   0.586 0.558145
## Fedu<=4ºEP                1.142e-01  5.072e-01   0.225 0.821970
## Fedu5ºEP-3ºESO            1.893e-01  5.111e-01   0.370 0.711303
## Fedu4ºESO-2ºBachiller     6.633e-02  5.203e-01   0.127 0.898608
## Feduestudios superiores   1.507e-01  5.352e-01   0.282 0.778379
## Mjobhealth                4.735e-02  2.588e-01   0.183 0.854908
## Mjobother                -5.222e-02  1.448e-01  -0.361 0.718472
## Mjobservices             -5.424e-02  1.775e-01  -0.306 0.759972
## Mjobteacher               6.939e-03  2.502e-01   0.028 0.977889
## Fjobhealth               -4.556e-01  3.577e-01  -1.274 0.203276
## Fjobother                -4.543e-01  2.197e-01  -2.068 0.039115 *
## Fjobservices             -5.726e-01  2.306e-01  -2.483 0.013322 *
## Fjobteacher              -1.822e-01  3.303e-01  -0.552 0.581377
## reasonhome                6.038e-02  1.356e-01   0.445 0.656365
## reasonother              -2.724e-01  1.761e-01  -1.547 0.122425
## reasonreputation          7.456e-02  1.410e-01   0.529 0.597107
## guardianmother           -9.204e-02  1.270e-01  -0.725 0.468879
## guardianother            -2.342e-02  2.582e-01  -0.091 0.927762
## traveltime15-30 min       1.662e-01  1.184e-01   1.403 0.161057
## traveltime30 min.-1 hora  3.454e-01  2.046e-01   1.688 0.091975 .
## traveltime>1 hora        -1.137e-01  3.427e-01  -0.332 0.740268
## studytime2-5 horas        9.414e-02  1.280e-01   0.735 0.462469
## studytime5-10 horas       1.178e-01  1.743e-01   0.676 0.499264
## studytime>10 horas       -1.506e-01  2.474e-01  -0.609 0.542897
## failures1                -6.410e-01  1.907e-01  -3.362 0.000827 ***
## failures2                -2.528e-01  3.643e-01  -0.694 0.488109
## failures>=3              -1.192e+00  3.769e-01  -3.162 0.001652 **
## schoolsupyes             -1.988e-01  1.763e-01  -1.128 0.259972
## famsupyes                -6.976e-02  1.091e-01  -0.639 0.522887
## paidyes                  -2.739e-02  2.202e-01  -0.124 0.901016
## activitiesyes             5.093e-02  1.067e-01   0.477 0.633434
## nurseryyes                6.444e-02  1.298e-01   0.496 0.619738
## higheryes                 4.850e-01  1.892e-01   2.563 0.010637 *
## internetyes               1.368e-01  1.334e-01   1.025 0.305657
## romanticyes              -1.014e-01  1.102e-01  -0.920 0.357903
## famrelmal                -2.132e-02  3.834e-01  -0.056 0.955671
## famrelregular             1.835e-01  3.210e-01   0.572 0.567838
## famrelbien                4.626e-01  3.012e-01   1.536 0.125148
## famrelmuy bien            4.473e-01  3.079e-01   1.453 0.146908
## freetimepoco              5.025e-01  2.320e-01   2.166 0.030731 *
## freetimealgo              1.677e-01  2.126e-01   0.789 0.430684
## freetimesuficiente        1.469e-01  2.271e-01   0.647 0.517989
## freetimemucho             2.480e-01  2.633e-01   0.942 0.346625
## gooutpoco                 2.428e-01  2.234e-01   1.087 0.277538
## gooutalgo                -4.303e-02  2.190e-01  -0.196 0.844303
## gooutsuficiente          -3.779e-02  2.320e-01  -0.163 0.870675
## gooutmucho               -2.549e-01  2.467e-01  -1.033 0.301864
## Dalcpoco                 -1.251e-01  1.549e-01  -0.808 0.419632
## Dalcalgo                  2.923e-02  2.457e-01   0.119 0.905342
## Dalcsuficiente            2.073e-01  3.890e-01   0.533 0.594339
```

```
## Dalcmucho               -1.781e-01  4.031e-01  -0.442 0.658857
## Walcpoco                -1.966e-01  1.415e-01  -1.390 0.165169
## Walcalgo                 7.804e-02  1.640e-01   0.476 0.634385
## Walcsuficiente          -1.304e-01  2.038e-01  -0.640 0.522516
## Walcmucho                1.903e-01  3.086e-01   0.617 0.537614
## healthmal               -2.066e-01  2.021e-01  -1.022 0.307094
## healthregular           -2.246e-01  1.817e-01  -1.236 0.216862
## healthbien              -2.298e-01  1.866e-01  -1.232 0.218613
## healthmuy bien          -4.802e-01  1.659e-01  -2.894 0.003945 **
## absences                -2.678e-02  1.196e-02  -2.238 0.025583 *
## G1                       7.769e-01  2.341e-02  33.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.222 on 562 degrees of freedom
## Multiple R-squared:  0.8118, Adjusted R-squared:  0.7884
## F-statistic: 34.63 on 70 and 562 DF,  p-value: < 2.2e-16
```

Resultan significativas las siguientes variables: age, Fjob, failures, higher, freetime, health, absences y G1.

El coeficiente de determinación cambia drásticamente en este escenario. Pasa de estar anteriormente, cuando no se consideraba la nota del primer trimestre, alrededor de 0.4 a estar ahora en 0.8 considerando G1. Esto se debe a que la nota final está ligada a la nota del primer trimestre al tenerla tenerla en cuenta. Además, que por lo normal, aquellos alumnos que les va bien en las pruebas intermedias también les vaya bien en el final y lo mismo con los alumnos que van mal.

```
R_Cuadrado_p<-c(R_Cuadrado_p, 0.8118)
```

Ajustando ahora el modelo a solo las variables significativas quedaría lo siguiente:

```
fit31 <- lm(G3 ~ age + Fjob + failures + higher + freetime +health + absences + G1, data=notas_p[,!(nam
summary(fit31)
```

```
##
## Call:
## lm(formula = G3 ~ age + Fjob + failures + higher + freetime +
##     health + absences + G1, data = notas_p[, !(names(notas_p) %in%
##     c("G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5784 -0.8143 -0.0136  0.7526  7.2698
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.48461    0.82982  -1.789 0.074094 .
## age                 0.27131    0.04357   6.227 8.84e-10 ***
## Fjobhealth         -0.33137    0.32284  -1.026 0.305105
## Fjobother          -0.37107    0.20302  -1.828 0.068078 .
## Fjobservices       -0.51331    0.21362  -2.403 0.016560 *
## Fjobteacher        -0.04380    0.28653  -0.153 0.878560
## failures1          -0.62564    0.17918  -3.492 0.000514 ***
## failures2          -0.23726    0.33872  -0.700 0.483901
## failures>=3        -1.11998    0.35662  -3.141 0.001767 **
## higheryes           0.54384    0.17850   3.047 0.002413 **
## freetimepoco        0.45485    0.21988   2.069 0.039000 *
```

```
## freetimealgo        0.16067    0.19893   0.808 0.419591
## freetimesuficiente  0.06947    0.20637   0.337 0.736503
## freetimemucho       0.15115    0.24024   0.629 0.529475
## healthmal          -0.15545    0.19191  -0.810 0.418258
## healthregular      -0.21652    0.17133  -1.264 0.206786
## healthbien         -0.19469    0.17748  -1.097 0.273084
## healthmuy bien     -0.46663    0.15308  -3.048 0.002401 **
## absences           -0.02824    0.01086  -2.601 0.009526 **
## G1                  0.81070    0.02107  38.482  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.218 on 613 degrees of freedom
## Multiple R-squared:  0.7962, Adjusted R-squared:  0.7899
## F-statistic:   126 on 19 and 613 DF,  p-value: < 2.2e-16
```

Comparación de ambos modelos:

```
anova(fit3,fit31)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences + G1
## Model 2: G3 ~ age + Fjob + failures + higher + freetime + health + absences +
##     G1
##   Res.Df    RSS  Df Sum of Sq      F Pr(>F)
## 1    562 839.55
## 2    613 909.27 -51   -69.724 0.9152 0.6425
```

En este caso ya sí se podrían considerar equivalentes el modelo completo y el modelo simplificado.

Asignatura: matemáticas Ajuste de la nota final de matemáticas a todas las variables exceptuando G2 y la variable creada calificación.

```
fit4 <- lm(G3 ~ ., data=notas_m[,!(names(notas_m) %in% c("G2", "calificacion"))])
summary(fit4)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = notas_m[, !(names(notas_m) %in% c("G2",
##     "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6083 -0.8357  0.0069  0.8672  3.7240
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.470600   2.277605   2.841  0.00482 **
## schoolMS                -0.122455   0.301391  -0.406  0.68483
## sexhombre               -0.065942   0.189445  -0.348  0.72804
## age                     -0.169012   0.082901  -2.039  0.04240 *
```

```
## addressRural              0.261891   0.222613   1.176  0.24040
## famsizeLE3                0.043001   0.183631   0.234  0.81502
## Pstatusseparados         -0.260989   0.274555  -0.951  0.34262
## Medu<=4ºEP               -0.713494   0.912205  -0.782  0.43477
## Medu5ºEP-3ºESO           -0.390781   0.914378  -0.427  0.66943
## Medu4ºESO-2ºBachiller    -0.208747   0.925378  -0.226  0.82169
## Meduestudios superiores  -0.244704   0.947717  -0.258  0.79644
## Fedu<=4ºEP               -0.472892   1.074573  -0.440  0.66022
## Fedu5ºEP-3ºESO           -0.693082   1.074457  -0.645  0.51941
## Fedu4ºESO-2ºBachiller    -0.549425   1.077382  -0.510  0.61047
## Feduestudios superiores  -0.502064   1.095715  -0.458  0.64715
## Mjobhealth                0.389114   0.427463   0.910  0.36344
## Mjobother                 0.194120   0.280678   0.692  0.48974
## Mjobservices              0.292933   0.314144   0.932  0.35188
## Mjobteacher              -0.124165   0.401738  -0.309  0.75749
## Fjobhealth               -0.177322   0.536155  -0.331  0.74109
## Fjobother                 0.300921   0.397196   0.758  0.44931
## Fjobservices              0.229972   0.411222   0.559  0.57644
## Fjobteacher               0.321803   0.513749   0.626  0.53156
## reasonhome                0.034169   0.211782   0.161  0.87194
## reasonother               0.256496   0.297713   0.862  0.38965
## reasonreputation         -0.150302   0.217244  -0.692  0.48959
## guardianmother            0.017400   0.202113   0.086  0.93145
## guardianother            -0.049985   0.395530  -0.126  0.89952
## traveltime15-30 min      -0.042004   0.193052  -0.218  0.82791
## traveltime30 min.-1 hora -0.219021   0.384614  -0.569  0.56949
## traveltime>1 hora         0.333528   0.656743   0.508  0.61195
## studytime2-5 horas       -0.246986   0.209466  -1.179  0.23933
## studytime5-10 horas      -0.189243   0.292718  -0.647  0.51847
## studytime>10 horas       -0.108921   0.380598  -0.286  0.77494
## failures1                 0.336880   0.288853   1.166  0.24448
## failures2                -0.084490   0.489704  -0.173  0.86314
## failures>=3               0.122138   0.534700   0.228  0.81948
## schoolsupyes             -0.364914   0.252244  -1.447  0.14909
## famsupyes                 0.147217   0.179227   0.821  0.41210
## paidyes                  -0.054122   0.181187  -0.299  0.76538
## activitiesyes            -0.052199   0.169918  -0.307  0.75891
## nurseryyes               -0.192019   0.210889  -0.911  0.36331
## higheryes                -0.180352   0.474507  -0.380  0.70417
## internetyes               0.405109   0.231056   1.753  0.08062 .
## romanticyes              -0.120753   0.184880  -0.653  0.51419
## famrelmal                -0.116339   0.719726  -0.162  0.87170
## famrelregular             0.086086   0.620388   0.139  0.88974
## famrelbien               -0.077856   0.601108  -0.130  0.89704
## famrelmuy bien            0.307842   0.610719   0.504  0.61460
## freetimepoco             -0.341831   0.432313  -0.791  0.42977
## freetimealgo             -0.186349   0.416019  -0.448  0.65454
## freetimesuficiente       -0.323573   0.430014  -0.752  0.45239
## freetimemucho            -0.603723   0.492177  -1.227  0.22097
## gooutpoco                 0.307157   0.391442   0.785  0.43329
## gooutalgo                -0.023699   0.392877  -0.060  0.95194
## gooutsuficiente          -0.330516   0.416035  -0.794  0.42760
## gooutmucho               -0.310650   0.446711  -0.695  0.48736
## Dalcpoco                 -0.001491   0.248585  -0.006  0.99522
```

```
## Dalcalgo                  0.035103   0.382777   0.092  0.92700
## Dalcsuficiente           -0.339246   0.581650  -0.583  0.56019
## Dalcmucho                -0.973905   0.676515  -1.440  0.15108
## Walcpoco                 -0.472339   0.233812  -2.020  0.04430 *
## Walcalgo                 -0.198603   0.255634  -0.777  0.43786
## Walcsuficiente           -0.167526   0.329572  -0.508  0.61163
## Walcmucho                 0.746679   0.523566   1.426  0.15492
## healthmal                -0.216178   0.338387  -0.639  0.52343
## healthregular            -0.607149   0.295394  -2.055  0.04075 *
## healthbien               -0.258590   0.314799  -0.821  0.41208
## healthmuy bien           -0.490543   0.276393  -1.775  0.07699 .
## absences                 -0.041711   0.010988  -3.796  0.00018 ***
## G1                        0.857464   0.029189  29.376  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.398 on 286 degrees of freedom
## Multiple R-squared:  0.8492, Adjusted R-squared:  0.8123
## F-statistic: 23.01 on 70 and 286 DF,  p-value: < 2.2e-16
```

Resultan significativas las siguientes variables: age, Walc, health, absences y G1.

El coeficiente de determinación al igual que en el caso de la asignatura de portugués incrementa drásticamente de lo que era antes pasando a 0.85.

```
R_Cuadrado_m<-c(R_Cuadrado_m, 0.8492)
```

Ajustando ahora el modelo a solo las variables significativas quedaría lo siguiente:

```
fit41 <- lm(G3 ~ age + Walc + health + absences + G1, data=notas_m[,!(names(notas_m) %in% c("G2", "cali
summary(fit41)
```

```
##
## Call:
## lm(formula = G3 ~ age + Walc + health + absences + G1, data = notas_m[,
##     !(names(notas_m) %in% c("G2", "calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3174 -0.9818  0.0025  0.9427  4.1708
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.064707   1.041877   5.821 1.34e-08 ***
## age            -0.216587   0.060681  -3.569 0.000409 ***
## Walcpoco       -0.477485   0.203232  -2.349 0.019364 *
## Walcalgo       -0.354067   0.201264  -1.759 0.079427 .
## Walcsuficiente -0.369331   0.240537  -1.535 0.125592
## Walcmucho       0.218712   0.301833   0.725 0.469182
## healthmal      -0.166211   0.305415  -0.544 0.586645
## healthregular  -0.507881   0.258081  -1.968 0.049879 *
## healthbien     -0.316989   0.276457  -1.147 0.252336
## healthmuy bien -0.329674   0.240348  -1.372 0.171063
## absences       -0.035032   0.009313  -3.761 0.000198 ***
## G1              0.870143   0.023293  37.357  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.382 on 345 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8167
## F-statistic: 145.2 on 11 and 345 DF,  p-value: < 2.2e-16
```

Comparación de ambos modelos:

```
anova(fit4,fit41)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences + G1
## Model 2: G3 ~ age + Walc + health + absences + G1
##   Res.Df    RSS  Df Sum of Sq     F Pr(>F)
## 1    286 559.33
## 2    345 659.01 -59   -99.688 0.864 0.7472
```

En este caso también ya sí se podrían considerar equivalentes el modelo completo y el modelo simplificado.

Escenario 3: con G1 y G2 En este escenario ya se consideran todas las variables (excepto la binaria creada por nosotros).

Asignatura: portugués Ajuste de la nota final a todas las variables.

```
fit5 <- lm(G3 ~ ., data=notas_p[,!(names(notas_p) %in% c("calificacion"))])
summary(fit5)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = notas_p[, !(names(notas_p) %in% c("calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4081 -0.5229 -0.0482  0.4680  5.3166
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.292210   0.845023  -0.346  0.72962
## schoolMS                 -0.097211   0.091109  -1.067  0.28644
## sexhombre                -0.099666   0.082529  -1.208  0.22769
## age                       0.099329   0.034428   2.885  0.00406 **
## addressRural             -0.018831   0.086189  -0.218  0.82713
## famsizeLE3               -0.061851   0.079907  -0.774  0.43923
## Pstatusseparados         -0.060129   0.114493  -0.525  0.59967
## Medu<=4ºEP                0.038567   0.372995   0.103  0.91768
## Medu5ºEP-3ºESO            0.109814   0.375231   0.293  0.76989
## Medu4ºESO-2ºBachiller     0.109416   0.379716   0.288  0.77334
## Meduestudios superiores   0.143322   0.391878   0.366  0.71470
## Fedu<=4ºEP               -0.162997   0.348074  -0.468  0.63976
## Fedu5ºEP-3ºESO           -0.073089   0.350705  -0.208  0.83499
## Fedu4ºESO-2ºBachiller    -0.167810   0.356983  -0.470  0.63848
```

```
## Feduestudios superiores  -0.195403   0.367362  -0.532  0.59500
## Mjobhealth                0.075172   0.177532   0.423  0.67215
## Mjobother                 0.122187   0.099541   1.227  0.22015
## Mjobservices              0.102363   0.121874   0.840  0.40132
## Mjobteacher               0.252254   0.171915   1.467  0.14285
## Fjobhealth               -0.357631   0.245364  -1.458  0.14552
## Fjobother                -0.372733   0.150736  -2.473  0.01370 *
## Fjobservices             -0.384884   0.158360  -2.430  0.01539 *
## Fjobteacher              -0.304499   0.226592  -1.344  0.17955
## reasonhome                0.068394   0.093025   0.735  0.46251
## reasonother              -0.132973   0.120886  -1.100  0.27181
## reasonreputation         -0.042981   0.096816  -0.444  0.65725
## guardianmother           -0.084384   0.087099  -0.969  0.33305
## guardianother            -0.072514   0.177136  -0.409  0.68243
## traveltime15-30 min       0.088406   0.081265   1.088  0.27712
## traveltime30 min.-1 hora  0.176315   0.140516   1.255  0.21008
## traveltime>1 hora         0.540288   0.236488   2.285  0.02271 *
## studytime2-5 horas        0.145647   0.087843   1.658  0.09787 .
## studytime5-10 horas       0.030577   0.119589   0.256  0.79829
## studytime>10 horas        0.016444   0.169847   0.097  0.92291
## failures1                -0.206609   0.131908  -1.566  0.11784
## failures2                -0.088263   0.249977  -0.353  0.72416
## failures>=3              -0.601617   0.259552  -2.318  0.02081 *
## schoolsupyes             -0.224996   0.120957  -1.860  0.06339 .
## famsupyes                 0.010090   0.074915   0.135  0.89291
## paidyes                  -0.134590   0.151059  -0.891  0.37332
## activitiesyes            -0.029586   0.073286  -0.404  0.68659
## nurseryyes               -0.055371   0.089148  -0.621  0.53478
## higheryes                 0.259793   0.130106   1.997  0.04633 *
## internetyes               0.130914   0.091490   1.431  0.15301
## romanticyes               0.018705   0.075713   0.247  0.80496
## famrelmal                 0.419264   0.263554   1.591  0.11222
## famrelregular             0.130224   0.220206   0.591  0.55451
## famrelbien                0.228864   0.206821   1.107  0.26895
## famrelmuy bien            0.186119   0.211469   0.880  0.37917
## freetimepoco              0.396646   0.159177   2.492  0.01300 *
## freetimealgo              0.171217   0.145834   1.174  0.24087
## freetimesuficiente        0.149815   0.155779   0.962  0.33661
## freetimemucho             0.219532   0.180593   1.216  0.22464
## gooutpoco                 0.050407   0.153427   0.329  0.74263
## gooutalgo                 0.016283   0.150222   0.108  0.91372
## gooutsuficiente          -0.008833   0.159145  -0.056  0.95576
## gooutmucho               -0.147787   0.169255  -0.873  0.38295
## Dalcpoco                 -0.219292   0.106329  -2.062  0.03963 *
## Dalcalgo                 -0.039617   0.168533  -0.235  0.81424
## Dalcsuficiente            0.385485   0.266888   1.444  0.14919
## Dalcmucho                -0.204391   0.276500  -0.739  0.46009
## Walcpoco                 -0.022929   0.097288  -0.236  0.81376
## Walcalgo                  0.093018   0.112493   0.827  0.40866
## Walcsuficiente            0.005427   0.139855   0.039  0.96906
## Walcmucho                 0.061112   0.211699   0.289  0.77294
## healthmal                -0.063757   0.138741  -0.460  0.64602
## healthregular            -0.102180   0.124714  -0.819  0.41295
## healthbien               -0.042837   0.128183  -0.334  0.73837
```

```
## healthmuy bien           -0.155800   0.114517  -1.360  0.17422
## absences                 -0.014606   0.008221  -1.777  0.07617 .
## G1                        0.180647   0.028618   6.312 5.58e-10 ***
## G2                        0.740175   0.029405  25.172  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8383 on 561 degrees of freedom
## Multiple R-squared:  0.9116, Adjusted R-squared:  0.9004
## F-statistic: 81.51 on 71 and 561 DF,  p-value: < 2.2e-16
```

Resultan significativas las siguientes variables: age, Fjob, traveltime, failures, higher, freetime, Dalc, G1 y G2.

El coeficiente de determinación ya solo ha aumentado unas décimas comparado con el del segundo escenario, es decir, solo considerando G1 y no G2. Esto se puede deber a que al igual que G1 y G2 están relacionadas con G3, G1 y G2 también están relacionadas. La información extra que aporta G2 estando ya incluida en el modelo G1 es mínima.

```
R_Cuadrado_p<-c(R_Cuadrado_p, 0.9116)
```

Ajustando ahora el modelo a solo las variables significativas quedaría lo siguiente:

```
fit51 <- lm(G3 ~ age + Fjob + traveltime + failures + higher + freetime + Dalc + G1 + G2, data=notas_p[
summary(fit51)
```

```
##
## Call:
## lm(formula = G3 ~ age + Fjob + traveltime + failures + higher +
##     freetime + Dalc + G1 + G2, data = notas_p[, !(names(notas_p) %in%
##     c("calificacion"))])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3765 -0.4742 -0.0750  0.5307  5.3440
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.77754    0.56958  -1.365 0.172724
## age                       0.10666    0.03045   3.502 0.000495 ***
## Fjobhealth               -0.36910    0.21993  -1.678 0.093809 .
## Fjobother                -0.30557    0.13928  -2.194 0.028623 *
## Fjobservices             -0.30439    0.14682  -2.073 0.038573 *
## Fjobteacher              -0.23211    0.19607  -1.184 0.236948
## traveltime15-30 min       0.05610    0.07435   0.755 0.450836
## traveltime30 min.-1 hora  0.08642    0.12653   0.683 0.494844
## traveltime>1 hora         0.38863    0.21765   1.786 0.074665 .
## failures1                -0.24336    0.12386  -1.965 0.049888 *
## failures2                -0.10280    0.23132  -0.444 0.656909
## failures>=3              -0.66083    0.24554  -2.691 0.007312 **
## higheryes                 0.31347    0.12197   2.570 0.010407 *
## freetimepoco              0.37168    0.15021   2.474 0.013619 *
## freetimealgo              0.19173    0.13588   1.411 0.158722
## freetimesuficiente        0.15013    0.14091   1.065 0.287080
## freetimemucho             0.17107    0.16413   1.042 0.297685
## Dalcpoco                 -0.22101    0.08756  -2.524 0.011851 *
## Dalcalgo                 -0.12085    0.13689  -0.883 0.377682
```

```
## Dalcsuficiente            0.30975    0.23829   1.300 0.194134
## Dalcmucho                -0.29431    0.21454  -1.372 0.170620
## G1                        0.19921    0.02702   7.372 5.51e-13 ***
## G2                        0.74604    0.02767  26.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8307 on 610 degrees of freedom
## Multiple R-squared:  0.9056, Adjusted R-squared:  0.9022
## F-statistic: 266.1 on 22 and 610 DF,  p-value: < 2.2e-16
```

Comparación de ambos modelos:

```
anova(fit5,fit51)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences + G1 + G2
## Model 2: G3 ~ age + Fjob + traveltime + failures + higher + freetime +
##     Dalc + G1 + G2
##   Res.Df    RSS  Df Sum of Sq      F Pr(>F)
## 1    561 394.26
## 2    610 420.96 -49   -26.697 0.7753 0.8655
```

Se pueden considerar equivalentes el modelo completo y el modelo simplificado.

Asignatura: matemáticas Ajuste de la nota final de matemáticas a todas las variables.

```
fit6 <- lm(G3 ~ ., data=notas_m[,!(names(notas_m) %in% c("calificacion"))])
summary(fit6)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = notas_m[, !(names(notas_m) %in% c("calificacion"))])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24764 -0.46470 -0.00082  0.45852  2.15637
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -0.699034   1.368595  -0.511  0.60991
## schoolMS                  -0.039679   0.176552  -0.225  0.82234
## sexhombre                 -0.066960   0.110953  -0.603  0.54666
## age                        0.044362   0.049400   0.898  0.36994
## addressRural               0.177742   0.130428   1.363  0.17404
## famsizeLE3                -0.080893   0.107678  -0.751  0.45312
## Pstatusseparados          -0.143223   0.160878  -0.890  0.37408
## Medu<=4ºEP                -0.077731   0.534944  -0.145  0.88457
## Medu5ºEP-3ºESO            -0.089199   0.535682  -0.167  0.86787
## Medu4ºESO-2ºBachiller     -0.225721   0.541971  -0.416  0.67737
## Meduestudios superiores   -0.201095   0.555057  -0.362  0.71740
```

```
## Fedu<=4ºEP                0.184211   0.629975   0.292  0.77019
## Fedu5ºEP-3ºESO            0.176001   0.630374   0.279  0.78029
## Fedu4ºESO-2ºBachiller     0.126846   0.631655   0.201  0.84099
## Feduestudios superiores   0.132764   0.642304   0.207  0.83639
## Mjobhealth                0.228857   0.250448   0.914  0.36160
## Mjobother                -0.165686   0.165102  -1.004  0.31645
## Mjobservices              0.046537   0.184287   0.253  0.80082
## Mjobteacher               0.209359   0.235718   0.888  0.37520
## Fjobhealth                0.248480   0.314538   0.790  0.43019
## Fjobother                 0.369483   0.232646   1.588  0.11336
## Fjobservices              0.261244   0.240846   1.085  0.27897
## Fjobteacher               0.362471   0.300895   1.205  0.22934
## reasonhome                0.191445   0.124217   1.541  0.12437
## reasonother               0.006599   0.174689   0.038  0.96989
## reasonreputation          0.139016   0.127832   1.087  0.27774
## guardianmother            0.004530   0.118374   0.038  0.96950
## guardianother            -0.280754   0.231861  -1.211  0.22695
## traveltime15-30 min      -0.084424   0.113080  -0.747  0.45593
## traveltime30 min.-1 hora -0.131452   0.225290  -0.583  0.56003
## traveltime>1 hora         0.795689   0.385143   2.066  0.03974 *
## studytime2-5 horas        0.002333   0.123140   0.019  0.98490
## studytime5-10 horas       0.008945   0.171646   0.052  0.95847
## studytime>10 horas        0.126222   0.223133   0.566  0.57206
## failures1                 0.138046   0.169386   0.815  0.41577
## failures2                -0.301259   0.286957  -1.050  0.29468
## failures>=3               0.227768   0.313193   0.727  0.46767
## schoolsupyes             -0.174614   0.147956  -1.180  0.23892
## famsupyes                 0.102718   0.104986   0.978  0.32871
## paidyes                  -0.118914   0.106153  -1.120  0.26357
## activitiesyes            -0.048652   0.099517  -0.489  0.62530
## nurseryyes               -0.204172   0.123514  -1.653  0.09943 .
## higheryes                 0.033668   0.278057   0.121  0.90371
## internetyes               0.033896   0.136248   0.249  0.80371
## romanticyes               0.068558   0.108581   0.631  0.52828
## famrelmal                 0.460788   0.422245   1.091  0.27607
## famrelregular             0.589290   0.363980   1.619  0.10655
## famrelbien                0.495473   0.352903   1.404  0.16141
## famrelmuy bien            0.878395   0.358511   2.450  0.01488 *
## freetimepoco              0.150180   0.254064   0.591  0.55491
## freetimealgo              0.171263   0.244130   0.702  0.48355
## freetimesuficiente        0.080136   0.252437   0.317  0.75113
## freetimemucho             0.115726   0.289887   0.399  0.69004
## gooutpoco                -0.293540   0.230687  -1.272  0.20425
## gooutalgo                -0.350775   0.230521  -1.522  0.12920
## gooutsuficiente          -0.516133   0.243790  -2.117  0.03512 *
## gooutmucho               -0.595394   0.261909  -2.273  0.02375 *
## Dalcpoco                  0.046053   0.145604   0.316  0.75201
## Dalcalgo                  0.293875   0.224455   1.309  0.19149
## Dalcsuficiente            0.084159   0.341137   0.247  0.80532
## Dalcmucho                -0.511723   0.396709  -1.290  0.19812
## Walcpoco                 -0.139240   0.137674  -1.011  0.31269
## Walcalgo                 -0.138412   0.149741  -0.924  0.35609
## Walcsuficiente           -0.133018   0.193028  -0.689  0.49131
## Walcmucho                 0.122859   0.307793   0.399  0.69007
```

```
## healthmal                 0.159152   0.198831   0.800  0.42412
## healthregular            -0.124628   0.174226  -0.715  0.47500
## healthbien               -0.027882   0.184633  -0.151  0.88007
## healthmuy bien           -0.250124   0.162201  -1.542  0.12417
## absences                 -0.010786   0.006569  -1.642  0.10171
## G1                        0.096449   0.036709   2.627  0.00907 **
## G2                        0.885391   0.037795  23.426  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.819 on 285 degrees of freedom
## Multiple R-squared:  0.9485, Adjusted R-squared:  0.9356
## F-statistic: 73.86 on 71 and 285 DF,  p-value: < 2.2e-16
```

Resultan significativas las siguientes variables: traveltime, famrel, goout, G1 y G2.

El coeficiente de determinación al igual que en el caso de la asignatura de portugués aumenta ligeramente pasando a 0.95.

```
R_Cuadrado_m<-c(R_Cuadrado_m, 0.9485)
```

Ajustando ahora el modelo a solo las variables significativas quedaría lo siguiente:

```
fit61 <- lm(G3 ~ traveltime + famrel + goout + G1 + G2, data=notas_m[,!(names(notas_m) %in% c("califica
summary(fit61)
```

```
##
## Call:
## lm(formula = G3 ~ traveltime + famrel + goout + G1 + G2, data = notas_m[,
##     !(names(notas_m) %in% c("calificacion"))])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.55694 -0.45934 -0.02275  0.54382  2.25982
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -0.007752   0.403512  -0.019 0.984683
## traveltime15-30 min       -0.117394   0.098178  -1.196 0.232630
## traveltime30 min.-1 hora  -0.103421   0.193578  -0.534 0.593507
## traveltime>1 hora          0.877178   0.322777   2.718 0.006910 **
## famrelmal                  0.315978   0.375841   0.841 0.401089
## famrelregular              0.502834   0.326127   1.542 0.124036
## famrelbien                 0.376129   0.314661   1.195 0.232777
## famrelmuy bien             0.780654   0.319057   2.447 0.014916 *
## gooutpoco                 -0.183243   0.205097  -0.893 0.372246
## gooutalgo                 -0.191404   0.200248  -0.956 0.339830
## gooutsuficiente           -0.352480   0.207190  -1.701 0.089804 .
## gooutmucho                -0.460516   0.225058  -2.046 0.041498 *
## G1                         0.110482   0.030704   3.598 0.000367 ***
## G2                         0.885428   0.031761  27.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8019 on 343 degrees of freedom
## Multiple R-squared:  0.9405, Adjusted R-squared:  0.9383
```

```
## F-statistic: 417.3 on 13 and 343 DF,  p-value: < 2.2e-16
```

Comparación de ambos modelos:

```
anova(fit6,fit61)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ school + sex + age + address + famsize + Pstatus + Medu +
##     Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##     failures + schoolsup + famsup + paid + activities + nursery +
##     higher + internet + romantic + famrel + freetime + goout +
##     Dalc + Walc + health + absences + G1 + G2
## Model 2: G3 ~ traveltime + famrel + goout + G1 + G2
##   Res.Df    RSS  Df Sum of Sq      F Pr(>F)
## 1    285 191.19
## 2    343 220.56 -58   -29.369 0.7548 0.9015
```

Se pueden considerar equivalentes el modelo completo y el modelo simplificado.

```
R_Cuadrado<-rbind(R_Cuadrado_p, R_Cuadrado_m)
colnames(R_Cuadrado)<-c("Sin G1 y G2", "Con G1 y sin G2", "Con G1 y G2")
rownames(R_Cuadrado)<-c("Portugués", "Matemáticas")
```

Predicción de la nota final de forma binaria (aprobado, suspenso)

Se realizarán a continuación los análisis considerando también las asignaturas separadas y los tres escenarios ya mencionados.

Se utilizarán distintos modelos de predicción para comparar y elegir el más adecuado.

Antes de continuar, se escalan las variables númericas y se recondicionan las variables categóricas en variables ficticias.

```
Classes=sapply(notas_p,class)
for(i in 1:ncol(notas_p))
  if(Classes[i]=='numeric')
  notas_p[,i]=scale(notas_p[,i])
head(notas_p)
```

```
##   school    sex        age address famsize   Pstatus                Medu
## 1     GP  mujer  1.0540149   Rural     GT3    juntos estudios superiores
## 2     GP  mujer  0.2303218   Rural     GT3 separados              <=4ºEP
## 3     GP  mujer -1.4170645   Rural     LE3 separados              <=4ºEP
## 4     GP  mujer -1.4170645   Rural     GT3 separados estudios superiores
## 5     GP  mujer -0.5933714   Rural     GT3 separados   4ºESO-2ºBachiller
## 6     GP hombre -0.5933714   Rural     LE3 separados estudios superiores
##                  Fedu     Mjob     Fjob     reason guardian traveltime
## 1 estudios superiores  at_home  teacher     course   mother  15-30 min
## 2             <=4ºEP  at_home    other     course   father    <15 min
## 3             <=4ºEP  at_home    other      other   mother    <15 min
## 4          5ºEP-3ºESO   health services       home   mother    <15 min
## 5   4ºESO-2ºBachiller    other    other       home   father    <15 min
## 6   4ºESO-2ºBachiller services    other reputation   mother    <15 min
##    studytime failures schoolsup famsup paid activities nursery higher internet
## 1  2-5 horas        0       yes     no   no         no     yes    yes       no
## 2  2-5 horas        0        no    yes   no         no      no    yes      yes
## 3  2-5 horas        0       yes     no   no         no     yes    yes      yes
```

```
## 4 5-10 horas        0        no    yes   no        yes     yes    yes      yes
## 5  2-5 horas        0        no    yes   no         no     yes    yes       no
## 6  2-5 horas        0        no    yes   no        yes     yes    yes      yes
##   romantic   famrel   freetime      goout Dalc Walc   health    absences
## 1       no     bien       algo suficiente nada nada  regular  0.05320144
## 2       no muy bien       algo       algo nada nada  regular -0.37579871
## 3       no     bien       algo       poco poco algo  regular  0.48220159
## 4      yes  regular       poco       poco nada nada muy bien -0.80479886
## 5       no     bien       algo       poco nada poco muy bien -0.80479886
## 6       no muy bien suficiente       poco nada poco muy bien  0.48220159
##           G1          G2          G3 calificacion
## 1 -4.2882867 -0.28775514 -0.45367889     aprobado
## 2 -0.9333884 -0.28775514 -0.45367889     aprobado
## 3  0.1849110  0.47278350 -0.07729784     aprobado
## 4  0.9304440  0.85305282  0.67546424     aprobado
## 5 -0.1878555  0.47278350  0.29908320     aprobado
## 6  0.1849110  0.09251418  0.29908320     aprobado
```

```
Classes=sapply(notas_m,class)
for(i in 1:ncol(notas_m))
  if(Classes[i]=='numeric')
  notas_m[,i]=scale(notas_m[,i])
head(notas_m)
```

```
##     school    sex        age address famsize   Pstatus                Medu
## 650     GP  mujer  1.0601421   Rural     GT3    juntos estudios superiores
## 651     GP  mujer  0.2716614   Rural     GT3 separados              <=4ºEP
## 652     GP  mujer -1.3052999   Rural     LE3 separados              <=4ºEP
## 653     GP  mujer -1.3052999   Rural     GT3 separados estudios superiores
## 654     GP  mujer -0.5168193   Rural     GT3 separados   4ºESO-2ºBachiller
## 655     GP hombre -0.5168193   Rural     LE3 separados estudios superiores
##                     Fedu     Mjob     Fjob     reason guardian traveltime
## 650 estudios superiores  at_home  teacher     course   mother  15-30 min
## 651              <=4ºEP  at_home    other     course   father    <15 min
## 652              <=4ºEP  at_home    other      other   mother    <15 min
## 653          5ºEP-3ºESO   health services       home   mother    <15 min
## 654   4ºESO-2ºBachiller    other    other       home   father    <15 min
## 655   4ºESO-2ºBachiller services    other reputation   mother    <15 min
##      studytime failures schoolsup famsup paid activities nursery higher
## 650  2-5 horas        0       yes     no   no         no     yes    yes
## 651  2-5 horas        0        no    yes   no         no      no    yes
## 652  2-5 horas      >=3       yes     no  yes         no     yes    yes
## 653 5-10 horas        0        no    yes  yes        yes     yes    yes
## 654  2-5 horas        0        no    yes  yes         no     yes    yes
## 655  2-5 horas        0        no    yes  yes        yes     yes    yes
##     internet romantic   famrel   freetime      goout Dalc Walc   health
## 650       no       no     bien       algo suficiente nada nada  regular
## 651      yes       no muy bien       algo       algo nada nada  regular
## 652      yes       no     bien       algo       poco poco algo  regular
## 653      yes      yes  regular       poco       poco nada nada muy bien
## 654       no       no     bien       algo       poco nada poco muy bien
## 655      yes       no muy bien suficiente       poco nada poco muy bien
##        absences        G1         G2         G3 calificacion
## 650 -0.03865916 -1.934579 -1.7026448 -1.7113250     suspenso
## 651 -0.28293030 -1.934579 -2.0203887 -1.7113250     suspenso
```

```
## 652  0.44988312 -1.317381 -1.0671569 -0.4720896     aprobado
## 653 -0.52720144  1.151412  0.8393068  1.0769545     aprobado
## 654 -0.28293030 -1.625980 -0.4316690 -0.4720896     aprobado
## 655  0.44988312  1.151412  1.1570508  1.0769545     aprobado
```

El conjunto de individuos de cada asignatura será divido a continuación en dos partes: una para entrenar el modelo que será el 70% y otra para validarlo con el 30% restante. La semilla utilizada será el número 2022. Se crearán tres conjuntos de entrenamiento y validación para cada asignatura: uno para el escenario 1 (sin G1 y G2), otro para el escenario 2 (con G1 y sin G2) y otro para el escenario 3 (con G1 y G2). En ninguno se incluirá tampoco la variable G3, correspondiente a la nota final numérica.

```
tr=round(nrow(notas_p)*0.7)
set.seed(2022)
muestra_p=sample.int(nrow(notas_p), tr)
Train1.notas_p=notas_p[muestra_p,!(names(notas_p) %in% c("G1","G2","G3"))]
Val1.notas_p=notas_p[-muestra_p,!(names(notas_p) %in% c("G1","G2","G3"))]
Train2.notas_p=notas_p[muestra_p,!(names(notas_p) %in% c("G2","G3"))]
Val2.notas_p=notas_p[-muestra_p,!(names(notas_p) %in% c("G2","G3"))]
Train3.notas_p=notas_p[muestra_p,!(names(notas_p) %in% c("G3"))]
Val3.notas_p=notas_p[-muestra_p,!(names(notas_p) %in% c("G3"))]

tr=round(nrow(notas_m)*0.7)
set.seed(2022)
muestra_m=sample.int(nrow(notas_m), tr)
Train1.notas_m=notas_m[muestra_m,!(names(notas_m) %in% c("G1","G2","G3"))]
Val1.notas_m=notas_m[-muestra_m,!(names(notas_m) %in% c("G1","G2","G3"))]
Train2.notas_m=notas_m[muestra_m,!(names(notas_m) %in% c("G2","G3"))]
Val2.notas_m=notas_m[-muestra_m,!(names(notas_m) %in% c("G2","G3"))]
Train3.notas_m=notas_m[muestra_m,!(names(notas_m) %in% c("G3"))]
Val3.notas_m=notas_m[-muestra_m,!(names(notas_m) %in% c("G3"))]
```

Escenario 1: sin G1 y G2

Método 1: Regresión logística Asignatura: portugués

Primero se ajusta al modelo completo.

```
gfit1=glm(calificacion~., data=notas_p[,!(names(notas_p) %in% c("G1","G2", "G3"))], family=binomial)
```

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

```
summary(gfit1)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_p[,
##     !(names(notas_p) %in% c("G1", "G2", "G3"))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0259  -0.3575  -0.1548  -0.0518   3.2740
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -29.80062 1782.68850  -0.017 0.986663
## schoolMS                   2.48380    0.46775   5.310  1.1e-07 ***
```

```
## sexhombre                   0.79809    0.42583   1.874 0.060901 .
## age                        -0.43721    0.21665  -2.018 0.043582 *
## addressRural               -0.16198    0.39472  -0.410 0.681543
## famsizeLE3                 -0.18719    0.41075  -0.456 0.648583
## Pstatusseparados            0.04696    0.57906   0.081 0.935370
## Medu<=4ºEP                 15.11437 1269.93269   0.012 0.990504
## Medu5ºEP-3ºESO             15.00313 1269.93277   0.012 0.990574
## Medu4ºESO-2ºBachiller      15.42723 1269.93277   0.012 0.990307
## Meduestudios superiores    14.36273 1269.93305   0.011 0.990976
## Fedu<=4ºEP                 15.76502 1251.09831   0.013 0.989946
## Fedu5ºEP-3ºESO             14.61874 1251.09836   0.012 0.990677
## Fedu4ºESO-2ºBachiller      14.19634 1251.09843   0.011 0.990947
## Feduestudios superiores    15.39887 1251.09847   0.012 0.990180
## Mjobhealth                  0.72101    0.82159   0.878 0.380171
## Mjobother                  -0.35911    0.43166  -0.832 0.405449
## Mjobservices               -0.14561    0.55567  -0.262 0.793284
## Mjobteacher                -1.74917    1.16185  -1.506 0.132192
## Fjobhealth                  1.09860    1.16644   0.942 0.346272
## Fjobother                   0.26348    0.68611   0.384 0.700968
## Fjobservices                0.72904    0.71942   1.013 0.310881
## Fjobteacher                 1.28247    1.28967   0.994 0.320019
## reasonhome                 -0.49454    0.50461  -0.980 0.327060
## reasonother                -0.39684    0.56154  -0.707 0.479761
## reasonreputation           -0.58990    0.57581  -1.024 0.305619
## guardianmother              0.98370    0.45693   2.153 0.031331 *
## guardianother               0.70486    0.88622   0.795 0.426408
## traveltime15-30 min        -1.11565    0.43426  -2.569 0.010196 *
## traveltime30 min.-1 hora   -0.94388    0.59898  -1.576 0.115069
## traveltime>1 hora          -0.27814    0.94751  -0.294 0.769101
## studytime2-5 horas         -0.20618    0.39554  -0.521 0.602185
## studytime5-10 horas        -0.10641    0.69259  -0.154 0.877894
## studytime>10 horas         -1.46396    1.11906  -1.308 0.190802
## failures1                   1.75607    0.47348   3.709 0.000208 ***
## failures2                   3.10994    0.88367   3.519 0.000433 ***
## failures>=3                 2.90180    0.86382   3.359 0.000782 ***
## schoolsupyes                1.08873    0.62588   1.740 0.081943 .
## famsupyes                  -0.01757    0.36878  -0.048 0.962002
## paidyes                     1.24536    0.60506   2.058 0.039569 *
## activitiesyes              -0.81101    0.36885  -2.199 0.027893 *
## nurseryyes                  0.36047    0.42647   0.845 0.397975
## higheryes                  -1.92002    0.46612  -4.119  3.8e-05 ***
## internetyes                 0.49959    0.43651   1.144 0.252419
## romanticyes                -0.16911    0.36389  -0.465 0.642122
## famrelmal                  -2.81759    1.10625  -2.547 0.010866 *
## famrelregular              -1.89244    0.88210  -2.145 0.031922 *
## famrelbien                 -2.50838    0.82846  -3.028 0.002464 **
## famrelmuy bien             -1.69506    0.83386  -2.033 0.042074 *
## freetimepoco               -0.68669    0.76053  -0.903 0.366574
## freetimealgo               -0.02870    0.68087  -0.042 0.966380
## freetimesuficiente         -0.49454    0.72612  -0.681 0.495820
## freetimemucho              -0.01828    0.76283  -0.024 0.980886
## gooutpoco                  -0.94854    0.72855  -1.302 0.192932
## gooutalgo                  -1.29157    0.70918  -1.821 0.068575 .
## gooutsuficiente            -0.51402    0.74227  -0.693 0.488619
```

```
## gooutmucho                 -0.26475    0.75812  -0.349 0.726922
## Dalcpoco                    0.02067    0.50479   0.041 0.967330
## Dalcalgo                   -1.08988    0.73637  -1.480 0.138854
## Dalcsuficiente             -2.16625    1.64644  -1.316 0.188270
## Dalcmucho                  -0.61482    1.15760  -0.531 0.595337
## Walcpoco                    0.25758    0.49853   0.517 0.605377
## Walcalgo                    0.52657    0.55359   0.951 0.341506
## Walcsuficiente              0.93457    0.64211   1.455 0.145539
## Walcmucho                   1.03181    0.96376   1.071 0.284341
## healthmal                  -1.48868    0.71210  -2.091 0.036569 *
## healthregular              -0.68029    0.59662  -1.140 0.254182
## healthbien                 -0.93636    0.68244  -1.372 0.170040
## healthmuy bien             -0.41758    0.51883  -0.805 0.420900
## absences                    0.58824    0.16977   3.465 0.000530 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 495.63  on 632  degrees of freedom
## Residual deviance: 281.76  on 563  degrees of freedom
## AIC: 421.76
##
## Number of Fisher Scoring iterations: 16
```

Resultan significativas las siguientes variables: school, age, guardian, traveltime, failures, paid, activities, higher, famrel, health y absences.

Sin embargo, para el aprendizaje automático lo que interesa es la predicción.

```
gfit12=glm(calificacion~., data=Train1.notas_p, family=binomial)
cbind(gfit1$coefficients, gfit12$coefficients)
```

```
##                                 [,1]         [,2]
## (Intercept)              -29.80061971 -24.98103040
## schoolMS                   2.48379650   2.77517697
## sexhombre                  0.79809361   0.78823221
## age                       -0.43720929  -0.48780648
## addressRural              -0.16197570  -0.38579568
## famsizeLE3                -0.18719096  -0.27238414
## Pstatusseparados           0.04695625  -0.11131619
## Medu<=4ºEP                15.11437316  11.58471544
## Medu5ºEP-3ºESO            15.00313352  11.03378413
## Medu4ºESO-2ºBachiller     15.42722528  11.61982076
## Meduestudios superiores   14.36273202  10.54920609
## Fedu<=4ºEP                15.76501684  15.89618574
## Fedu5ºEP-3ºESO            14.61873926  14.74012108
## Fedu4ºESO-2ºBachiller     14.19634327  13.64423002
## Feduestudios superiores   15.39887360  14.66926269
## Mjobhealth                 0.72101098   1.22644507
## Mjobother                 -0.35911232   0.16803632
## Mjobservices              -0.14561269  -0.06239277
## Mjobteacher               -1.74917480  -2.33178610
## Fjobhealth                 1.09860480   0.99116376
## Fjobother                  0.26347530  -0.16785477
## Fjobservices               0.72904187   0.91242380
```

```
## Fjobteacher                1.28247110   1.99693598
## reasonhome                -0.49454263  -0.77112496
## reasonother               -0.39683662   0.45952139
## reasonreputation          -0.58989555  -1.08866723
## guardianmother             0.98370017   1.01471622
## guardianother              0.70485564   0.37632401
## traveltime15-30 min       -1.11564994  -1.49226706
## traveltime30 min.-1 hora  -0.94387781  -0.81166582
## traveltime>1 hora         -0.27814254  -0.02506285
## studytime2-5 horas        -0.20617720  -0.24854543
## studytime5-10 horas       -0.10640874  -1.19512640
## studytime>10 horas        -1.46396095  -2.79399144
## failures1                  1.75606678   1.40071831
## failures2                  3.10994382   3.51788705
## failures>=3                2.90180083   1.66235699
## schoolsupyes               1.08873130   0.75137969
## famsupyes                 -0.01756958   0.16443327
## paidyes                    1.24535813   2.64171894
## activitiesyes             -0.81101213  -0.69400972
## nurseryyes                 0.36047048   0.77270902
## higheryes                 -1.92002089  -2.09654085
## internetyes                0.49958652   0.61593978
## romanticyes               -0.16911222  -0.23194101
## famrelmal                 -2.81759051  -4.51222014
## famrelregular             -1.89244269  -2.28783174
## famrelbien                -2.50837779  -3.14382972
## famrelmuy bien            -1.69505752  -2.31112041
## freetimepoco              -0.68668726  -0.06097995
## freetimealgo              -0.02869768   0.40593348
## freetimesuficiente        -0.49454411   0.28103869
## freetimemucho             -0.01827572   0.25524973
## gooutpoco                 -0.94853841  -1.90570850
## gooutalgo                 -1.29157373  -1.84455250
## gooutsuficiente           -0.51402331  -0.83542680
## gooutmucho                -0.26475142  -0.92574473
## Dalcpoco                   0.02067481  -0.43027570
## Dalcalgo                  -1.08987510  -1.44995266
## Dalcsuficiente            -2.16624753  -2.36213644
## Dalcmucho                 -0.61482002  -1.62614863
## Walcpoco                   0.25758092  -0.77683627
## Walcalgo                   0.52656892   0.37569633
## Walcsuficiente             0.93456732   0.81184548
## Walcmucho                  1.03181480   1.49291918
## healthmal                 -1.48868209  -1.67326196
## healthregular             -0.68029209  -0.82180570
## healthbien                -0.93636117  -0.91983081
## healthmuy bien            -0.41758304  -0.94371124
## absences                   0.58823514   0.87477995
```

Con este modelo, predecimos los valores de calificación en la asignatura de portugués.

```
p=predict(gfit12, Val1.notas_p, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.5
```

```
## Loading required package: lattice
```

```
matrizLogis<-confusionMatrix(Val1.notas_p$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      152       14
##   suspenso       20        4
##
##                Accuracy : 0.8211
##                  95% CI : (0.759, 0.8728)
##     No Information Rate : 0.9053
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.0922
##
##  Mcnemar's Test P-Value : 0.3912
##
##             Sensitivity : 0.8837
##             Specificity : 0.2222
##          Pos Pred Value : 0.9157
##          Neg Pred Value : 0.1667
##              Prevalence : 0.9053
##          Detection Rate : 0.8000
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.5530
##
##        'Positive' Class : aprobado
##
```

El porcentaje de clasificación correcta es del 82%.

```
precision_p1<-c(matrizLogis$overall[1])
names(precision_p1)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC para comprobar el modelo.

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.0.5
```

```
## Type 'citation("pROC")' for a citation.
```

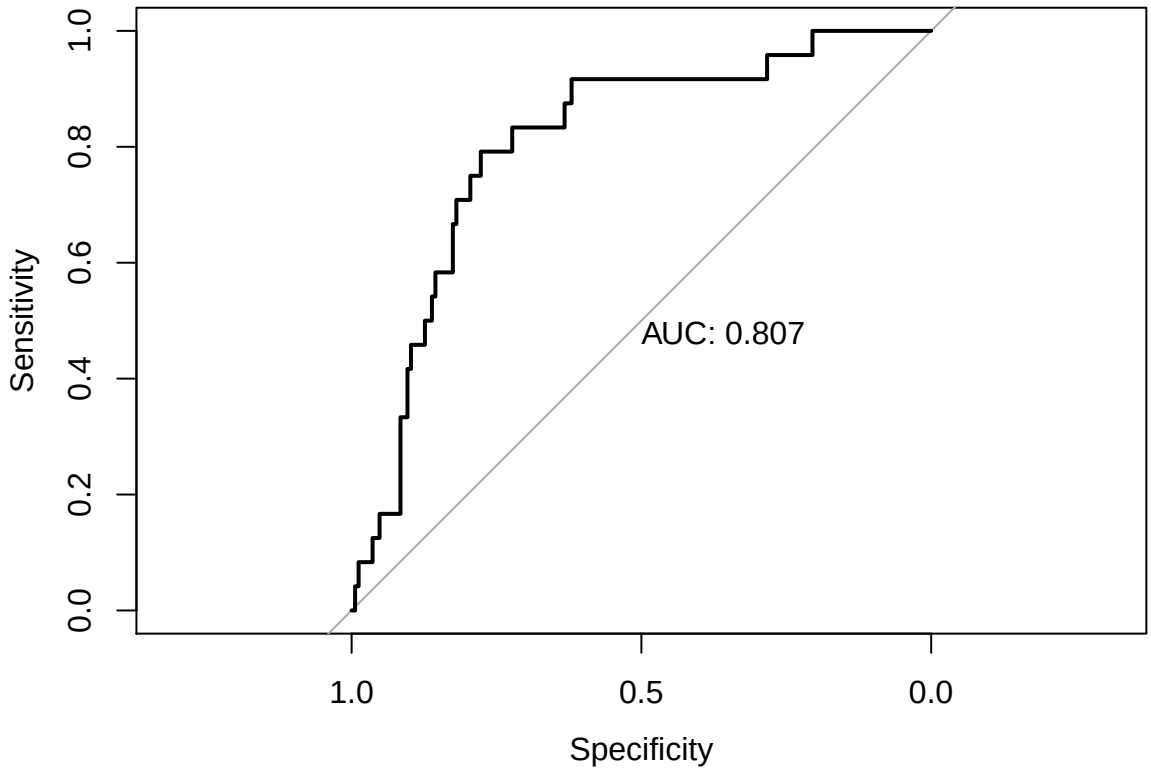
```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
test_prob = predict(gfit12, newdata = Val1.notas_p, type = "response")
test_roc = roc(Val1.notas_p$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0,807 que es un valor alto y por tanto confirma que el modelo es bueno.

Asignatura: matemáticas

Primero se ajusta al modelo completo.

```
gfit2=glm(calificacion~., data=notas_m[,!(names(notas_m) %in% c("G1","G2", "G3"))], family=binomial)
summary(gfit2)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_m[,
##     !(names(notas_m) %in% c("G1", "G2", "G3"))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1232  -0.6204  -0.3302   0.1316   2.4569
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.705e+01  1.022e+03  -0.017  0.98669
## schoolMS                 -6.057e-02  6.012e-01  -0.101  0.91974
## sexhombre                -3.209e-01  4.119e-01  -0.779  0.43595
## age                       3.862e-01  2.284e-01   1.691  0.09092 .
## addressRural             -5.589e-01  4.471e-01  -1.250  0.21127
## famsizeLE3               -2.390e-01  3.835e-01  -0.623  0.53309
## Pstatusseparados          5.993e-01  5.945e-01   1.008  0.31341
```

```
## Medu<=4ºEP               -5.644e-01  2.790e+00  -0.202  0.83967
## Medu5ºEP-3ºESO           -6.322e-01  2.808e+00  -0.225  0.82189
## Medu4ºESO-2ºBachiller    -3.741e-01  2.820e+00  -0.133  0.89447
## Meduestudios superiores  -6.626e-01  2.853e+00  -0.232  0.81636
## Fedu<=4ºEP                1.397e+01  1.022e+03   0.014  0.98909
## Fedu5ºEP-3ºESO            1.336e+01  1.022e+03   0.013  0.98957
## Fedu4ºESO-2ºBachiller     1.372e+01  1.022e+03   0.013  0.98929
## Feduestudios superiores   1.272e+01  1.022e+03   0.012  0.99007
## Mjobhealth               -3.716e-01  8.766e-01  -0.424  0.67160
## Mjobother                 7.659e-01  5.442e-01   1.407  0.15933
## Mjobservices             -3.047e-01  6.223e-01  -0.490  0.62442
## Mjobteacher               1.177e+00  8.057e-01   1.461  0.14398
## Fjobhealth                6.185e-01  1.012e+00   0.611  0.54092
## Fjobother                -6.056e-01  7.376e-01  -0.821  0.41166
## Fjobservices             -8.797e-02  7.731e-01  -0.114  0.90940
## Fjobteacher              -6.988e-01  1.061e+00  -0.659  0.51019
## reasonhome               -6.988e-01  4.474e-01  -1.562  0.11830
## reasonother               1.203e-02  6.064e-01   0.020  0.98417
## reasonreputation         -4.365e-01  4.567e-01  -0.956  0.33920
## guardianmother            3.873e-01  4.443e-01   0.872  0.38335
## guardianother            -4.175e-01  8.268e-01  -0.505  0.61362
## traveltime15-30 min       7.616e-02  3.942e-01   0.193  0.84683
## traveltime30 min.-1 hora -1.169e+00  8.292e-01  -1.410  0.15842
## traveltime>1 hora        -5.919e-01  1.489e+00  -0.397  0.69104
## studytime2-5 horas       -3.083e-01  4.359e-01  -0.707  0.47943
## studytime5-10 horas      -1.192e+00  6.563e-01  -1.816  0.06942 .
## studytime>10 horas       -1.016e+00  8.273e-01  -1.228  0.21942
## failures1                -3.794e-02  5.225e-01  -0.073  0.94211
## failures2                 3.564e+00  1.100e+00   3.241  0.00119 **
## failures>=3               1.829e+00  9.336e-01   1.960  0.05004 .
## schoolsupyes              1.585e+00  4.908e-01   3.230  0.00124 **
## famsupyes                 8.566e-01  3.809e-01   2.249  0.02453 *
## paidyes                  -2.115e-01  3.854e-01  -0.549  0.58306
## activitiesyes             9.453e-03  3.667e-01   0.026  0.97944
## nurseryyes                8.465e-01  4.808e-01   1.761  0.07830 .
## higheryes                -9.873e-01  9.192e-01  -1.074  0.28281
## internetyes              -2.585e-01  4.387e-01  -0.589  0.55578
## romanticyes              -2.688e-01  4.030e-01  -0.667  0.50484
## famrelmal                -4.834e-01  1.863e+00  -0.259  0.79530
## famrelregular             1.762e-01  1.693e+00   0.104  0.91707
## famrelbien               -3.748e-01  1.681e+00  -0.223  0.82355
## famrelmuy bien           -6.851e-01  1.700e+00  -0.403  0.68692
## freetimepoco             -3.660e-01  8.959e-01  -0.409  0.68287
## freetimealgo             -2.078e-01  8.270e-01  -0.251  0.80162
## freetimesuficiente        5.844e-02  8.556e-01   0.068  0.94554
## freetimemucho            -6.759e-01  1.006e+00  -0.672  0.50164
## gooutpoco                 2.089e+00  1.387e+00   1.506  0.13200
## gooutalgo                 3.204e+00  1.428e+00   2.244  0.02482 *
## gooutsuficiente           3.850e+00  1.474e+00   2.611  0.00903 **
## gooutmucho                3.848e+00  1.452e+00   2.651  0.00803 **
## Dalcpoco                 -1.680e-01  5.293e-01  -0.317  0.75095
## Dalcalgo                 -1.637e-01  7.568e-01  -0.216  0.82879
## Dalcsuficiente            1.773e+00  1.081e+00   1.640  0.10097
## Dalcmucho                -7.450e-01  1.430e+00  -0.521  0.60239
```

```
## Walcpoco                -4.667e-01  4.951e-01  -0.943  0.34589
## Walcalgo                -5.486e-01  5.406e-01  -1.015  0.31019
## Walcsuficiente          -5.107e-01  6.824e-01  -0.748  0.45421
## Walcmucho               -9.199e-01  1.045e+00  -0.880  0.37884
## healthmal                8.062e-01  7.769e-01   1.038  0.29940
## healthregular            1.157e+00  6.898e-01   1.677  0.09360 .
## healthbien               8.284e-01  7.035e-01   1.177  0.23900
## healthmuy bien           1.443e+00  6.702e-01   2.153  0.03134 *
## absences                 5.925e-01  1.905e-01   3.110  0.00187 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 407.44  on 356  degrees of freedom
## Residual deviance: 283.06  on 287  degrees of freedom
## AIC: 423.06
##
## Number of Fisher Scoring iterations: 14
```

Resultan significativas las siguientes variables: failures, schoolsup, famsup, goout, health y absences.

A continuación la predicción.

```
gfit21=glm(calificacion~., data=Train1.notas_m, family=binomial)
cbind(gfit1$coefficients, gfit21$coefficients)
```

```
##                                [,1]         [,2]
## (Intercept)              -29.80061971 -53.74894469
## schoolMS                   2.48379650  -0.79783162
## sexhombre                  0.79809361  -0.23482472
## age                       -0.43720929   0.54110823
## addressRural              -0.16197570  -0.38717269
## famsizeLE3                -0.18719096   0.07709239
## Pstatusseparados           0.04695625   1.74689206
## Medu<=4ºEP                15.11437316  13.61541493
## Medu5ºEP-3ºESO            15.00313352  13.47468951
## Medu4ºESO-2ºBachiller     15.42722528  14.29886687
## Meduestudios superiores   14.36273202  14.37908711
## Fedu<=4ºEP                15.76501684  15.89050485
## Fedu5ºEP-3ºESO            14.61873926  15.93514749
## Fedu4ºESO-2ºBachiller     14.19634327  15.52968513
## Feduestudios superiores   15.39887360  15.05717488
## Mjobhealth                 0.72101098  -1.92716629
## Mjobother                 -0.35911232   1.14158586
## Mjobservices              -0.14561269  -0.62578885
## Mjobteacher               -1.74917480  -0.02138570
## Fjobhealth                 1.09860480   0.65567904
## Fjobother                  0.26347530  -0.79136903
## Fjobservices               0.72904187  -0.55733722
## Fjobteacher                1.28247110  -1.35340343
## reasonhome                -0.49454263  -0.61779798
## reasonother               -0.39683662  -1.24956345
## reasonreputation          -0.58989555  -0.94080197
## guardianmother             0.98370017   0.74492547
## guardianother              0.70485564  -0.27962509
```

```
## traveltime15-30 min       -1.11564994   0.85370682
## traveltime30 min.-1 hora  -0.94387781  -0.15325478
## traveltime>1 hora         -0.27814254   0.90979840
## studytime2-5 horas        -0.20617720  -0.45320319
## studytime5-10 horas       -0.10640874  -1.05378013
## studytime>10 horas        -1.46396095  -1.13248135
## failures1                  1.75606678  -0.48888783
## failures2                  3.10994382   3.39774248
## failures>=3                2.90180083   3.29014084
## schoolsupyes               1.08873130   1.79046584
## famsupyes                 -0.01756958   0.95411297
## paidyes                    1.24535813   0.22926120
## activitiesyes             -0.81101213   0.56879777
## nurseryyes                 0.36047048   1.18615581
## higheryes                 -1.92002089  -2.60157880
## internetyes                0.49958652  -0.21496983
## romanticyes               -0.16911222  -0.04274016
## famrelmal                 -2.81759051  -1.90600219
## famrelregular             -1.89244269  -1.88313585
## famrelbien                -2.50837779  -1.73412410
## famrelmuy bien            -1.69505752  -3.03121398
## freetimepoco              -0.68668726  -1.04877638
## freetimealgo              -0.02869768  -0.99057589
## freetimesuficiente        -0.49454411  -0.41365103
## freetimemucho             -0.01827572  -1.18113113
## gooutpoco                 -0.94853841  22.84460219
## gooutalgo                 -1.29157373  23.99410180
## gooutsuficiente           -0.51402331  24.75084024
## gooutmucho                -0.26475142  24.29536210
## Dalcpoco                   0.02067481  -0.46037052
## Dalcalgo                  -1.08987510   0.36696983
## Dalcsuficiente            -2.16624753   5.15509549
## Dalcmucho                 -0.61482002   1.92204260
## Walcpoco                   0.25758092   0.01640357
## Walcalgo                   0.52656892  -0.04238608
## Walcsuficiente             0.93456732   0.08696835
## Walcmucho                  1.03181480  -3.28601578
## healthmal                 -1.48868209   1.50383570
## healthregular             -0.68029209   1.88662165
## healthbien                -0.93636117   2.18312754
## healthmuy bien            -0.41758304   1.75822332
## absences                   0.58823514   0.46732129
```

Con este modelo, predecimos los valores de calificación en la asignatura de matemáticas.

```
p=predict(gfit21, Val1.notas_m, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
library(caret)
matrizLogis<-confusionMatrix(Val1.notas_m$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction aprobado suspenso
##   aprobado       65       10
##   suspenso       21       11
##
##                Accuracy : 0.7103
##                  95% CI : (0.6146, 0.7939)
##     No Information Rate : 0.8037
##     P-Value [Acc > NIR] : 0.99280
##
##                   Kappa : 0.2334
##
##  Mcnemar's Test P-Value : 0.07249
##
##             Sensitivity : 0.7558
##             Specificity : 0.5238
##          Pos Pred Value : 0.8667
##          Neg Pred Value : 0.3438
##              Prevalence : 0.8037
##          Detection Rate : 0.6075
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.6398
##
##        'Positive' Class : aprobado
##
```

El porcentaje de clasificación correcta es del 71%, menor que en la asignatura de portugués.

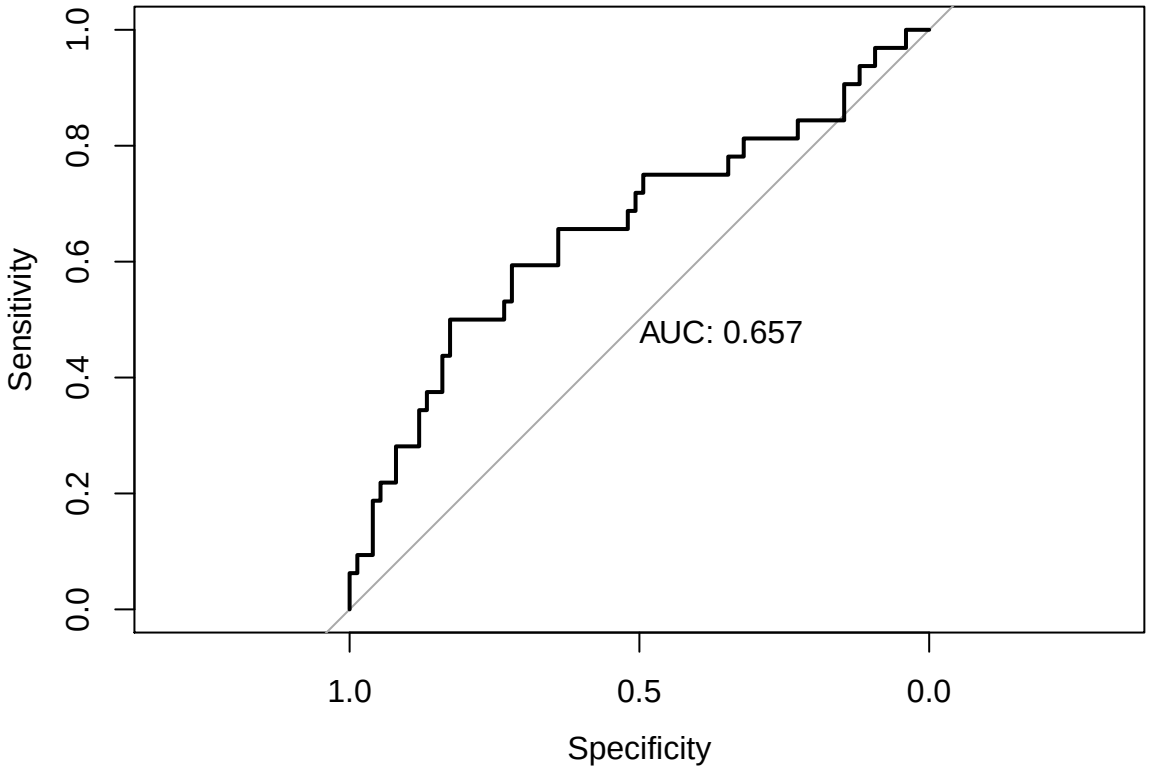
```
precision_m1<-c(matrizLogis$overall[1])
names(precision_m1)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC para comprobar el modelo.

```
library(pROC)
test_prob = predict(gfit21, newdata = Val1.notas_m, type = "response")
test_roc = roc(Val1.notas_m$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0,657. Este valor no es muy alto.

Método 2: Redes neuronales Asignatura: portugués

Se prueba primero con una red neuronal de una capa y cinco neuronas.

```
require(neuralnet)
```

```
## Loading required package: neuralnet
```

```
## Warning: package 'neuralnet' was built under R version 4.0.5
```

```
Train=data.frame(Train1.notas_p$calificacion,model.matrix(calificacion~., data=Train1.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn1)
```

```
Validate=data.frame(Val1.notas_p$calificacion,model.matrix(calificacion~., data=Val1.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val1.notas_p$calificacion, predictedNN1)
matrizNN1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       25      141
```

```
##   suspenso        9       15
##
##                Accuracy : 0.2105
##                  95% CI : (0.1549, 0.2754)
##     No Information Rate : 0.8211
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0669
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.73529
##             Specificity : 0.09615
##          Pos Pred Value : 0.15060
##          Neg Pred Value : 0.62500
##              Prevalence : 0.17895
##          Detection Rate : 0.13158
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.41572
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_p<-c(matrizNN1$overall[1])
```

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train1.notas_p$calificacion,model.matrix(calificacion~., data=Train1.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_p$calificacion,model.matrix(calificacion~., data=Val1.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val1.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train1.notas_p$calificacion,model.matrix(calificacion~., data=Train1.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_p$calificacion,model.matrix(calificacion~., data=Val1.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val1.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train1.notas_p$calificacion,model.matrix(calificacion~., data=Train1.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_p$calificacion,model.matrix(calificacion~., data=Val1.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val1.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
names(precisionNN_p)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas")
```

```
precisionNN_p
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas
##   0.2105263   0.1684211   0.1578947   0.1578947
```

El porcentaje de clasificación mediante redes neuronales de una capa es muy bajo y ni aumentando el número de neuronas se mejora.

Se prueba a continuación con una red neuronal de dos capas.

```
Train=data.frame(Train1.notas_p$calificacion,model.matrix(calificacion~., data=Train1.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn12=neuralnet(calificacion ~., data=Train, hidden=c(10,5), act.fct = "logistic", linear.output = FALSE)
plot(nn12)
```

```
Validate=data.frame(Val1.notas_p$calificacion,model.matrix(calificacion~., data=Val1.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn12,Validate)
predictedNN12=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN12<-confusionMatrix(Val1.notas_p$calificacion, predictedNN12)
matrizNN12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       14      152
##   suspenso        3       21
##
##                Accuracy : 0.1842
##                  95% CI : (0.1318, 0.2468)
##     No Information Rate : 0.9105
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0111
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.82353
##             Specificity : 0.12139
##          Pos Pred Value : 0.08434
##          Neg Pred Value : 0.87500
##              Prevalence : 0.08947
##          Detection Rate : 0.07368
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.47246
##
##        'Positive' Class : aprobado
##
```

De esta forma tampoco mejora la clasificación.

```
precision_p1<-c(precision_p1, max(precisionNN_p))
names(precision_p1)[2]<-c("Redes Neuronales")
```

Asignatura: Matemáticas

Se prueba primero, al igual que en la asignatura de portugués, con una red neuronal de una capa y cinco

neuronas.

```
Train=data.frame(Train1.notas_m$calificacion,model.matrix(calificacion~., data=Train1.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn2)
```

```
Validate=data.frame(Val1.notas_m$calificacion,model.matrix(calificacion~., data=Val1.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val1.notas_m$calificacion, predictedNN2)
matrizNN2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       14       61
##   suspenso        9       23
##
##                Accuracy : 0.3458
##                  95% CI : (0.2565, 0.4439)
##     No Information Rate : 0.785
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0645
##
##  Mcnemar's Test P-Value : 1.09e-09
##
##             Sensitivity : 0.6087
##             Specificity : 0.2738
##          Pos Pred Value : 0.1867
##          Neg Pred Value : 0.7188
##              Prevalence : 0.2150
##          Detection Rate : 0.1308
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.4413
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_m<-c(matrizNN2$overall[1])
```

El porcentaje de clasificación correcta en la asignatura de matemáticas duplica al de la asignatura de portugués y con el mismo modelo. Sin embargo, sigue siendo bastante bajo.

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train1.notas_m$calificacion,model.matrix(calificacion~., data=Train1.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_m$calificacion,model.matrix(calificacion~., data=Val1.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val1.notas_m$calificacion, predictedNN2)
```

```
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train1.notas_m$calificacion,model.matrix(calificacion~., data=Train1.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_m$calificacion,model.matrix(calificacion~., data=Val1.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val1.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train1.notas_m$calificacion,model.matrix(calificacion~., data=Train1.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val1.notas_m$calificacion,model.matrix(calificacion~., data=Val1.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val1.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
names(precisionNN_m)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas")
precisionNN_m
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas
##   0.3457944   0.3364486   0.3271028   0.2897196
```

El porcentaje de clasificación mediante redes neuronales de una capa, a pesar de ser mayor que en la asignatura de portugués, sigue siendo muy bajo y ni aumentando el número de neuronas se mejora notabemente.

Se prueba a continuación con una red neuronal de dos capas.

```
Train=data.frame(Train1.notas_m$calificacion,model.matrix(calificacion~., data=Train1.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn21=neuralnet(calificacion ~., data=Train, hidden=c(5,5), act.fct = "logistic", linear.output = FALSE)
plot(nn21)
```



```
Validate=data.frame(Val1.notas_m$calificacion,model.matrix(calificacion~., data=Val1.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn21,Validate)
predictedNN21=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN21<-confusionMatrix(Val1.notas_m$calificacion, predictedNN21)
matrizNN21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       25       50
##   suspenso       16       16
##
##                Accuracy : 0.3832
##                  95% CI : (0.2908, 0.4822)
##     No Information Rate : 0.6168
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1278
##
```

```
##  Mcnemar's Test P-Value : 4.865e-05
##
##             Sensitivity : 0.6098
##             Specificity : 0.2424
##          Pos Pred Value : 0.3333
##          Neg Pred Value : 0.5000
##              Prevalence : 0.3832
##          Detection Rate : 0.2336
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.4261
##
##        'Positive' Class : aprobado
##
```

Con esta estructura la red neuronal tampoco mejora. Con otras que se ha probado pero no se muestran tampoco mejoró.

```
precision_m1<-c(precision_m1, max(precisionNN_m))
names(precision_m1)[2]<-c("Redes Neuronales")
```

Método 3: Máquina de vector soporte Asignatura: portugués

Se ajusta, a continuación, el modelo para los datos de la asignatura de portugués con el kernel radial.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```
fitsvm11 <-svm(calificacion ~., data = Train1.notas_p)
summary(fitsvm11)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train1.notas_p)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  160
##
##  ( 100 60 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

Se predicen ahora los valores de la respuesta y se calcula la matriz de confusión.

```
predictedSVM = predict(fitsvm11,Val1.notas_p)
matrizSVM11<-confusionMatrix(Val1.notas_p$calificacion, predictedSVM)
matrizSVM11
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(matrizSVM11$overall[1])
names(precisionSVM_p)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm12 <-svm(calificacion ~., data = Train1.notas_p, kernel="polynomial")
summary(fitsvm12)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train1.notas_p, kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  218
##
##  ( 158 60 )
##
##
## Number of Classes:  2
##
## Levels:
```

```
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm12,Val1.notas_p)
matrizSVM12<-confusionMatrix(Val1.notas_p$calificacion, predictedSVM)
matrizSVM12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM12$overall[1])
names(precisionSVM_p)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm13 <-svm(calificacion ~., data = Train1.notas_p, kernel="sigmoid")
summary(fitsvm13)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train1.notas_p, kernel = "sigmoid")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  sigmoid
##        cost:  1
##      coef.0:  0
##
## Number of Support Vectors:  142
##
##  ( 82 60 )
```

```
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm13,Val1.notas_p)
matrizSVM13<-confusionMatrix(Val1.notas_p$calificacion, predictedSVM)
matrizSVM13
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM13$overall[1])
names(precisionSVM_p)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm14 <-svm(calificacion ~., data = Train1.notas_p, kernel="linear")
summary(fitsvm14)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train1.notas_p, kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
```

```
##
## Number of Support Vectors:  130
##
##  ( 77 53 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm14,Val1.notas_p)
matrizSVM14<-confusionMatrix(Val1.notas_p$calificacion, predictedSVM)
matrizSVM14
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      158        8
##   suspenso       19        5
##
##                Accuracy : 0.8579
##                  95% CI : (0.8, 0.9042)
##     No Information Rate : 0.9316
##     P-Value [Acc > NIR] : 0.99989
##
##                   Kappa : 0.1992
##
##  Mcnemar's Test P-Value : 0.05429
##
##             Sensitivity : 0.8927
##             Specificity : 0.3846
##          Pos Pred Value : 0.9518
##          Neg Pred Value : 0.2083
##              Prevalence : 0.9316
##          Detection Rate : 0.8316
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.6386
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM14$overall[1])
names(precisionSVM_p)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenidos de los distintos kernel.

```
precisionSVM_p
```

```
##     radial polinomial  sigmoidal     lineal
##  0.8736842  0.8736842  0.8736842  0.8578947
```

La predicción de los SVM de kernel radial, polinomial y sigmoidal es la misma, la cual es ligeramente mayor que la del SVM de kernel lineal.

```
precision_p1<-c(precision_p1, max(precisionSVM_p))
names(precision_p1)[3]<-c("SVM")
```

Asignatura: matemáticas

Se prueba primero con el kernel radial.

```
fitsvm21 <-svm(calificacion ~., data = Train1.notas_m)
predictedSVM = predict(fitsvm21,Val1.notas_m)
matrizSVM21<-confusionMatrix(Val1.notas_m$calificacion, predictedSVM)
matrizSVM21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       75        0
##   suspenso       32        0
##
##                Accuracy : 0.7009
##                  95% CI : (0.6048, 0.7856)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 0.7009
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.7009
##    Detection Prevalence : 0.7009
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(matrizSVM21$overall[1])
names(precisionSVM_m)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm22 <-svm(calificacion ~., data = Train1.notas_m, kernel="polynomial")
predictedSVM = predict(fitsvm22,Val1.notas_m)
matrizSVM22<-confusionMatrix(Val1.notas_m$calificacion, predictedSVM)
matrizSVM22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       75        0
##   suspenso       32        0
##
```

```
##                Accuracy : 0.7009
##                  95% CI : (0.6048, 0.7856)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 0.7009
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.7009
##    Detection Prevalence : 0.7009
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM22$overall[1])
names(precisionSVM_m)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm23 <-svm(calificacion ~., data = Train1.notas_m, kernel="sigmoid")
predictedSVM = predict(fitsvm23,Val1.notas_m)
matrizSVM23<-confusionMatrix(Val1.notas_m$calificacion, predictedSVM)
matrizSVM23
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       75        0
##   suspenso       32        0
##
##                Accuracy : 0.7009
##                  95% CI : (0.6048, 0.7856)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 0.7009
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.7009
##    Detection Prevalence : 0.7009
##       Balanced Accuracy :     NA
##
```

```
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM23$overall[1])
names(precisionSVM_m)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm24 <-svm(calificacion ~., data = Train1.notas_m, kernel="linear")
predictedSVM = predict(fitsvm24,Val1.notas_m)
matrizSVM24<-confusionMatrix(Val1.notas_m$calificacion, predictedSVM)
matrizSVM24
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       70        5
##   suspenso       21       11
##
##                Accuracy : 0.757
##                  95% CI : (0.6646, 0.8347)
##     No Information Rate : 0.8505
##     P-Value [Acc > NIR] : 0.996251
##
##                   Kappa : 0.3234
##
##  Mcnemar's Test P-Value : 0.003264
##
##             Sensitivity : 0.7692
##             Specificity : 0.6875
##          Pos Pred Value : 0.9333
##          Neg Pred Value : 0.3438
##              Prevalence : 0.8505
##          Detection Rate : 0.6542
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.7284
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM24$overall[1])
names(precisionSVM_m)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenid os de los distintos kernel.

```
precisionSVM_m
```

```
##     radial polinomial  sigmoidal     lineal
##  0.7009346  0.7009346  0.7009346  0.7570093
```

Al igual que en la asignatura de portugués, la predicción de los SVM de kernel radial, polinomial y sigmoidal es la misma. sin embargo, en este caso la predicción del SVM de kernel lineal es ligeramente mayor que las otras.

```
precision_m1<-c(precision_m1, max(precisionSVM_m))
names(precision_m1)[3]<-c("SVM")
```

Método 4: Naive Bayes Asignatura: portugués

```
fitbayes1 <-naiveBayes(calificacion ~., data = Train1.notas_p)
predictedBayes= predict(fitbayes1,Val1.notas_p)
matrizNB1<-confusionMatrix(Val1.notas_p$calificacion, predictedBayes)
matrizNB1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      150       16
##   suspenso       13       11
##
##                Accuracy : 0.8474
##                  95% CI : (0.7882, 0.8953)
##     No Information Rate : 0.8579
##     P-Value [Acc > NIR] : 0.7045
##
##                   Kappa : 0.3436
##
##  Mcnemar's Test P-Value : 0.7103
##
##             Sensitivity : 0.9202
##             Specificity : 0.4074
##          Pos Pred Value : 0.9036
##          Neg Pred Value : 0.4583
##              Prevalence : 0.8579
##          Detection Rate : 0.7895
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.6638
##
##        'Positive' Class : aprobado
##
```

```
precision_p1<-c(precision_p1, matrizNB1$overall[1])
names(precision_p1)[4]<-c("Naive Bayes")
```

Asignatura: matemáticas

```
fitbayes2 <-naiveBayes(calificacion ~., data = Train1.notas_m)
predictedBayes= predict(fitbayes2,Val1.notas_m)
matrizNB2<-confusionMatrix(Val1.notas_m$calificacion, predictedBayes)
matrizNB2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       69        6
##   suspenso       24        8
##
##                Accuracy : 0.7196
##                  95% CI : (0.6245, 0.8022)
##     No Information Rate : 0.8692
##     P-Value [Acc > NIR] : 0.999988
```

```
##
##                   Kappa : 0.2027
##
##  Mcnemar's Test P-Value : 0.001911
##
##             Sensitivity : 0.7419
##             Specificity : 0.5714
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.2500
##              Prevalence : 0.8692
##          Detection Rate : 0.6449
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.6567
##
##        'Positive' Class : aprobado
##
```

```
precision_m1<-c(precision_m1, matrizNB2$overall[1])
names(precision_m1)[4]<-c("Naive Bayes")
```

Método 5: Árboles de clasificación Asignatura: portugués

```
require(tree)
```

```
## Loading required package: tree
```

```
## Warning: package 'tree' was built under R version 4.0.5
```

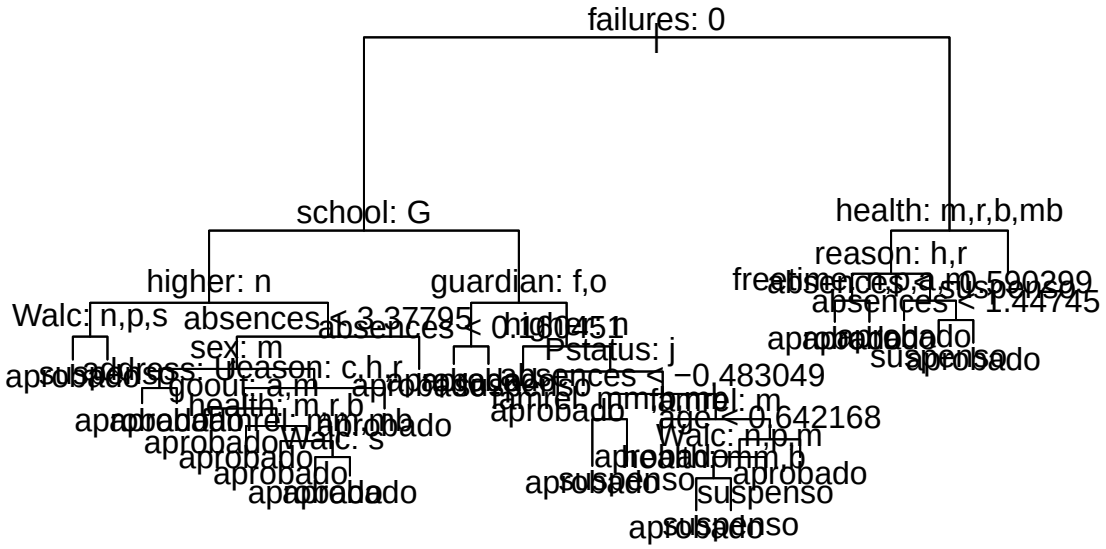
```
tree11 = tree(calificacion~., data = Train1.notas_p)
summary(tree11)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train1.notas_p)
## Variables actually used in tree construction:
##  [1] "failures" "school"   "higher"   "Walc"     "absences" "sex"
##  [7] "address"  "reason"   "goout"    "health"   "famrel"   "guardian"
## [13] "Pstatus"  "age"      "freetime"
## Number of terminal nodes:  28
## Residual mean deviance:  0.3385 = 140.5 / 415
## Misclassification error rate: 0.07901 = 35 / 443
```

```
plot(tree11)
text(tree11, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

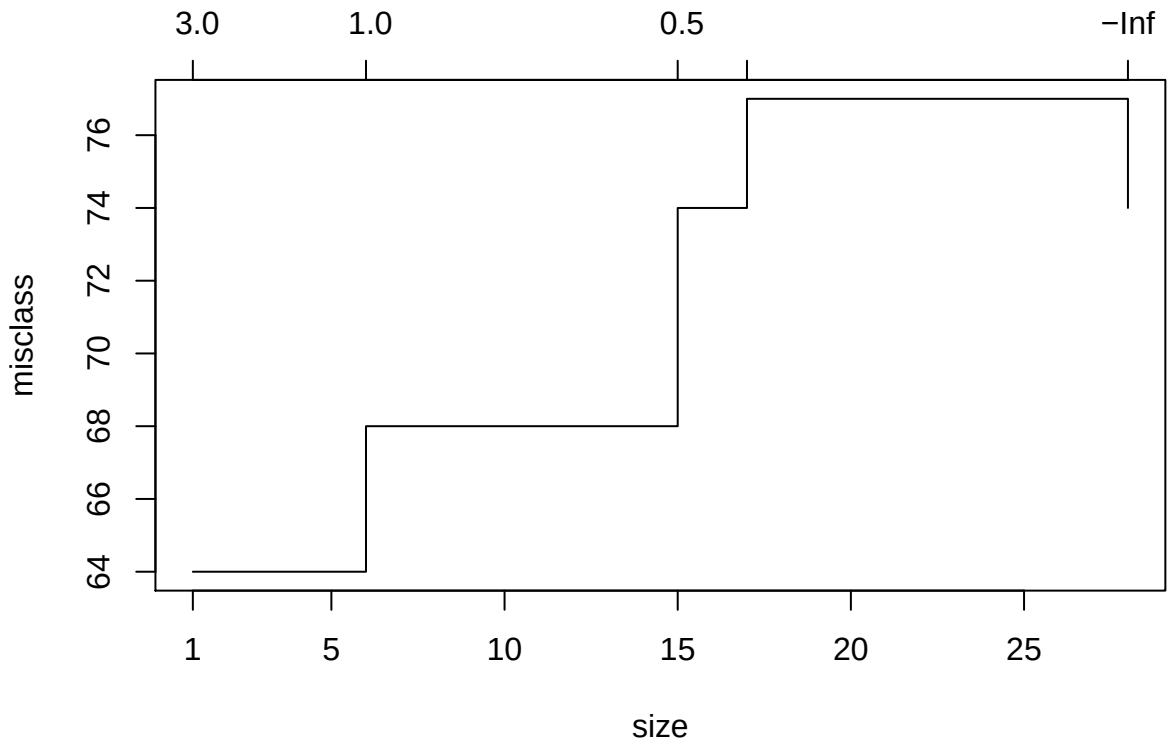
```
predicedtree = predict(tree11, Val1.notas_p, type="class")
matriztree11<-confusionMatrix(Val1.notas_p$calificacion, predicedtree)
matriztree11
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      150       16
##   suspenso       10       14
##
##                Accuracy : 0.8632
##                  95% CI : (0.806, 0.9086)
##     No Information Rate : 0.8421
##     P-Value [Acc > NIR] : 0.2468
##
##                   Kappa : 0.4399
##
##  Mcnemar's Test P-Value : 0.3268
##
##             Sensitivity : 0.9375
##             Specificity : 0.4667
##          Pos Pred Value : 0.9036
##          Neg Pred Value : 0.5833
##              Prevalence : 0.8421
##          Detection Rate : 0.7895
```

```
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7021
##
##        'Positive' Class : aprobado
##
```

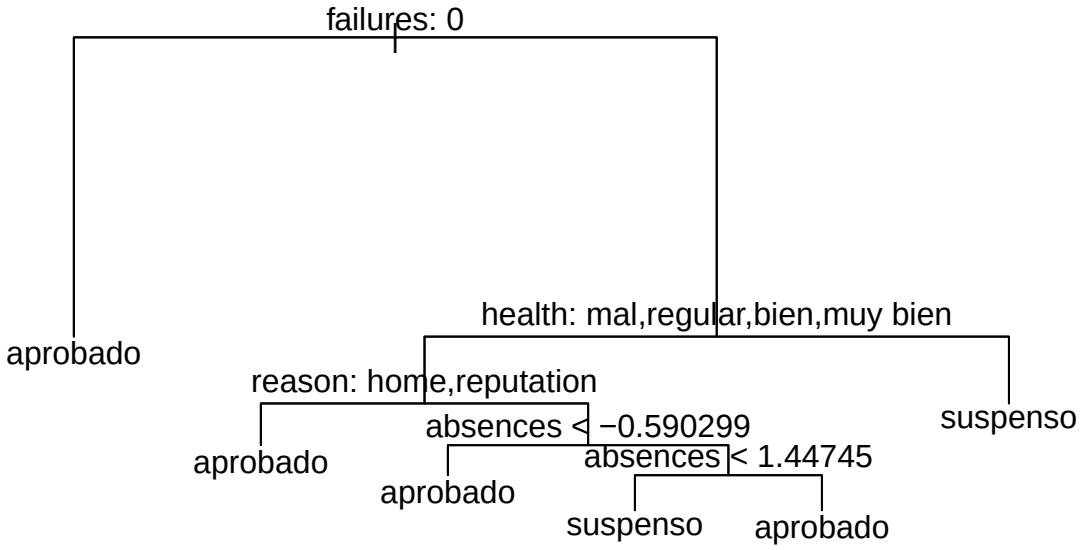
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree11 = cv.tree(tree11, FUN = prune.misclass)
plot(cv.tree11)
```



Se observa como al aumentar el tamaño del árbol, el porcentaje de error aumenta debido a un posible sobre ajuste. Por ello, se elige que tenga 5 ramas.

```
prune.tree11 = prune.misclass(tree11, best = 5)
plot(prune.tree11)
text(prune.tree11, pretty=0)
```



Se observa que las ramas corresponden a los suspensos, la salud, la razón de elección del colegio y las ausencias.

```
predicedtree12 = predict(prune.tree11, Val1.notas_p, type="class")
matriztree12<-confusionMatrix(Val1.notas_p$calificacion, predicedtree12)
matriztree12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      158        8
##   suspenso       15        9
##
##                Accuracy : 0.8789
##                  95% CI : (0.8239, 0.9217)
##     No Information Rate : 0.9105
##     P-Value [Acc > NIR] : 0.9455
##
##                   Kappa : 0.3734
##
##  Mcnemar's Test P-Value : 0.2109
##
##             Sensitivity : 0.9133
##             Specificity : 0.5294
##          Pos Pred Value : 0.9518
##          Neg Pred Value : 0.3750
##              Prevalence : 0.9105
```

```
##          Detection Rate : 0.8316
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7214
##
##        'Positive' Class : aprobado
##
```

```
precision_p1<-c(precision_p1, matriztree12$overall[1])
names(precision_p1)[5]<-c("Arbol de clasificación")
```

Asignatura: matemáticas

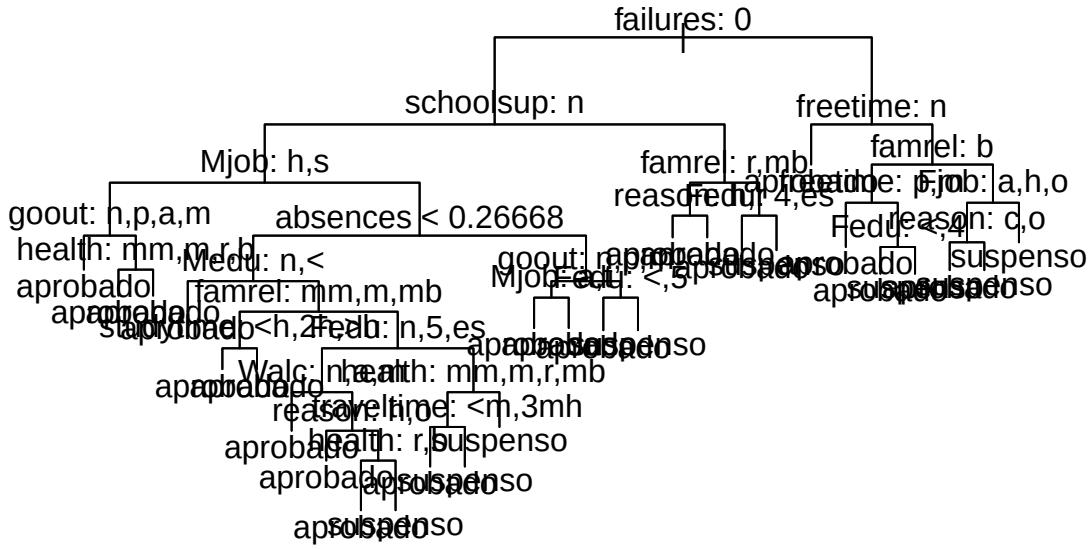
```
tree21 = tree(calificacion~., data = Train1.notas_m)
summary(tree21)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train1.notas_m)
## Variables actually used in tree construction:
##  [1] "failures"   "schoolsup"  "Mjob"       "goout"      "health"
##  [6] "absences"   "Medu"       "famrel"     "studytime"  "Fedu"
## [11] "Walc"       "reason"     "traveltime" "freetime"   "Fjob"
## Number of terminal nodes:  28
## Residual mean deviance:  0.4633 = 102.9 / 222
## Misclassification error rate: 0.1 = 25 / 250
```

```
plot(tree21)
text(tree21, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII

## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

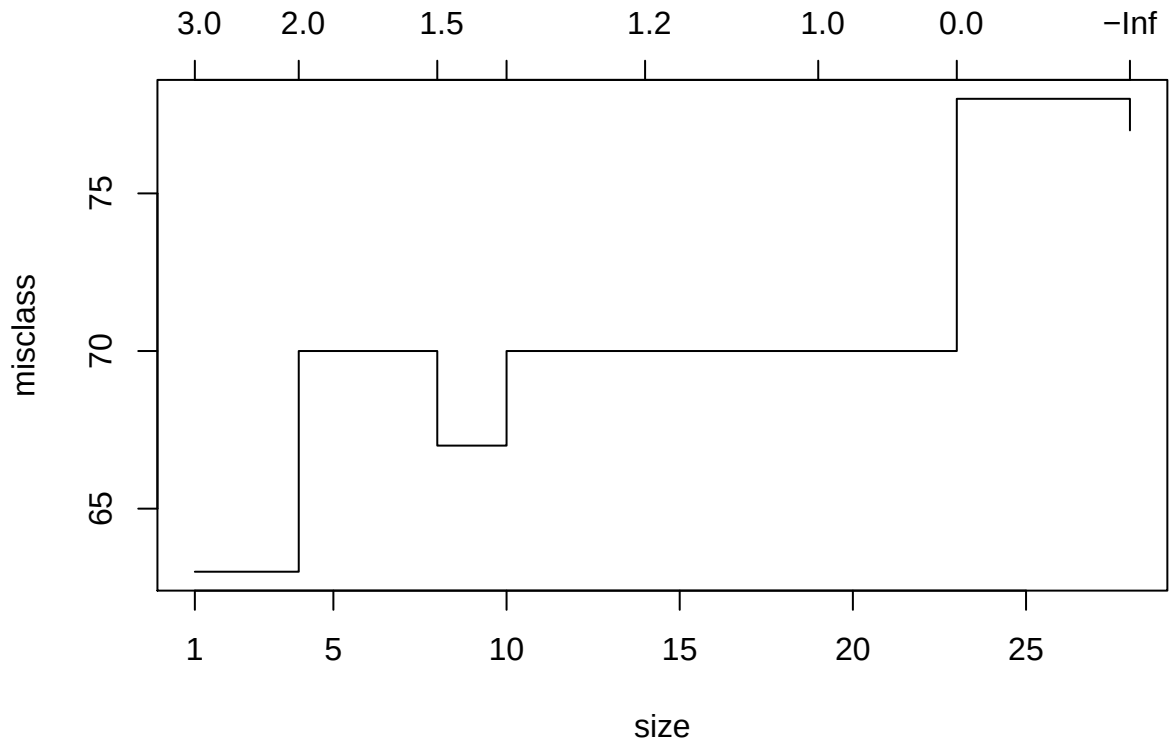
```
predicedtree = predict(tree21, Val1.notas_m, type="class")
matriztree21<-confusionMatrix(Val1.notas_m$calificacion, predicedtree)
matriztree21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       67        8
##   suspenso       21       11
##
##                Accuracy : 0.729
##                  95% CI : (0.6345, 0.8104)
##     No Information Rate : 0.8224
##     P-Value [Acc > NIR] : 0.99424
##
##                   Kappa : 0.2683
##
##  Mcnemar's Test P-Value : 0.02586
##
##             Sensitivity : 0.7614
##             Specificity : 0.5789
##          Pos Pred Value : 0.8933
##          Neg Pred Value : 0.3438
##              Prevalence : 0.8224
##          Detection Rate : 0.6262
```

```
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.6702
##
##        'Positive' Class : aprobado
##
```

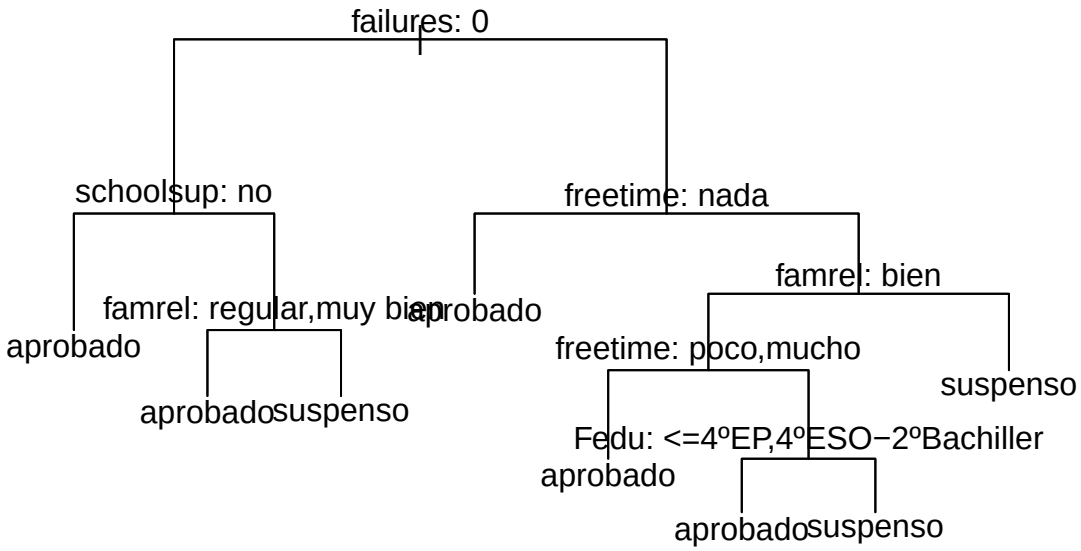
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree21 = cv.tree(tree21, FUN = prune.misclass)
plot(cv.tree21)
```



Se observa como al aumentar el tamaño del árbol, el porcentaje de error aumenta debido a un posible sobre ajuste. Por ello, se elige que tenga 5 ramas.

```
prune.tree21 = prune.misclass(tree21, best = 5)
plot(prune.tree21)
text(prune.tree21, pretty=0)
```



Se observa que las ramas corresponden a los suspensos, el apoyo del colegio, el tiempo libre, la relación con la familia y la educación del padre. Variables totalmente distinta a excepción de los suspensos a las de la asignatura de portugués.

```
predicedtree22 = predict(prune.tree21, Val1.notas_m, type="class")
matriztree22<-confusionMatrix(Val1.notas_m$calificacion, predicedtree22)
matriztree22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       68        7
##   suspenso       21       11
##
##                Accuracy : 0.7383
##                  95% CI : (0.6445, 0.8185)
##     No Information Rate : 0.8318
##     P-Value [Acc > NIR] : 0.99494
##
##                   Kappa : 0.2863
##
##  Mcnemar's Test P-Value : 0.01402
##
##             Sensitivity : 0.7640
##             Specificity : 0.6111
##          Pos Pred Value : 0.9067
```

```
##          Neg Pred Value : 0.3438
##              Prevalence : 0.8318
##          Detection Rate : 0.6355
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.6876
##
##        'Positive' Class : aprobado
##
```

```
precision_m1<-c(precision_m1, matriztree22$overall[1])
names(precision_m1)[5]<-c("Arbol de clasificación")
```

Escenario 2: con G1 y sin G2

Método 1: Regresión logística Asignatura: portugués

Primero se ajusta al modelo completo.

```
gfit1=glm(calificacion~., data=notas_p[,!(names(notas_p) %in% c("G2", "G3"))], family=binomial)
```

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

```
summary(gfit1)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_p[,
##     !(names(notas_p) %in% c("G2", "G3"))])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.15909  -0.13532  -0.02742  -0.00320   2.93644
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -29.31126 1748.65943  -0.017  0.98663
## schoolMS                    1.16011    0.65836   1.762  0.07805 .
## sexhombre                   0.92580    0.64737   1.430  0.15269
## age                        -0.79636    0.32214  -2.472  0.01343 *
## addressRural               -0.34757    0.54714  -0.635  0.52527
## famsizeLE3                 -0.02923    0.57029  -0.051  0.95912
## Pstatusseparados           -0.20692    0.84846  -0.244  0.80732
## Medu<=4ºEP                 14.24574 1290.42506   0.011  0.99119
## Medu5ºEP-3ºESO             13.74610 1290.42523   0.011  0.99150
## Medu4ºESO-2ºBachiller      13.79168 1290.42527   0.011  0.99147
## Meduestudios superiores    13.03984 1290.42580   0.010  0.99194
## Fedu<=4ºEP                 14.08902 1180.08831   0.012  0.99047
## Fedu5ºEP-3ºESO             12.51912 1180.08857   0.011  0.99154
## Fedu4ºESO-2ºBachiller      13.21377 1180.08850   0.011  0.99107
## Feduestudios superiores    13.86775 1180.08870   0.012  0.99062
## Mjobhealth                  1.03556    1.12178   0.923  0.35594
## Mjobother                   0.04080    0.63341   0.064  0.94864
## Mjobservices                0.80859    0.78406   1.031  0.30241
## Mjobteacher                -0.44090    1.52989  -0.288  0.77320
## Fjobhealth                  1.09193    1.50390   0.726  0.46780
## Fjobother                   1.11391    0.98559   1.130  0.25839
```

```
## Fjobservices                1.40498    1.06938   1.314  0.18891
## Fjobteacher                 0.74070    2.36682   0.313  0.75432
## reasonhome                 -0.22699    0.71920  -0.316  0.75230
## reasonother                 0.09503    0.76988   0.123  0.90176
## reasonreputation           -0.31224    0.82056  -0.381  0.70356
## guardianmother              0.35892    0.65592   0.547  0.58424
## guardianother               0.23629    1.17178   0.202  0.84019
## traveltime15-30 min        -1.72969    0.61494  -2.813  0.00491 **
## traveltime30 min.-1 hora   -1.17006    0.83367  -1.403  0.16047
## traveltime>1 hora          -1.19053    1.29421  -0.920  0.35763
## studytime2-5 horas         -0.14110    0.52585  -0.268  0.78844
## studytime5-10 horas         0.62058    0.91114   0.681  0.49581
## studytime>10 horas          0.03053    1.58605   0.019  0.98464
## failures1                   0.98411    0.64088   1.536  0.12464
## failures2                   1.54276    1.08350   1.424  0.15448
## failures>=3                 2.14993    1.02431   2.099  0.03583 *
## schoolsupyes               -0.20036    0.86767  -0.231  0.81738
## famsupyes                   0.19246    0.50629   0.380  0.70385
## paidyes                     1.42727    0.87151   1.638  0.10148
## activitiesyes              -0.75746    0.49291  -1.537  0.12436
## nurseryyes                  0.08549    0.58293   0.147  0.88340
## higheryes                  -1.51382    0.64218  -2.357  0.01841 *
## internetyes                 1.08324    0.65138   1.663  0.09631 .
## romanticyes                -0.06592    0.51274  -0.129  0.89770
## famrelmal                  -3.03027    1.58674  -1.910  0.05617 .
## famrelregular              -2.26491    1.21133  -1.870  0.06152 .
## famrelbien                 -3.04181    1.17842  -2.581  0.00984 **
## famrelmuy bien             -2.49356    1.21421  -2.054  0.04001 *
## freetimepoco               -0.86863    1.08093  -0.804  0.42163
## freetimealgo               -0.91274    1.01544  -0.899  0.36873
## freetimesuficiente         -1.48074    1.01018  -1.466  0.14270
## freetimemucho              -0.77058    1.05696  -0.729  0.46597
## gooutpoco                  -1.46272    1.00938  -1.449  0.14730
## gooutalgo                  -1.38292    0.97851  -1.413  0.15757
## gooutsuficiente            -1.11475    1.04835  -1.063  0.28763
## gooutmucho                 -0.33131    1.07258  -0.309  0.75740
## Dalcpoco                    0.54539    0.70892   0.769  0.44170
## Dalcalgo                   -0.89861    0.90832  -0.989  0.32251
## Dalcsuficiente             -2.39178    2.57809  -0.928  0.35355
## Dalcmucho                  -0.50349    1.55498  -0.324  0.74610
## Walcpoco                    0.48819    0.73849   0.661  0.50857
## Walcalgo                    0.50976    0.81112   0.628  0.52970
## Walcsuficiente              0.22729    0.92168   0.247  0.80522
## Walcmucho                   0.67304    1.24200   0.542  0.58789
## healthmal                  -0.60287    0.97298  -0.620  0.53551
## healthregular              -0.55547    0.88921  -0.625  0.53218
## healthbien                  0.03287    0.96676   0.034  0.97288
## healthmuy bien              0.79420    0.85894   0.925  0.35516
## absences                    0.32627    0.25587   1.275  0.20227
## G1                         -3.86658    0.61516  -6.286 3.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 495.63  on 632  degrees of freedom
## Residual deviance: 174.08  on 562  degrees of freedom
## AIC: 316.08
##
## Number of Fisher Scoring iterations: 16
```

Resultan significativas las siguientes variables: age, traveltime, failures, higher, famrel y ,especialmente significativa como era de esperar, G1.

Sin embargo, lo que interesa es la predicción.

```
gfit12=glm(calificacion~., data=Train2.notas_p, family=binomial)
cbind(gfit1$coefficients, gfit12$coefficients)
```

```
##                                [,1]          [,2]
## (Intercept)              -29.31125777 -23.283417805
## schoolMS                   1.16011490   0.636266292
## sexhombre                  0.92579859   1.466099191
## age                       -0.79635548  -1.016754407
## addressRural              -0.34756562  -0.669316549
## famsizeLE3                -0.02923013  -0.379560479
## Pstatusseparados          -0.20692485  -0.173613220
## Medu<=4ºEP                14.24573877   9.897907498
## Medu5ºEP-3ºESO            13.74610397   8.006819846
## Medu4ºESO-2ºBachiller     13.79168055   8.367164232
## Meduestudios superiores   13.03983525   6.737901322
## Fedu<=4ºEP                14.08901743  13.524762601
## Fedu5ºEP-3ºESO            12.51912126  12.264157600
## Fedu4ºESO-2ºBachiller     13.21376938  12.119234937
## Feduestudios superiores   13.86775132  12.390437873
## Mjobhealth                 1.03556063   2.679710856
## Mjobother                  0.04080301   0.932556158
## Mjobservices               0.80858578   1.195847331
## Mjobteacher               -0.44089893   0.827691602
## Fjobhealth                 1.09193313   2.515292396
## Fjobother                  1.11391261   1.659602073
## Fjobservices               1.40497750   2.160482821
## Fjobteacher                0.74069826   2.150600145
## reasonhome                -0.22698626  -0.570849508
## reasonother                0.09502952   1.392905430
## reasonreputation          -0.31223638  -0.539215246
## guardianmother             0.35892165   0.692757505
## guardianother              0.23629231  -0.107412342
## traveltime15-30 min       -1.72968979  -2.817789706
## traveltime30 min.-1 hora  -1.17005595  -1.801277320
## traveltime>1 hora         -1.19052573  -0.087001591
## studytime2-5 horas        -0.14110358   0.270388138
## studytime5-10 horas        0.62058180   0.338404961
## studytime>10 horas         0.03052523  -1.049010302
## failures1                  0.98411090   0.073845944
## failures2                  1.54275546   1.765206761
## failures>=3                2.14992668   0.326896923
## schoolsupyes              -0.20036078  -1.074790350
## famsupyes                  0.19245589   1.515611933
## paidyes                    1.42727436   1.786545187
```

```
## activitiesyes             -0.75746433  -0.331142226
## nurseryyes                 0.08549442   1.313278993
## higheryes                 -1.51381639  -2.276029129
## internetyes                1.08323658   1.186538576
## romanticyes               -0.06592445   0.592172540
## famrelmal                 -3.03027296  -6.219658955
## famrelregular             -2.26491241  -2.554899189
## famrelbien                -3.04180941  -4.839881776
## famrelmuy bien            -2.49355997  -4.155278060
## freetimepoco              -0.86863012   0.461653052
## freetimealgo              -0.91273991  -0.214858749
## freetimesuficiente        -1.48073673  -0.959845835
## freetimemucho             -0.77058255  -0.052979737
## gooutpoco                 -1.46272226  -3.967108783
## gooutalgo                 -1.38291521  -2.379964182
## gooutsuficiente           -1.11475236  -2.449802904
## gooutmucho                -0.33131356  -1.330843554
## Dalcpoco                   0.54539143   0.125684535
## Dalcalgo                  -0.89861370  -0.599163161
## Dalcsuficiente            -2.39177573  -3.857590420
## Dalcmucho                 -0.50349013  -1.341464625
## Walcpoco                   0.48819042  -0.881865058
## Walcalgo                   0.50975775  -0.177422666
## Walcsuficiente             0.22728842   0.002738512
## Walcmucho                  0.67303910   0.935482322
## healthmal                 -0.60287155  -2.018313369
## healthregular             -0.55547424  -1.377112513
## healthbien                 0.03287101  -0.700574901
## healthmuy bien             0.79420076  -0.041334946
## absences                   0.32627029   0.471646650
## G1                        -3.86657530  -4.336479394
```

```
p=predict(gfit12, Val2.notas_p, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
matrizLogis<-confusionMatrix(Val2.notas_p$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      157        9
##   suspenso       13       11
##
##                Accuracy : 0.8842
##                  95% CI : (0.83, 0.926)
##     No Information Rate : 0.8947
##     P-Value [Acc > NIR] : 0.7295
##
##                   Kappa : 0.4351
##
##  Mcnemar's Test P-Value : 0.5224
##
##             Sensitivity : 0.9235
```

```
##             Specificity : 0.5500
##          Pos Pred Value : 0.9458
##          Neg Pred Value : 0.4583
##              Prevalence : 0.8947
##          Detection Rate : 0.8263
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7368
##
##        'Positive' Class : aprobado
##
```

El porcentaje de clasificación correcta es del 88%.

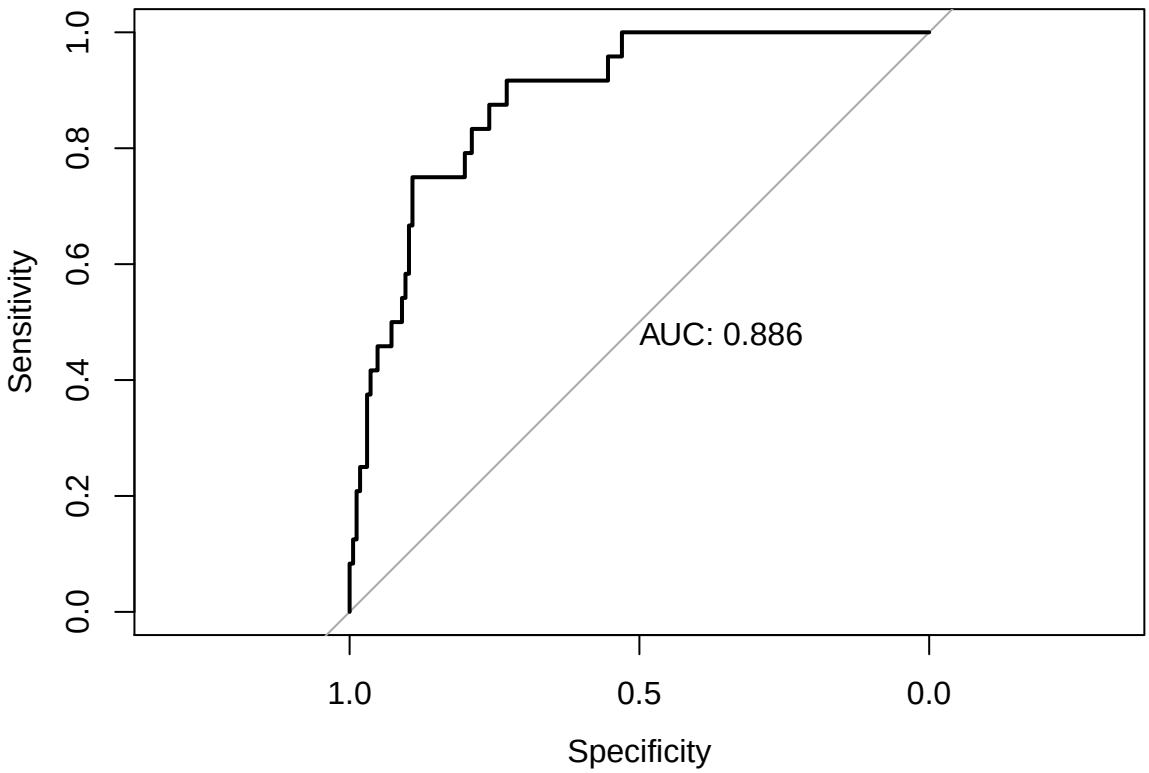
```
precision_p2<-c(matrizLogis$overall[1])
names(precision_p2)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC.

```
test_prob = predict(gfit12, newdata = Val2.notas_p, type = "response")
test_roc = roc(Val2.notas_p$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0,886 que es un valor alto y por tanto confirma que el modelo es bueno.

Asignatura: matemáticas

Primero se ajusta al modelo completo.

```r
gfit2=glm(calificacion~., data=notas_m[,!(names(notas_m) %in% c("G2", "G3"))], family=binomial)
summary(gfit2)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_m[,
##     !(names(notas_m) %in% c("G2", "G3"))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4295  -0.0693  -0.0022   0.0011   1.7897
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -20.71831 1474.01314  -0.014  0.98879
## schoolMS                   -2.50046    1.58516  -1.577  0.11470
## sexhombre                   1.41877    0.86265   1.645  0.10004
## age                         1.18204    0.55128   2.144  0.03202 *
## addressRural               -1.94914    0.95414  -2.043  0.04107 *
## famsizeLE3                  1.02057    0.82357   1.239  0.21527
## Pstatusseparados            1.47993    1.07165   1.381  0.16729
## Medu<=4ºEP                 -3.81186  444.80765  -0.009  0.99316
## Medu5ºEP-3ºESO             -3.76876  444.80795  -0.008  0.99324
## Medu4ºESO-2ºBachiller      -2.61838  444.80923  -0.006  0.99530
## Meduestudios superiores    -4.71188  444.81176  -0.011  0.99155
## Fedu<=4ºEP                 11.85018 1405.23764   0.008  0.99327
## Fedu5ºEP-3ºESO             12.62180 1405.23774   0.009  0.99283
## Fedu4ºESO-2ºBachiller      12.95886 1405.23735   0.009  0.99264
## Feduestudios superiores    12.16376 1405.23772   0.009  0.99309
## Mjobhealth                  0.02114    2.08809   0.010  0.99192
## Mjobother                   0.17030    1.11264   0.153  0.87835
## Mjobservices               -0.77470    1.37180  -0.565  0.57225
## Mjobteacher                 2.79986    1.62392   1.724  0.08468 .
## Fjobhealth                  1.88429    2.13734   0.882  0.37799
## Fjobother                  -3.89635    2.02893  -1.920  0.05481 .
## Fjobservices               -1.82717    1.88041  -0.972  0.33121
## Fjobteacher                -4.07917    3.30446  -1.234  0.21704
## reasonhome                 -1.74483    1.04089  -1.676  0.09368 .
## reasonother                 0.27647    1.32683   0.208  0.83494
## reasonreputation           -1.19331    1.11298  -1.072  0.28364
## guardianmother              0.96495    0.92028   1.049  0.29439
## guardianother               2.52066    1.84060   1.369  0.17085
## traveltime15-30 min         0.73464    0.91396   0.804  0.42152
## traveltime30 min.-1 hora   -1.75773    1.77660  -0.989  0.32248
## traveltime>1 hora           3.93666    2.72884   1.443  0.14913
## studytime2-5 horas          0.82371    1.14826   0.717  0.47316
## studytime5-10 horas         1.45703    1.58891   0.917  0.35914
## studytime>10 horas          0.39098    1.69453   0.231  0.81752
## failures1                  -1.13024    1.12131  -1.008  0.31347
## failures2                   2.69201    2.80618   0.959  0.33740
## failures>=3                 0.45234    1.86229   0.243  0.80809
## schoolsupyes                1.48967    0.83292   1.788  0.07370 .
## famsupyes                   0.95985    0.73449   1.307  0.19127
## paidyes                    -0.94687    0.86881  -1.090  0.27578
```

```
## activitiesyes              -0.22493    0.78355  -0.287  0.77406
## nurseryyes                  1.04619    1.06259   0.985  0.32484
## higheryes                   0.33837    2.37543   0.142  0.88673
## internetyes                -1.30719    0.95515  -1.369  0.17113
## romanticyes                 1.51341    0.97617   1.550  0.12106
## famrelmal                   0.02255   12.39580   0.002  0.99855
## famrelregular              -0.03035   12.34889  -0.002  0.99804
## famrelbien                 -2.08110   12.34599  -0.169  0.86614
## famrelmuy bien             -2.67473   12.35693  -0.216  0.82863
## freetimepoco                2.58716    1.91084   1.354  0.17576
## freetimealgo                0.47303    1.72056   0.275  0.78337
## freetimesuficiente          2.82348    1.73519   1.627  0.10370
## freetimemucho               3.87843    2.26663   1.711  0.08706 .
## gooutpoco                   4.17604    2.37291   1.760  0.07843 .
## gooutalgo                   6.04660    2.57966   2.344  0.01908 *
## gooutsuficiente             7.32131    2.86235   2.558  0.01053 *
## gooutmucho                  6.47198    2.71966   2.380  0.01733 *
## Dalcpoco                   -0.28222    1.07123  -0.263  0.79220
## Dalcalgo                   -0.21118    1.76581  -0.120  0.90481
## Dalcsuficiente              1.72560    2.03982   0.846  0.39758
## Dalcmucho                   0.29155    3.24720   0.090  0.92846
## Walcpoco                   -1.71337    1.06463  -1.609  0.10754
## Walcalgo                   -2.61901    1.18661  -2.207  0.02730 *
## Walcsuficiente             -4.07027    1.47925  -2.752  0.00593 **
## Walcmucho                  -6.65797    2.75840  -2.414  0.01579 *
## healthmal                   1.74856    1.74525   1.002  0.31639
## healthregular               3.57877    1.83121   1.954  0.05066 .
## healthbien                  2.11837    1.58975   1.333  0.18269
## healthmuy bien              3.12776    1.78070   1.756  0.07901 .
## absences                    1.12591    0.36499   3.085  0.00204 **
## G1                         -7.34567    1.36726  -5.373 7.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 407.44  on 356  degrees of freedom
## Residual deviance: 114.66  on 286  degrees of freedom
## AIC: 256.66
##
## Number of Fisher Scoring iterations: 15
```

Resultan significativas las siguientes variables: age, address, goout, walc, absences y G1. Comparando con el anterior escenario, únicamente goout y absences se mantienen significativas, el resto son nuevas.

A continuación la predicción.

```
gfit21=glm(calificacion~., data=Train2.notas_m, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
cbind(gfit1$coefficients, gfit21$coefficients)
```

```
##                                [,1]         [,2]
## (Intercept)             -29.31125777  253.6869226
```

```
## schoolMS                   1.16011490  -72.7839401
## sexhombre                  0.92579859   47.3124667
## age                       -0.79635548   18.1264280
## addressRural              -0.34756562  -32.9727157
## famsizeLE3                -0.02923013    9.6199180
## Pstatusseparados          -0.20692485   59.7065479
## Medu<=4ºEP                14.24573877 -241.5534970
## Medu5ºEP-3ºESO            13.74610397 -316.1085998
## Medu4ºESO-2ºBachiller     13.79168055 -274.6763472
## Meduestudios superiores   13.03983525 -279.9626655
## Fedu<=4ºEP                14.08901743 -218.1402807
## Fedu5ºEP-3ºESO            12.51912126 -164.1769820
## Fedu4ºESO-2ºBachiller     13.21376938 -239.6502200
## Feduestudios superiores   13.86775132 -199.3954623
## Mjobhealth                 1.03556063  -57.9155728
## Mjobother                  0.04080301   29.7343479
## Mjobservices               0.80858578  -11.7084378
## Mjobteacher               -0.44089893  -24.2478741
## Fjobhealth                 1.09193313   12.4353506
## Fjobother                  1.11391261 -112.5900777
## Fjobservices               1.40497750 -102.9716887
## Fjobteacher                0.74069826 -152.6068038
## reasonhome                -0.22698626    2.9206468
## reasonother                0.09502952    0.6170199
## reasonreputation          -0.31223638  -39.7041184
## guardianmother             0.35892165   48.8877139
## guardianother              0.23629231   60.8003913
## traveltime15-30 min       -1.72968979   10.0738850
## traveltime30 min.-1 hora  -1.17005595  -33.6668625
## traveltime>1 hora         -1.19052573   67.7489103
## studytime2-5 horas        -0.14110358    3.0404557
## studytime5-10 horas        0.62058180   18.9094829
## studytime>10 horas         0.03052523 -115.8991707
## failures1                  0.98411090    0.6436571
## failures2                  1.54275546   62.5136782
## failures>=3                2.14992668  -56.1738967
## schoolsupyes              -0.20036078   18.1398025
## famsupyes                  0.19245589   38.3766722
## paidyes                    1.42727436   -7.1040037
## activitiesyes             -0.75746433   15.8771706
## nurseryyes                 0.08549442   38.9957195
## higheryes                 -1.51381639    8.1681567
## internetyes                1.08323658   37.6495909
## romanticyes               -0.06592445   16.0307564
## famrelmal                 -3.03027296 -214.0610465
## famrelregular             -2.26491241 -193.6366959
## famrelbien                -3.04180941 -226.5210078
## famrelmuy bien            -2.49355997 -218.7131676
## freetimepoco              -0.86863012   63.0963456
## freetimealgo              -0.91273991    7.6365230
## freetimesuficiente        -1.48073673    5.0701077
## freetimemucho             -0.77058255   80.3371020
## gooutpoco                 -1.46272226  168.6093021
## gooutalgo                 -1.38291521  214.2928315
```

```
## gooutsuficiente           -1.11475236  215.6115497
## gooutmucho                -0.33131356  248.8774759
## Dalcpoco                   0.54539143  -55.1612645
## Dalcalgo                  -0.89861370  -12.6950504
## Dalcsuficiente            -2.39177573  138.4396365
## Dalcmucho                 -0.50349013  128.8527210
## Walcpoco                   0.48819042    1.9927701
## Walcalgo                   0.50975775  -23.0673339
## Walcsuficiente             0.22728842  -16.0988386
## Walcmucho                  0.67303910 -250.5977989
## healthmal                 -0.60287155   54.2917452
## healthregular             -0.55547424  113.1521499
## healthbien                 0.03287101  103.1333469
## healthmuy bien             0.79420076   31.1992476
## absences                   0.32627029   21.0308796
## G1                        -3.86657530 -148.9220728
```

Con este modelo, predecimos los valores de calificación en la asignatura de matemáticas.

```
p=predict(gfit21, Val2.notas_m, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
matrizLogis<-confusionMatrix(Val2.notas_m$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       61       14
##   suspenso       13       19
##
##                Accuracy : 0.7477
##                  95% CI : (0.6545, 0.8267)
##     No Information Rate : 0.6916
##     P-Value [Acc > NIR] : 0.1238
##
##                   Kappa : 0.4035
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.8243
##             Specificity : 0.5758
##          Pos Pred Value : 0.8133
##          Neg Pred Value : 0.5938
##              Prevalence : 0.6916
##          Detection Rate : 0.5701
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.7000
##
##        'Positive' Class : aprobado
##
```

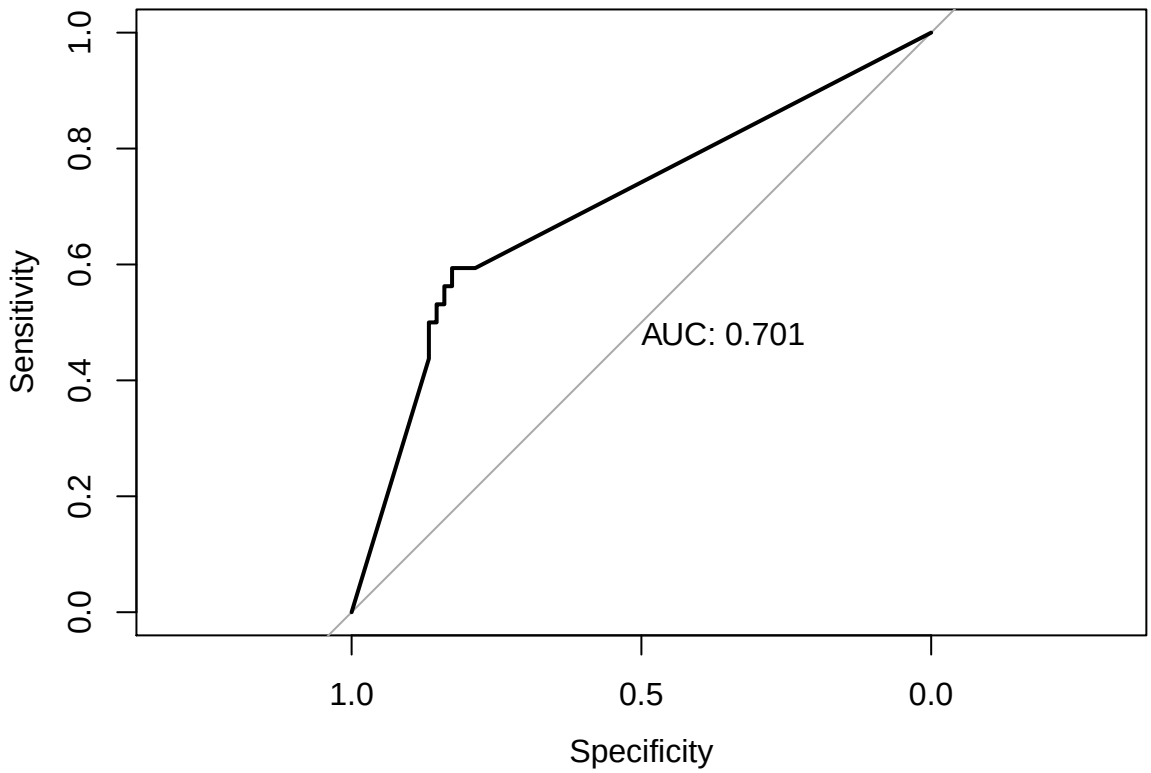
```
precision_m2<-c(matrizLogis$overall[1])
names(precision_m2)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC para comprobar el modelo.

```
test_prob = predict(gfit21, newdata = Val2.notas_m, type = "response")
test_roc = roc(Val2.notas_m$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0.701, la cual es ligeramente mayor que en el anterior escenario pero aun así sigue siendo un valor no muy alto.

Método 2: Redes neuronales Asignatura: portugués

Se prueba primero con una red neuronal de una capa y cinco neuronas.

```
Train=data.frame(Train2.notas_p$calificacion,model.matrix(calificacion~., data=Train2.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn1)
```

```
Validate=data.frame(Val2.notas_p$calificacion,model.matrix(calificacion~., data=Val2.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val2.notas_p$calificacion, predictedNN1)
matrizNN1
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction aprobado suspenso
##   aprobado        6      160
##   suspenso        6       18
##
##                Accuracy : 0.1263
##                  95% CI : (0.0826, 0.1821)
##     No Information Rate : 0.9368
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0571
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.50000
##             Specificity : 0.10112
##          Pos Pred Value : 0.03614
##          Neg Pred Value : 0.75000
##              Prevalence : 0.06316
##          Detection Rate : 0.03158
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.30056
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_p<-c(matrizNN1$overall[1])
```

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train2.notas_p$calificacion,model.matrix(calificacion~., data=Train2.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_p$calificacion,model.matrix(calificacion~., data=Val2.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val2.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train2.notas_p$calificacion,model.matrix(calificacion~., data=Train2.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_p$calificacion,model.matrix(calificacion~., data=Val2.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val2.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train2.notas_p$calificacion,model.matrix(calificacion~., data=Train2.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_p$calificacion,model.matrix(calificacion~., data=Val2.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
```

```
matrizNN1<-confusionMatrix(Val2.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
names(precisionNN_p)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas")
precisionNN_p
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas
##   0.1263158   0.1157895   0.1263158   0.1105263
```

El porcentaje de clasificación mediante redes neuronales de una capa es muy bajo y ni aumentando el número de neuronas se mejora notablemente. Por ahora la mejor estructura es con 10 neuronas.

Se prueba a continuación con una red neuronal de dos capas.

```
Train=data.frame(Train2.notas_p$calificacion,model.matrix(calificacion~., data=Train2.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn12=neuralnet(calificacion ~., data=Train, hidden=c(10,5), act.fct = "logistic", linear.output = FALSE)
plot(nn12)
```

```
Validate=data.frame(Val2.notas_p$calificacion,model.matrix(calificacion~., data=Val2.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn12,Validate)
predictedNN12=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN12<-confusionMatrix(Val2.notas_p$calificacion, predictedNN12)
matrizNN12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       12      154
##   suspenso       13       11
##
##                Accuracy : 0.1211
##                  95% CI : (0.0783, 0.1761)
##     No Information Rate : 0.8684
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1336
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.48000
##             Specificity : 0.06667
##          Pos Pred Value : 0.07229
##          Neg Pred Value : 0.45833
##              Prevalence : 0.13158
##          Detection Rate : 0.06316
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.27333
##
##        'Positive' Class : aprobado
##
```

De esta forma tampoco mejora la clasificación.

```
precision_p2<-c(precision_p2, max(precisionNN_p))
names(precision_p2)[2]<-c("Redes Neuronales")
```

Asignatura: Matemáticas

Se prueba primero con una red neuronal de una capa y cinco neuronas.

```
Train=data.frame(Train2.notas_m$calificacion,model.matrix(calificacion~., data=Train2.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn2)
```

```
Validate=data.frame(Val2.notas_m$calificacion,model.matrix(calificacion~., data=Val2.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val2.notas_m$calificacion, predictedNN2)
matrizNN2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado        9       66
##   suspenso       13       19
##
##                Accuracy : 0.2617
##                  95% CI : (0.1815, 0.3555)
##     No Information Rate : 0.7944
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1941
##
##  Mcnemar's Test P-Value : 4.902e-09
##
##             Sensitivity : 0.40909
##             Specificity : 0.22353
##          Pos Pred Value : 0.12000
##          Neg Pred Value : 0.59375
##              Prevalence : 0.20561
##          Detection Rate : 0.08411
##    Detection Prevalence : 0.70093
##       Balanced Accuracy : 0.31631
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_m<-c(matrizNN2$overall[1])
```

El porcentaje de clasificación correcta en la asignatura de matemáticas duplica al de la asignatura de portugués y con el mismo modelo al igual que en el escenario anterior. Sin embargo, sigue siendo bastante bajo.

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train2.notas_m$calificacion,model.matrix(calificacion~., data=Train2.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_m$calificacion,model.matrix(calificacion~., data=Val2.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
```

```
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val2.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train2.notas_m$calificacion,model.matrix(calificacion~., data=Train2.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_m$calificacion,model.matrix(calificacion~., data=Val2.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val2.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train2.notas_m$calificacion,model.matrix(calificacion~., data=Train2.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val2.notas_m$calificacion,model.matrix(calificacion~., data=Val2.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val2.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
names(precisionNN_m)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas")
precisionNN_m
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas
##   0.2616822   0.2149533   0.2616822   0.2523364
```

El porcentaje de clasificación mediante redes neuronales de una capa, a pesar de ser mayor que en la asignatura de portugués, sigue siendo muy bajo y ni aumentando el número de neuronas se mejora notablemente.

Se prueba a continuación con una red neuronal de dos capas.

```
Train=data.frame(Train2.notas_m$calificacion,model.matrix(calificacion~., data=Train2.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn21=neuralnet(calificacion ~., data=Train, hidden=c(5,5), act.fct = "logistic", linear.output = FALSE)
plot(nn21)
```

```
Validate=data.frame(Val2.notas_m$calificacion,model.matrix(calificacion~., data=Val2.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn21,Validate)
predictedNN21=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN21<-confusionMatrix(Val2.notas_m$calificacion, predictedNN21)
matrizNN21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       13       62
##   suspenso       17       15
##
##                Accuracy : 0.2617
##                  95% CI : (0.1815, 0.3555)
##     No Information Rate : 0.7196
##     P-Value [Acc > NIR] : 1
##
```

```
##                  Kappa : -0.2551
##
##  Mcnemar's Test P-Value : 7.407e-07
##
##             Sensitivity : 0.4333
##             Specificity : 0.1948
##          Pos Pred Value : 0.1733
##          Neg Pred Value : 0.4688
##              Prevalence : 0.2804
##          Detection Rate : 0.1215
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.3141
##
##        'Positive' Class : aprobado
##
```

Con esta estructura la red neuronal tampoco mejora. Con otras que se ha probado pero no se muestran tampoco mejoró.

```
precision_m2<-c(precision_m2, max(precisionNN_m))
names(precision_m2)[2]<-c("Redes Neuronales")
```

Método 3: Máquina de vector soporte Asignatura: portugués

Se ajusta, a continuación, el modelo para los datos de la asignatura de portugués con el kernel radial.

```
fitsvm11 <-svm(calificacion ~., data = Train2.notas_p)
summary(fitsvm11)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train2.notas_p)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  138
##
##  ( 78 60 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm11,Val2.notas_p)
matrizSVM11<-confusionMatrix(Val2.notas_p$calificacion, predictedSVM)
matrizSVM11
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
```

```
##   aprobado      166        0
##   suspenso       23        1
##
##                Accuracy : 0.8789
##                  95% CI : (0.8239, 0.9217)
##     No Information Rate : 0.9947
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0706
##
##  Mcnemar's Test P-Value : 4.49e-06
##
##             Sensitivity : 0.87831
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 0.04167
##              Prevalence : 0.99474
##          Detection Rate : 0.87368
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.93915
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(matrizSVM11$overall[1])
names(precisionSVM_p)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm12 <-svm(calificacion ~., data = Train2.notas_p, kernel="polynomial")
summary(fitsvm12)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train2.notas_p, kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  169
##
##  ( 109 60 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm12,Val2.notas_p)
matrizSVM12<-confusionMatrix(Val2.notas_p$calificacion, predictedSVM)
```

```
matrizSVM12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM12$overall[1])
names(precisionSVM_p)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm13 <-svm(calificacion ~., data = Train2.notas_p, kernel="sigmoid")
summary(fitsvm13)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train2.notas_p, kernel = "sigmoid")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  sigmoid
##        cost:  1
##      coef.0:  0
##
## Number of Support Vectors:  136
##
##  ( 76 60 )
##
##
## Number of Classes:  2
```

```
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm13,Val2.notas_p)
matrizSVM13<-confusionMatrix(Val2.notas_p$calificacion, predictedSVM)
matrizSVM13
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM13$overall[1])
names(precisionSVM_p)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm14 <-svm(calificacion ~., data = Train2.notas_p, kernel="linear")
summary(fitsvm14)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train2.notas_p, kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  102
##
```

```
##  ( 62 40 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm14,Val2.notas_p)
matrizSVM14<-confusionMatrix(Val2.notas_p$calificacion, predictedSVM)
matrizSVM14
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      156       10
##   suspenso       11       13
##
##                Accuracy : 0.8895
##                  95% CI : (0.836, 0.9303)
##     No Information Rate : 0.8789
##     P-Value [Acc > NIR] : 0.379
##
##                   Kappa : 0.4902
##
##  Mcnemar's Test P-Value : 1.000
##
##             Sensitivity : 0.9341
##             Specificity : 0.5652
##          Pos Pred Value : 0.9398
##          Neg Pred Value : 0.5417
##              Prevalence : 0.8789
##          Detection Rate : 0.8211
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7497
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM14$overall[1])
names(precisionSVM_p)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenidos de los distintos kernel.

```
precisionSVM_p
```

```
##     radial polinomial  sigmoidal     lineal
##  0.8789474  0.8736842  0.8736842  0.8894737
```

La predicción de los SVM de kernel polinomial y sigmoidal es la misma. La clasificación de estos kernels es la peor de los cuatros. La mejor es la del kernel lineal.

```
precision_p2<-c(precision_p2, max(precisionSVM_p))
names(precision_p2)[3]<-c("SVM")
```

Asignatura: matemáticas

Se prueba primero con el kernel radial.

```
fitsvm21 <-svm(calificacion ~., data = Train2.notas_m)
predictedSVM = predict(fitsvm21,Val2.notas_m)
matrizSVM21<-confusionMatrix(Val2.notas_m$calificacion, predictedSVM)
matrizSVM21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       73        2
##   suspenso       22       10
##
##                Accuracy : 0.7757
##                  95% CI : (0.6849, 0.8507)
##     No Information Rate : 0.8879
##     P-Value [Acc > NIR] : 0.9997231
##
##                   Kappa : 0.3482
##
##  Mcnemar's Test P-Value : 0.0001052
##
##             Sensitivity : 0.7684
##             Specificity : 0.8333
##          Pos Pred Value : 0.9733
##          Neg Pred Value : 0.3125
##              Prevalence : 0.8879
##          Detection Rate : 0.6822
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8009
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(matrizSVM21$overall[1])
names(precisionSVM_m)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm22 <-svm(calificacion ~., data = Train2.notas_m, kernel="polynomial")
predictedSVM = predict(fitsvm22,Val2.notas_m)
matrizSVM22<-confusionMatrix(Val2.notas_m$calificacion, predictedSVM)
matrizSVM22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       75        0
##   suspenso       32        0
##
##                Accuracy : 0.7009
##                  95% CI : (0.6048, 0.7856)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
```

```
##
##                  Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 0.7009
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.7009
##    Detection Prevalence : 0.7009
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM22$overall[1])
names(precisionSVM_m)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm23 <-svm(calificacion ~., data = Train2.notas_m, kernel="sigmoid")
predictedSVM = predict(fitsvm23,Val2.notas_m)
matrizSVM23<-confusionMatrix(Val2.notas_m$calificacion, predictedSVM)
matrizSVM23
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       74        1
##   suspenso       29        3
##
##                Accuracy : 0.7196
##                  95% CI : (0.6245, 0.8022)
##     No Information Rate : 0.9626
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1073
##
##  Mcnemar's Test P-Value : 8.244e-07
##
##             Sensitivity : 0.71845
##             Specificity : 0.75000
##          Pos Pred Value : 0.98667
##          Neg Pred Value : 0.09375
##              Prevalence : 0.96262
##          Detection Rate : 0.69159
##    Detection Prevalence : 0.70093
##       Balanced Accuracy : 0.73422
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM23$overall[1])
names(precisionSVM_m)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm24 <-svm(calificacion ~., data = Train2.notas_m, kernel="linear")
predictedSVM = predict(fitsvm24,Val2.notas_m)
matrizSVM24<-confusionMatrix(Val2.notas_m$calificacion, predictedSVM)
matrizSVM24
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       66        9
##   suspenso       12       20
##
##                Accuracy : 0.8037
##                  95% CI : (0.7158, 0.8742)
##     No Information Rate : 0.729
##     P-Value [Acc > NIR] : 0.04804
##
##                   Kappa : 0.5189
##
##  Mcnemar's Test P-Value : 0.66252
##
##             Sensitivity : 0.8462
##             Specificity : 0.6897
##          Pos Pred Value : 0.8800
##          Neg Pred Value : 0.6250
##              Prevalence : 0.7290
##          Detection Rate : 0.6168
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.7679
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM24$overall[1])
names(precisionSVM_m)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenidos de los distintos kernel.

```
precisionSVM_m
```

```
##     radial polinomial  sigmoidal     lineal
##  0.7757009  0.7009346  0.7196262  0.8037383
```

La mejor clasificación es la del kernel lineal, seguida por la del kernel radial, luego el sigmoidal y por último el polinomial.

```
precision_m2<-c(precision_m2, max(precisionSVM_m))
names(precision_m2)[3]<-c("SVM")
```

Método 4: Naive Bayes Asignatura: portugués

```
fitbayes1 <-naiveBayes(calificacion ~., data = Train2.notas_p)
predictedBayes= predict(fitbayes1,Val2.notas_p)
matrizNB1<-confusionMatrix(Val2.notas_p$calificacion, predictedBayes)
matrizNB1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      149       17
##   suspenso        6       18
##
##                Accuracy : 0.8789
##                  95% CI : (0.8239, 0.9217)
##     No Information Rate : 0.8158
##     P-Value [Acc > NIR] : 0.01259
##
##                   Kappa : 0.5414
##
##  Mcnemar's Test P-Value : 0.03706
##
##             Sensitivity : 0.9613
##             Specificity : 0.5143
##          Pos Pred Value : 0.8976
##          Neg Pred Value : 0.7500
##              Prevalence : 0.8158
##          Detection Rate : 0.7842
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7378
##
##        'Positive' Class : aprobado
##
```

```
precision_p2<-c(precision_p2, matrizNB1$overall[1])
names(precision_p2)[4]<-c("Naive Bayes")
```

Asignatura: matemáticas

```
fitbayes2 <-naiveBayes(calificacion ~., data = Train2.notas_m)
predictedBayes= predict(fitbayes2,Val2.notas_m)
matrizNB2<-confusionMatrix(Val2.notas_m$calificacion, predictedBayes)
matrizNB2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       68        7
##   suspenso       16       16
##
##                Accuracy : 0.785
##                  95% CI : (0.6951, 0.8586)
##     No Information Rate : 0.785
##     P-Value [Acc > NIR] : 0.55555
##
```

```
##                   Kappa : 0.4423
##
##  Mcnemar's Test P-Value : 0.09529
##
##             Sensitivity : 0.8095
##             Specificity : 0.6957
##          Pos Pred Value : 0.9067
##          Neg Pred Value : 0.5000
##              Prevalence : 0.7850
##          Detection Rate : 0.6355
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.7526
##
##        'Positive' Class : aprobado
##
```

```
precision_m2<-c(precision_m2, matrizNB2$overall[1])
names(precision_m2)[4]<-c("Naive Bayes")
```

Método 5: Árboles de clasificación Asignatura: portugués

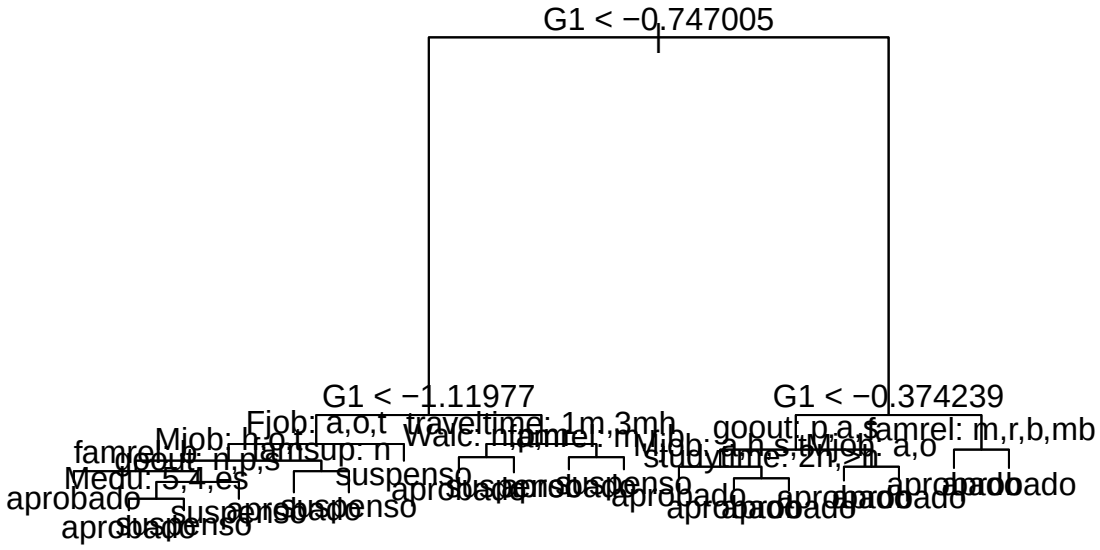
```
require(tree)
tree11 = tree(calificacion~., data = Train2.notas_p)
summary(tree11)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train2.notas_p)
## Variables actually used in tree construction:
##  [1] "G1"         "Fjob"       "Mjob"       "famrel"     "goout"
##  [6] "Medu"       "famsup"     "traveltime" "Walc"       "studytime"
## Number of terminal nodes:  18
## Residual mean deviance:  0.2046 = 86.96 / 425
## Misclassification error rate: 0.04515 = 20 / 443
```

```
plot(tree11)
text(tree11, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII

## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

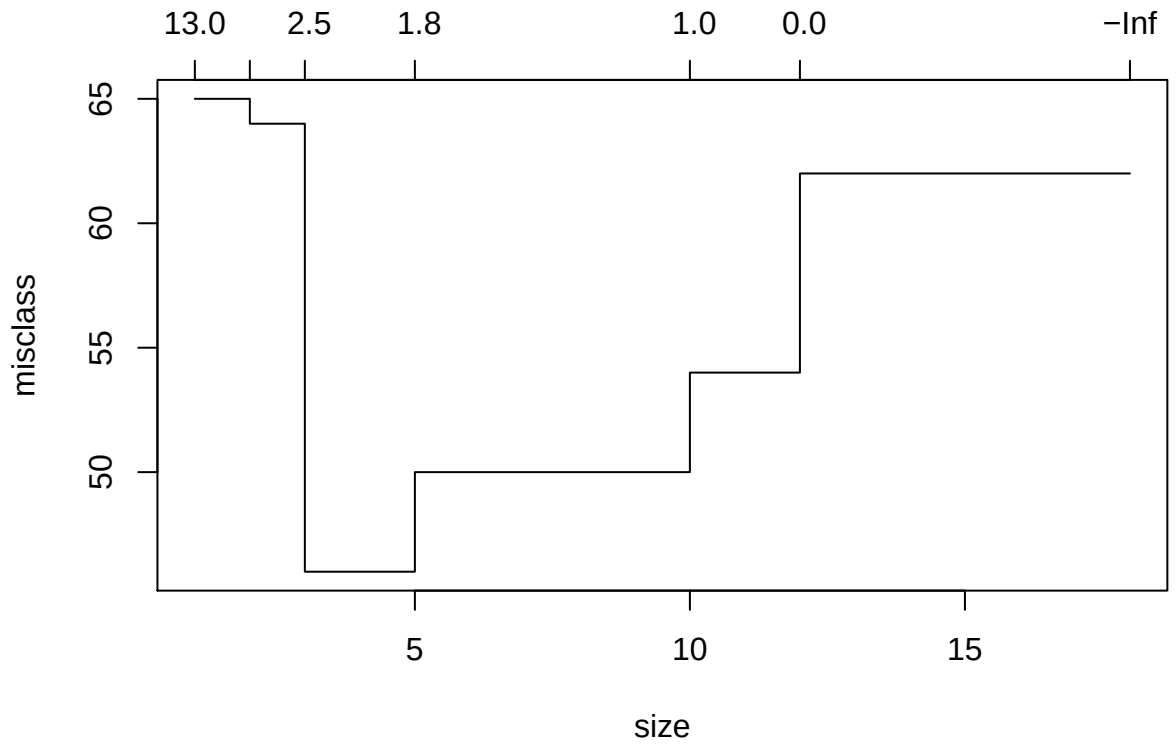
```
predicedtree = predict(tree11, Val2.notas_p, type="class")
matriztree11<-confusionMatrix(Val2.notas_p$calificacion, predicedtree)
matriztree11
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      157        9
##   suspenso       15        9
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 0.9053
##     P-Value [Acc > NIR] : 0.9414
##
##                   Kappa : 0.3592
##
##  Mcnemar's Test P-Value : 0.3074
##
##             Sensitivity : 0.9128
##             Specificity : 0.5000
##          Pos Pred Value : 0.9458
##          Neg Pred Value : 0.3750
##              Prevalence : 0.9053
##          Detection Rate : 0.8263
```

```
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7064
##
##        'Positive' Class : aprobado
##
```

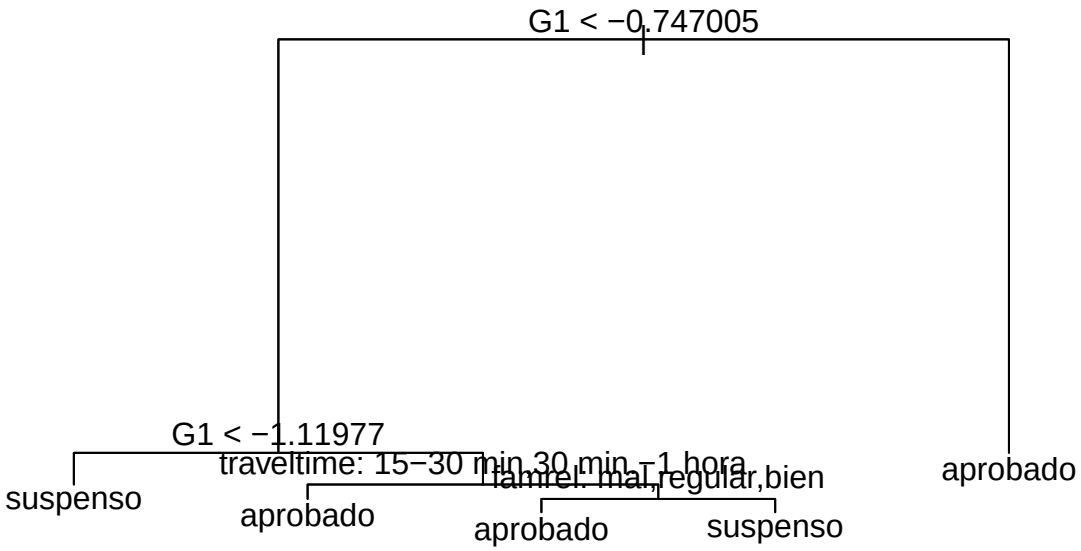
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree11 = cv.tree(tree11, FUN = prune.misclass)
plot(cv.tree11)
```



Se observa como o al tener muy pocas ramas o al aumentar el tamaño del árbol a más de cinco el error aumenta. Por ello, se elige que tenga 5 ramas.

```
prune.tree11 = prune.misclass(tree11, best = 5)
plot(prune.tree11)
text(prune.tree11, pretty=0)
```



Se observa que las ramas corresponden a las variables: G1, traveltime y famrel.

```
predicedtree12 = predict(prune.tree11, Val2.notas_p, type="class")
matriztree12<-confusionMatrix(Val2.notas_p$calificacion, predicedtree12)
matriztree12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      159        7
##   suspenso        7       17
##
##                Accuracy : 0.9263
##                  95% CI : (0.8795, 0.9591)
##     No Information Rate : 0.8737
##     P-Value [Acc > NIR] : 0.01434
##
##                   Kappa : 0.6662
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.9578
##             Specificity : 0.7083
##          Pos Pred Value : 0.9578
##          Neg Pred Value : 0.7083
##              Prevalence : 0.8737
```

```
##          Detection Rate : 0.8368
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.8331
##
##        'Positive' Class : aprobado
##
```

```
precision_p2<-c(precision_p2, matriztree12$overall[1])
names(precision_p2)[5]<-c("Arbol de clasificación")
```

Asignatura: matemáticas

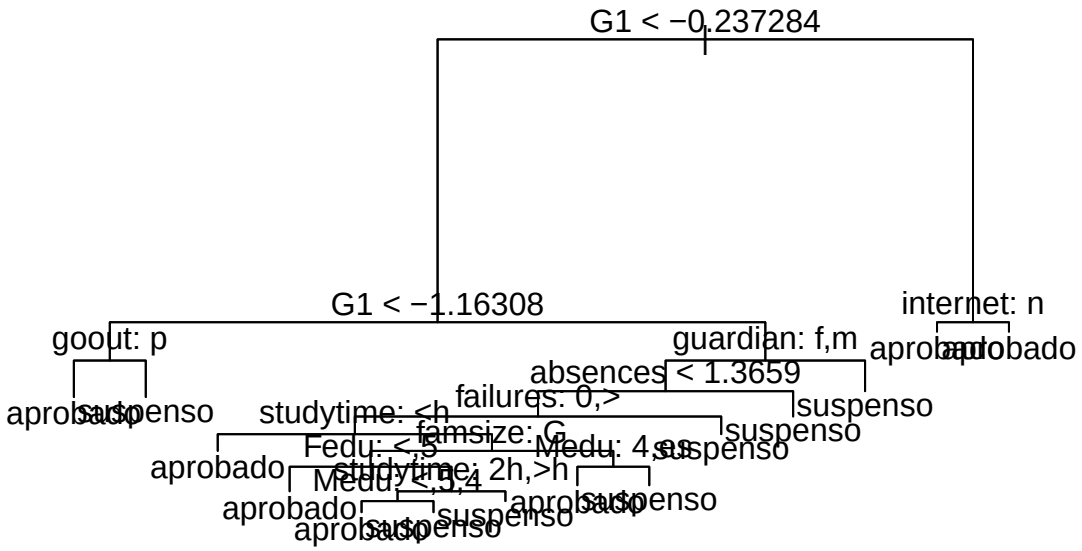
```
tree21 = tree(calificacion~., data = Train2.notas_m)
summary(tree21)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train2.notas_m)
## Variables actually used in tree construction:
##  [1] "G1"        "goout"     "guardian"  "absences"  "failures"  "studytime"
##  [7] "famsize"   "Fedu"      "Medu"      "internet"
## Number of terminal nodes:  14
## Residual mean deviance:  0.2304 = 54.38 / 236
## Misclassification error rate: 0.052 = 13 / 250
```

```
plot(tree21)
text(tree21, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII

## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

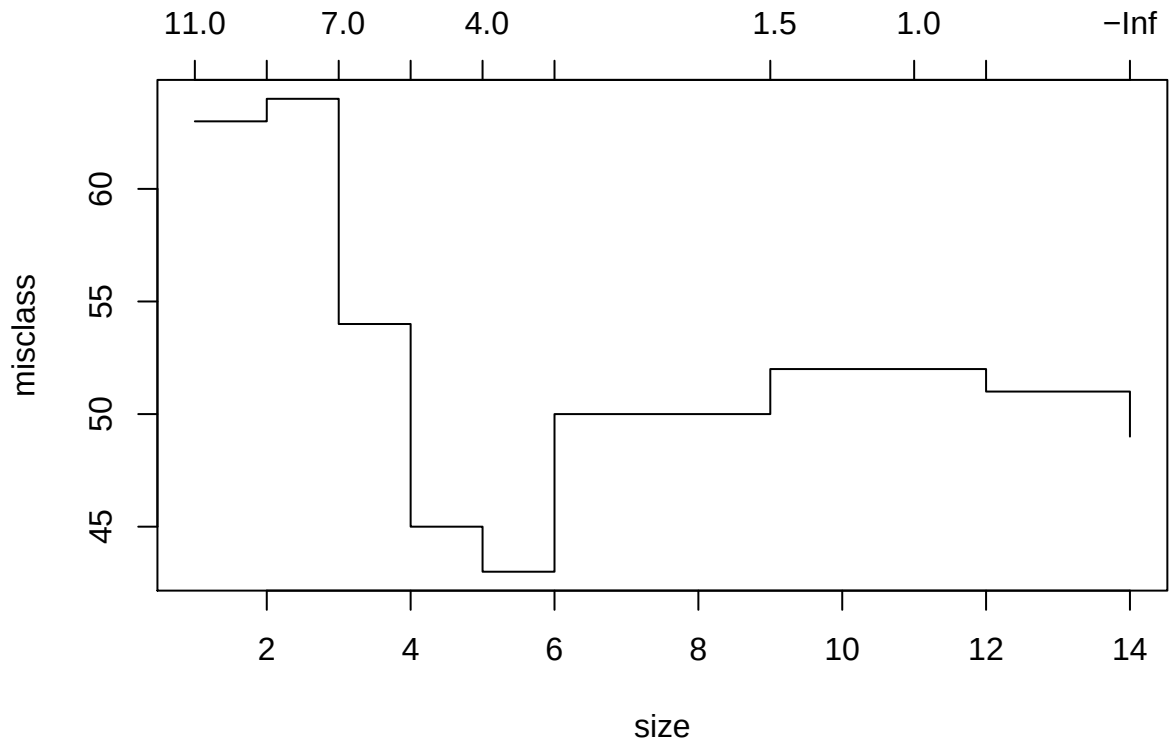
```
predicedtree = predict(tree21, Val2.notas_m, type="class")
matriztree21<-confusionMatrix(Val2.notas_m$calificacion, predicedtree)
matriztree21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       69        6
##   suspenso       13       19
##
##                Accuracy : 0.8224
##                  95% CI : (0.7367, 0.8896)
##     No Information Rate : 0.7664
##     P-Value [Acc > NIR] : 0.1019
##
##                   Kappa : 0.5481
##
##  Mcnemar's Test P-Value : 0.1687
##
##             Sensitivity : 0.8415
##             Specificity : 0.7600
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.5938
##              Prevalence : 0.7664
##          Detection Rate : 0.6449
```

```
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8007
##
##        'Positive' Class : aprobado
##
```

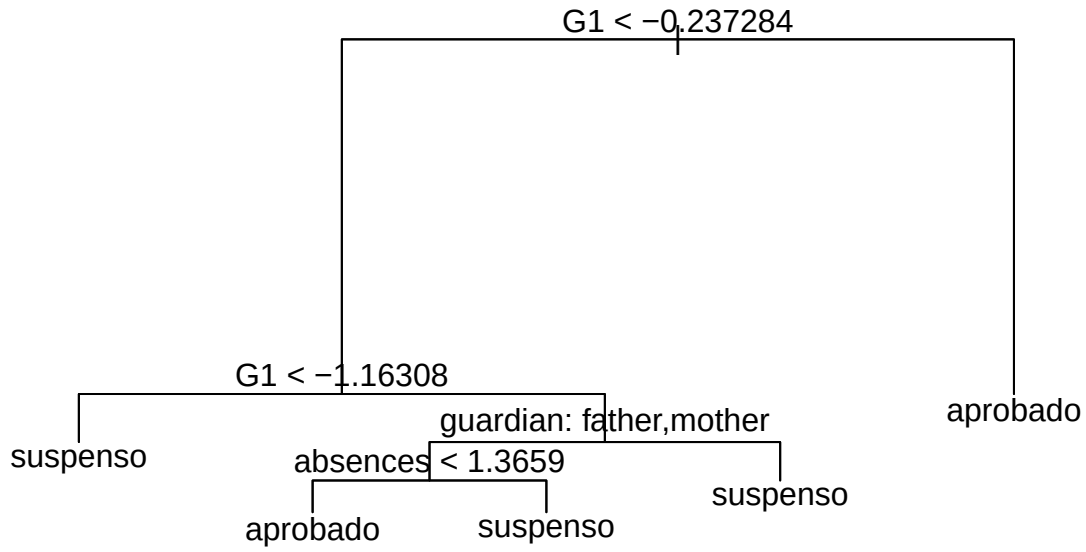
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree21 = cv.tree(tree21, FUN = prune.misclass)
plot(cv.tree21)
```



Se observa como al tener muy pocas ramas, el porcentaje de error aumenta. Se elige que tenga 5 ramas que es el número de ramas con menor errores y a partir del cual el error vuelve a crecer.

```
prune.tree21 = prune.misclass(tree21, best = 5)
plot(prune.tree21)
text(prune.tree21, pretty=0)
```



Se observa que las ramas corresponden a G1, guardian y absences.

```
predicedtree22 = predict(prune.tree21, Val2.notas_m, type="class")
matriztree22<-confusionMatrix(Val2.notas_m$calificacion, predicedtree22)
matriztree22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       72        3
##   suspenso       14       18
##
##                Accuracy : 0.8411
##                  95% CI : (0.7579, 0.9046)
##     No Information Rate : 0.8037
##     P-Value [Acc > NIR] : 0.19898
##
##                   Kappa : 0.5796
##
##  Mcnemar's Test P-Value : 0.01529
##
##             Sensitivity : 0.8372
##             Specificity : 0.8571
##          Pos Pred Value : 0.9600
##          Neg Pred Value : 0.5625
##              Prevalence : 0.8037
```

```
##          Detection Rate : 0.6729
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8472
##
##        'Positive' Class : aprobado
##
```

```
precision_m2<-c(precision_m2, matriztree22$overall[1])
names(precision_m2)[5]<-c("Arbol de clasificación")
```

Escenario 3: con G1 y con G2

Método 1: Regresión logística Asignatura: portugués

Primero se ajusta al modelo completo.

```
gfit1=glm(calificacion~., data=notas_p[,!(names(notas_p) %in% c("G3"))], family=binomial)
```

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

```
summary(gfit1)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_p[,
##     !(names(notas_p) %in% c("G3"))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8331  -0.0074  -0.0001   0.0000   3.8137
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -31.89739 2717.22131  -0.012   0.9906
## schoolMS                    2.34042    1.25421   1.866   0.0620 .
## sexhombre                   2.02623    1.28143   1.581   0.1138
## age                        -0.71906    0.67648  -1.063   0.2878
## addressRural               -0.23819    1.02190  -0.233   0.8157
## famsizeLE3                  0.24492    1.05799   0.231   0.8169
## Pstatusseparados           -2.02064    1.51903  -1.330   0.1834
## Medu<=4ºEP                 13.40163 2041.63310   0.007   0.9948
## Medu5ºEP-3ºESO             12.02158 2041.63355   0.006   0.9953
## Medu4ºESO-2ºBachiller      12.08746 2041.63353   0.006   0.9953
## Meduestudios superiores    12.68279 2041.63518   0.006   0.9950
## Fedu<=4ºEP                 14.62781 1793.04471   0.008   0.9935
## Fedu5ºEP-3ºESO             11.99760 1793.04503   0.007   0.9947
## Fedu4ºESO-2ºBachiller      13.91930 1793.04587   0.008   0.9938
## Feduestudios superiores    13.61431 1793.04537   0.008   0.9939
## Mjobhealth                 -0.53373    2.09414  -0.255   0.7988
## Mjobother                  -2.16536    1.26319  -1.714   0.0865 .
## Mjobservices               -0.30818    1.64601  -0.187   0.8515
## Mjobteacher                -3.46712    3.06622  -1.131   0.2582
## Fjobhealth                  2.89423    3.15331   0.918   0.3587
## Fjobother                   2.22657    2.53346   0.879   0.3795
## Fjobservices                1.71012    2.57652   0.664   0.5069
## Fjobteacher                 1.80707    3.65198   0.495   0.6207
```

```
## reasonhome                 -1.22229    1.64665  -0.742   0.4579
## reasonother                 0.06034    1.39971   0.043   0.9656
## reasonreputation           -1.22707    1.61436  -0.760   0.4472
## guardianmother              1.25595    1.37304   0.915   0.3603
## guardianother               1.77716    2.04477   0.869   0.3848
## traveltime15-30 min        -3.06447    1.25419  -2.443   0.0146 *
## traveltime30 min.-1 hora   -1.36224    1.58912  -0.857   0.3913
## traveltime>1 hora          -3.85864    2.44119  -1.581   0.1140
## studytime2-5 horas         -1.31457    1.07932  -1.218   0.2232
## studytime5-10 horas         1.12018    2.03919   0.549   0.5828
## studytime>10 horas         -4.58959    4.35945  -1.053   0.2924
## failures1                  -0.38075    1.32187  -0.288   0.7733
## failures2                  -1.89614    2.08626  -0.909   0.3634
## failures>=3                 1.53616    1.71562   0.895   0.3706
## schoolsupyes                0.64794    1.50876   0.429   0.6676
## famsupyes                  -0.03192    0.92245  -0.035   0.9724
## paidyes                     4.05601    1.89234   2.143   0.0321 *
## activitiesyes              -0.08533    0.94501  -0.090   0.9281
## nurseryyes                  0.33893    1.05366   0.322   0.7477
## higheryes                  -1.82638    1.29667  -1.409   0.1590
## internetyes                 1.08628    1.23936   0.876   0.3808
## romanticyes                -0.86238    1.20445  -0.716   0.4740
## famrelmal                 -13.15390    5.50753  -2.388   0.0169 *
## famrelregular              -4.22757    2.90650  -1.455   0.1458
## famrelbien                 -6.49818    3.27768  -1.983   0.0474 *
## famrelmuy bien             -5.19097    3.30374  -1.571   0.1161
## freetimepoco               -1.23470    2.01705  -0.612   0.5405
## freetimealgo               -2.46212    1.96194  -1.255   0.2095
## freetimesuficiente         -2.15958    1.70793  -1.264   0.2061
## freetimemucho              -1.75253    1.86454  -0.940   0.3473
## gooutpoco                  -0.32028    1.97224  -0.162   0.8710
## gooutalgo                  -1.71772    1.69066  -1.016   0.3096
## gooutsuficiente            -0.93425    1.88841  -0.495   0.6208
## gooutmucho                  1.46036    2.07756   0.703   0.4821
## Dalcpoco                    0.83172    1.48507   0.560   0.5754
## Dalcalgo                   -0.30749    1.76953  -0.174   0.8620
## Dalcsuficiente             -6.39680    4.73560  -1.351   0.1768
## Dalcmucho                   1.77021    3.12636   0.566   0.5712
## Walcpoco                   -0.45750    1.56511  -0.292   0.7700
## Walcalgo                    0.19665    1.49797   0.131   0.8956
## Walcsuficiente             -1.23150    1.93492  -0.636   0.5245
## Walcmucho                  -1.58475    2.42086  -0.655   0.5127
## healthmal                  -0.55063    1.76420  -0.312   0.7550
## healthregular               0.51727    1.96612   0.263   0.7925
## healthbien                  2.24925    1.93816   1.161   0.2458
## healthmuy bien              1.87715    1.68036   1.117   0.2639
## absences                    0.45932    0.57328   0.801   0.4230
## G1                         -3.35951    1.11540  -3.012   0.0026 **
## G2                         -8.88991    2.23940  -3.970 7.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 495.626  on 632  degrees of freedom
## Residual deviance:  91.878  on 561  degrees of freedom
## AIC: 235.88
##
## Number of Fisher Scoring iterations: 17
```

Resultan significativas las siguientes variables: traveltime, paid, famrel y, especialmente significativas como era de esperar, G1 y G2.

Sin embargo, lo que interesa es la predicción.

```
gfit12=glm(calificacion~., data=Train3.notas_p, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
cbind(gfit1$coefficients, gfit12$coefficients)
```

```
##                                 [,1]         [,2]
## (Intercept)              -31.89738951  258.4286644
## schoolMS                   2.34042317    1.3728736
## sexhombre                  2.02622755   23.3371352
## age                       -0.71905701   12.8354455
## addressRural              -0.23819380  -12.2731118
## famsizeLE3                 0.24492168  -12.7601380
## Pstatusseparados          -2.02064203  -84.9746422
## Medu<=4ºEP                13.40162653 -186.9415231
## Medu5ºEP-3ºESO            12.02157920 -233.4570981
## Medu4ºESO-2ºBachiller     12.08745876 -202.7890166
## Meduestudios superiores   12.68279305 -199.5147432
## Fedu<=4ºEP                14.62781202   62.9955200
## Fedu5ºEP-3ºESO            11.99759733   -1.4804836
## Fedu4ºESO-2ºBachiller     13.91929786    1.9658498
## Feduestudios superiores   13.61430893   74.1420772
## Mjobhealth                -0.53372667   43.6015838
## Mjobother                 -2.16535647  -28.7745790
## Mjobservices              -0.30818023   22.0663329
## Mjobteacher               -3.46712454  -28.7355119
## Fjobhealth                 2.89423114   69.6695158
## Fjobother                  2.22657018   58.8464511
## Fjobservices               1.71012187   60.4760469
## Fjobteacher                1.80707218    4.6772694
## reasonhome                -1.22228607  -26.9557658
## reasonother                0.06034011   29.2328187
## reasonreputation          -1.22707394  -36.3750075
## guardianmother             1.25594568   16.4670497
## guardianother              1.77716226  -29.0302035
## traveltime15-30 min       -3.06446575  -44.2156054
## traveltime30 min.-1 hora  -1.36224170  -27.6995938
## traveltime>1 hora         -3.85863585  -49.9827936
## studytime2-5 horas        -1.31457488  -26.4931419
## studytime5-10 horas        1.12017608   -0.7510549
## studytime>10 horas        -4.58958772 -112.6945094
## failures1                 -0.38074913  -50.2045677
## failures2                 -1.89614197  -33.1316987
## failures>=3                1.53616172    2.8534835
```

```
## schoolsupyes               0.64794332  -80.9338584
## famsupyes                 -0.03192399   23.9167637
## paidyes                    4.05600944   28.7054158
## activitiesyes             -0.08532794    1.1462520
## nurseryyes                 0.33893285   55.2920331
## higheryes                 -1.82638030  -52.5086219
## internetyes                1.08628351   -1.3354273
## romanticyes               -0.86237621  -15.8635872
## famrelmal                -13.15390461 -210.3816215
## famrelregular             -4.22757029  -63.6982582
## famrelbien                -6.49818413 -126.9187656
## famrelmuy bien            -5.19097048 -129.0864408
## freetimepoco              -1.23470094  -56.2434428
## freetimealgo              -2.46211509  -22.7088722
## freetimesuficiente        -2.15957506  -14.9770479
## freetimemucho             -1.75252604  -47.8002909
## gooutpoco                 -0.32027626  -96.1400591
## gooutalgo                 -1.71772190  -71.1843291
## gooutsuficiente           -0.93424943  -55.9618005
## gooutmucho                 1.46036007  -64.1883763
## Dalcpoco                   0.83172135   20.5266614
## Dalcalgo                  -0.30748694  -37.1984918
## Dalcsuficiente            -6.39679990 -106.0905599
## Dalcmucho                  1.77020695   -5.9051010
## Walcpoco                  -0.45750288  -46.8337849
## Walcalgo                   0.19664668   -0.7405896
## Walcsuficiente            -1.23150305   -8.2844288
## Walcmucho                 -1.58474780   23.3465236
## healthmal                 -0.55062722  -44.1494082
## healthregular              0.51727100  -24.8165595
## healthbien                 2.24924995  -30.7985478
## healthmuy bien             1.87715005   -7.3430654
## absences                   0.45932371    4.7371140
## G1                        -3.35950951  -37.4046196
## G2                        -8.88991145 -147.0807517
```

Los coeficientes cambian drásticamente.

```
p=predict(gfit12, Val3.notas_p, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
matrizLogis<-confusionMatrix(Val3.notas_p$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      154       12
##   suspenso        9       15
##
##                Accuracy : 0.8895
##                  95% CI : (0.836, 0.9303)
##     No Information Rate : 0.8579
##     P-Value [Acc > NIR] : 0.1246
```

```
##
##                   Kappa : 0.5247
##
##  Mcnemar's Test P-Value : 0.6625
##
##             Sensitivity : 0.9448
##             Specificity : 0.5556
##          Pos Pred Value : 0.9277
##          Neg Pred Value : 0.6250
##              Prevalence : 0.8579
##          Detection Rate : 0.8105
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7502
##
##        'Positive' Class : aprobado
##
```

El porcentaje de clasificación correcta es del 89%.

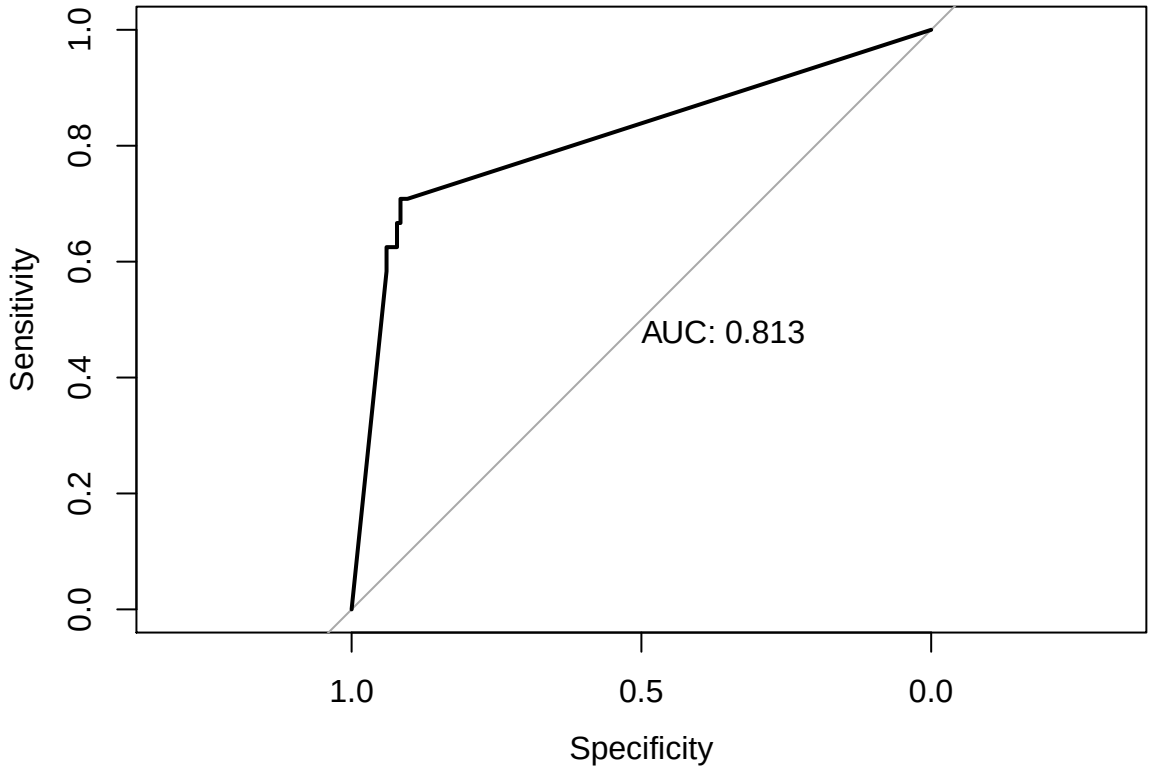
```
precision_p3<-c(matrizLogis$overall[1])
names(precision_p3)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC.

```
test_prob = predict(gfit12, newdata = Val3.notas_p, type = "response")
test_roc = roc(Val3.notas_p$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0,813, ligeramente menor que en el escenario anterior, pero que es un valor alto y por tanto confirma que el modelo es bueno.

Asignatura: matemáticas

Primero se ajusta al modelo completo.

```
gfit2=glm(calificacion~., data=notas_m[,!(names(notas_m) %in% c("G3"))], family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
summary(gfit2)
```

```
##
## Call:
## glm(formula = calificacion ~ ., family = binomial, data = notas_m[,
##     !(names(notas_m) %in% c("G3"))])
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -6.318e-05  -2.100e-08  -2.100e-08   2.100e-08   5.402e-05
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                9.197e+01  8.334e+05   0.000    1.000
## schoolMS                  -4.572e+01  1.543e+05   0.000    1.000
## sexhombre                  9.968e+00  8.009e+04   0.000    1.000
## age                        1.666e+01  7.279e+04   0.000    1.000
```

```
## addressRural             -1.201e+01  1.066e+05   0.000    1.000
## famsizeLE3                5.451e+01  5.228e+04   0.001    0.999
## Pstatusseparados          2.660e+01  8.264e+04   0.000    1.000
## Medu<=4ºEP               -1.654e+02  4.742e+05   0.000    1.000
## Medu5ºEP-3ºESO           -1.656e+02  4.993e+05   0.000    1.000
## Medu4ºESO-2ºBachiller    -1.377e+02  5.285e+05   0.000    1.000
## Meduestudios superiores  -1.691e+02  5.427e+05   0.000    1.000
## Fedu<=4ºEP               -8.368e+00  3.239e+05   0.000    1.000
## Fedu5ºEP-3ºESO           -5.202e+00  2.718e+05   0.000    1.000
## Fedu4ºESO-2ºBachiller     1.698e+01  2.398e+05   0.000    1.000
## Feduestudios superiores   1.415e+01  2.438e+05   0.000    1.000
## Mjobhealth                2.270e+00  2.649e+05   0.000    1.000
## Mjobother                 4.813e+01  1.368e+05   0.000    1.000
## Mjobservices             -8.725e-01  2.108e+05   0.000    1.000
## Mjobteacher               1.550e+01  1.712e+05   0.000    1.000
## Fjobhealth                1.722e+01  1.439e+05   0.000    1.000
## Fjobother                -6.227e+01  1.153e+05  -0.001    1.000
## Fjobservices              4.296e+00  1.446e+05   0.000    1.000
## Fjobteacher              -4.526e+01  2.557e+05   0.000    1.000
## reasonhome               -4.053e+01  9.351e+04   0.000    1.000
## reasonother               8.753e+00  1.276e+05   0.000    1.000
## reasonreputation         -2.879e+01  1.052e+05   0.000    1.000
## guardianmother            3.212e+00  5.482e+04   0.000    1.000
## guardianother             8.537e+00  3.118e+05   0.000    1.000
## traveltime15-30 min       3.385e+01  9.799e+04   0.000    1.000
## traveltime30 min.-1 hora -1.085e+00  2.480e+05   0.000    1.000
## traveltime>1 hora        -4.595e+01  5.138e+05   0.000    1.000
## studytime2-5 horas       -5.766e+00  1.357e+05   0.000    1.000
## studytime5-10 horas       2.553e+01  1.223e+05   0.000    1.000
## studytime>10 horas        3.106e+01  2.373e+05   0.000    1.000
## failures1                -1.563e+01  1.235e+05   0.000    1.000
## failures2                 6.000e+00  3.977e+05   0.000    1.000
## failures>=3               1.767e+00  1.234e+05   0.000    1.000
## schoolsupyes             -1.298e+01  7.523e+04   0.000    1.000
## famsupyes                 2.248e+01  1.553e+05   0.000    1.000
## paidyes                  -1.831e+01  9.221e+04   0.000    1.000
## activitiesyes             2.974e+00  9.107e+04   0.000    1.000
## nurseryyes                1.444e+01  9.493e+04   0.000    1.000
## higheryes                 2.075e+01  4.330e+05   0.000    1.000
## internetyes               8.225e+00  5.385e+04   0.000    1.000
## romanticyes               3.235e+00  1.435e+05   0.000    1.000
## famrelmal                -1.395e+02  2.735e+05  -0.001    1.000
## famrelregular            -1.608e+02  2.230e+05  -0.001    0.999
## famrelbien               -1.917e+02  2.412e+05  -0.001    0.999
## famrelmuy bien           -2.189e+02  2.075e+05  -0.001    0.999
## freetimepoco             -7.646e+00  1.408e+05   0.000    1.000
## freetimealgo             -2.536e+01  1.652e+05   0.000    1.000
## freetimesuficiente        2.481e+01  1.902e+05   0.000    1.000
## freetimemucho             2.080e+01  1.407e+05   0.000    1.000
## gooutpoco                 9.208e+01  2.106e+05   0.000    1.000
## gooutalgo                 9.856e+01  2.403e+05   0.000    1.000
## gooutsuficiente           8.511e+01  2.469e+05   0.000    1.000
## gooutmucho                8.818e+01  2.341e+05   0.000    1.000
## Dalcpoco                  5.331e+00  9.452e+04   0.000    1.000
```

```
## Dalcalgo                -4.729e+01  1.925e+05   0.000    1.000
## Dalcsuficiente           3.680e+01  2.262e+05   0.000    1.000
## Dalcmucho               -3.905e+01  4.459e+05   0.000    1.000
## Walcpoco                -5.787e+01  1.332e+05   0.000    1.000
## Walcalgo                -5.158e+00  1.561e+05   0.000    1.000
## Walcsuficiente          -5.522e+01  2.248e+05   0.000    1.000
## Walcmucho               -2.382e+01  1.778e+05   0.000    1.000
## healthmal                1.640e+01  2.355e+05   0.000    1.000
## healthregular            1.646e+01  1.920e+05   0.000    1.000
## healthbien               1.516e+01  1.289e+05   0.000    1.000
## healthmuy bien           4.731e+01  1.565e+05   0.000    1.000
## absences                 3.008e+00  4.688e+04   0.000    1.000
## G1                       1.314e+01  1.346e+05   0.000    1.000
## G2                      -1.688e+02  1.359e+05  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.0744e+02  on 356  degrees of freedom
## Residual deviance: 6.0686e-08  on 285  degrees of freedom
## AIC: 144
##
## Number of Fisher Scoring iterations: 25
```

El modelo no resulta válido. Puede haber afectado la colinealidad entre las notas: G1, G2 y G3.

A continuación la predicción.

```
gfit21=glm(calificacion~., data=Train3.notas_m, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
cbind(gfit1$coefficients, gfit21$coefficients)
```

```
##                                 [,1]         [,2]
## (Intercept)              -31.89738951  166.2045609
## schoolMS                   2.34042317  -31.4521947
## sexhombre                  2.02622755   18.8882420
## age                       -0.71905701   -3.7440674
## addressRural              -0.23819380  -10.8672240
## famsizeLE3                 0.24492168   19.5827662
## Pstatusseparados          -2.02064203   39.4469448
## Medu<=4ºEP                13.40162653 -166.6224005
## Medu5ºEP-3ºESO            12.02157920 -187.1639944
## Medu4ºESO-2ºBachiller     12.08745876 -155.1366473
## Meduestudios superiores   12.68279305 -179.2288954
## Fedu<=4ºEP                14.62781202   -7.3273948
## Fedu5ºEP-3ºESO            11.99759733   -8.6172752
## Fedu4ºESO-2ºBachiller     13.91929786  -39.5688634
## Feduestudios superiores   13.61430893  -27.7152168
## Mjobhealth                -0.53372667  -16.2866371
## Mjobother                 -2.16535647   20.4048405
## Mjobservices              -0.30818023  -16.8649144
## Mjobteacher               -3.46712454   14.1730056
## Fjobhealth                 2.89423114   14.0758806
## Fjobother                  2.22657018  -12.6860245
```

```
## Fjobservices               1.71012187   -7.6184039
## Fjobteacher                1.80707218   18.6368242
## reasonhome                -1.22228607  -35.3719394
## reasonother                0.06034011  -56.8840776
## reasonreputation          -1.22707394  -35.1053414
## guardianmother             1.25594568    1.5150775
## guardianother              1.77716226   48.8714088
## traveltime15-30 min       -3.06446575   -8.6836413
## traveltime30 min.-1 hora  -1.36224170   -1.3563116
## traveltime>1 hora         -3.85863585  -36.5623073
## studytime2-5 horas        -1.31457488  -15.5618252
## studytime5-10 horas        1.12017608    3.6464088
## studytime>10 horas        -4.58958772  -58.0643346
## failures1                 -0.38074913  -25.4679077
## failures2                 -1.89614197   -3.9872920
## failures>=3                1.53616172    3.7303446
## schoolsupyes               0.64794332    7.9416381
## famsupyes                 -0.03192399   17.1991715
## paidyes                    4.05600944   17.4489040
## activitiesyes             -0.08532794   12.7456007
## nurseryyes                 0.33893285    2.2954595
## higheryes                 -1.82638030  -30.9736162
## internetyes                1.08628351   -7.5804156
## romanticyes               -0.86237621   29.1926618
## famrelmal                -13.15390461  -41.2319444
## famrelregular             -4.22757029  -69.5889033
## famrelbien                -6.49818413  -92.7256380
## famrelmuy bien            -5.19097048 -116.9007876
## freetimepoco              -1.23470094  -31.1775589
## freetimealgo              -2.46211509  -27.0610954
## freetimesuficiente        -2.15957506  -21.0288024
## freetimemucho             -1.75252604  -38.1769535
## gooutpoco                 -0.32027626   78.8712582
## gooutalgo                 -1.71772190  100.5669631
## gooutsuficiente           -0.93424943  101.4063514
## gooutmucho                 1.46036007  114.4032203
## Dalcpoco                   0.83172135    0.3267556
## Dalcalgo                  -0.30748694    0.8142449
## Dalcsuficiente            -6.39679990  137.1291376
## Dalcmucho                  1.77020695   73.7112332
## Walcpoco                  -0.45750288  -20.2344112
## Walcalgo                   0.19664668  -37.8435959
## Walcsuficiente            -1.23150305  -14.8228359
## Walcmucho                 -1.58474780 -123.1646948
## healthmal                 -0.55062722  -11.3389900
## healthregular              0.51727100   -3.9453162
## healthbien                 2.24924995   25.5304487
## healthmuy bien             1.87715005   -8.8840634
## absences                   0.45932371    7.6937576
## G1                        -3.35950951  -17.0932299
## G2                        -8.88991145  -73.8415592
```

Con este modelo, predecimos los valores de calificación en la asignatura de matemáticas.

```
p=predict(gfit21, Val3.notas_m, type="response")
PredCalificacion=as.factor(p>0.5)
levels(PredCalificacion)=c("aprobado", "suspenso")
matrizLogis<-confusionMatrix(Val3.notas_m$calificacion, PredCalificacion)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       67        8
##   suspenso       10       22
##
##                Accuracy : 0.8318
##                  95% CI : (0.7472, 0.8971)
##     No Information Rate : 0.7196
##     P-Value [Acc > NIR] : 0.004948
##
##                   Kappa : 0.5914
##
##  Mcnemar's Test P-Value : 0.813664
##
##             Sensitivity : 0.8701
##             Specificity : 0.7333
##          Pos Pred Value : 0.8933
##          Neg Pred Value : 0.6875
##              Prevalence : 0.7196
##          Detection Rate : 0.6262
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8017
##
##        'Positive' Class : aprobado
##
```

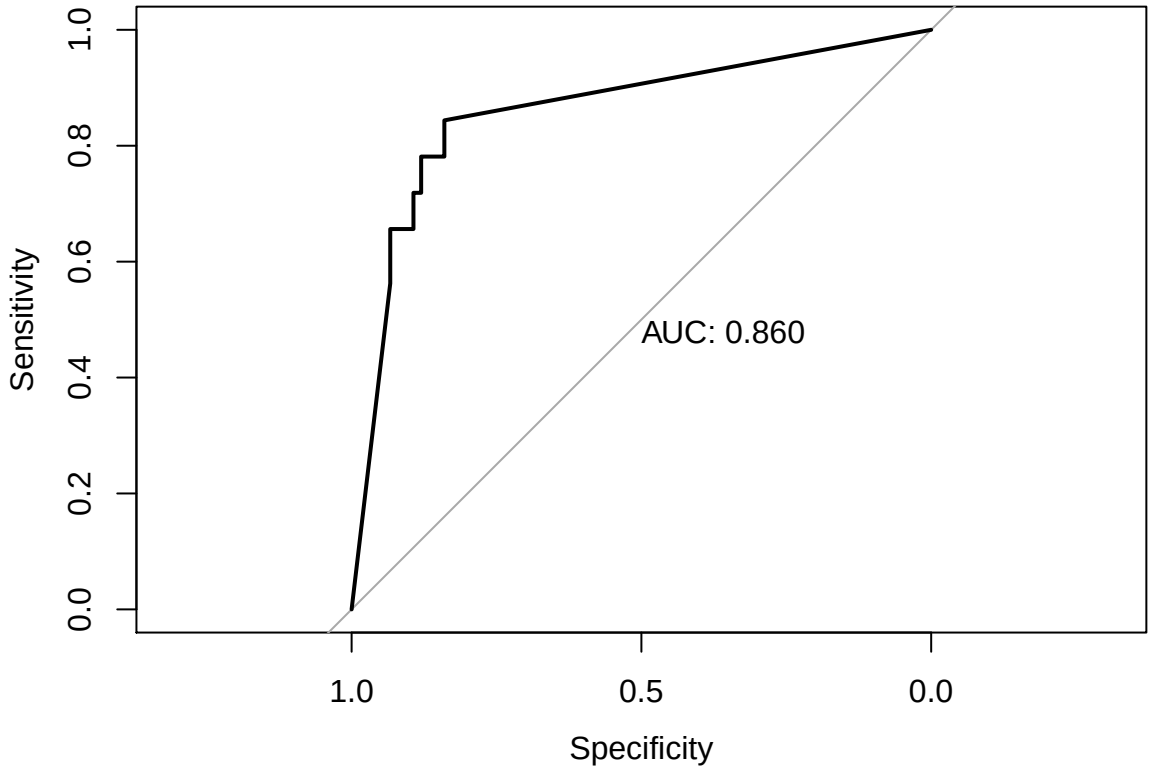
```
precision_m3<-c(matrizLogis$overall[1])
names(precision_m3)<-c("Regresion Logistica")
```

Se dibuja también la curva ROC para comprobar el modelo.

```
test_prob = predict(gfit21, newdata = Val3.notas_m, type = "response")
test_roc = roc(Val3.notas_m$calificacion ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = aprobado, case = suspenso
```

```
## Setting direction: controls < cases
```



El área bajo la curva es de 0.86, la cual es mayor que en el anterior escenario, siendo ya un valor notablemente alto.

Método 2: Redes neuronales Asignatura: portugués

Se prueba primero con una red neuronal de una capa y cinco neuronas.

```
Train=data.frame(Train3.notas_p$calificacion,model.matrix(calificacion~., data=Train3.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn1)
```

```
Validate=data.frame(Val3.notas_p$calificacion,model.matrix(calificacion~., data=Val3.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val3.notas_p$calificacion, predictedNN1)
matrizNN1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado        6      160
##   suspenso       14       10
##
##                Accuracy : 0.0842
##                  95% CI : (0.0489, 0.1332)
```

```
##     No Information Rate : 0.8947
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1519
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.30000
##             Specificity : 0.05882
##          Pos Pred Value : 0.03614
##          Neg Pred Value : 0.41667
##              Prevalence : 0.10526
##          Detection Rate : 0.03158
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.17941
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_p<-c(matrizNN1$overall[1])
```

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train3.notas_p$calificacion,model.matrix(calificacion~., data=Train3.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_p$calificacion,model.matrix(calificacion~., data=Val3.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val3.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train3.notas_p$calificacion,model.matrix(calificacion~., data=Train3.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_p$calificacion,model.matrix(calificacion~., data=Val3.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val3.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
Train=data.frame(Train3.notas_p$calificacion,model.matrix(calificacion~., data=Train3.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn1=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_p$calificacion,model.matrix(calificacion~., data=Val3.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn1,Validate)
predictedNN1=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN1<-confusionMatrix(Val3.notas_p$calificacion, predictedNN1)
precisionNN_p<-c(precisionNN_p, matrizNN1$overall[1])
names(precisionNN_p)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas")
precisionNN_p
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas
##  0.08421053  0.07894737  0.07894737  0.08421053
```

El porcentaje de clasificación mediante redes neuronales de una capa es muy bajo y ni aumentando el número de neuronas se mejora..

Se prueba a continuación con una red neuronal de dos capas.

```
Train=data.frame(Train3.notas_p$calificacion,model.matrix(calificacion~., data=Train3.notas_p)[,-1])
colnames(Train)[1]="calificacion"
nn12=neuralnet(calificacion ~., data=Train, hidden=c(5,3), act.fct = "logistic", linear.output = FALSE)
plot(nn12)
```

```
Validate=data.frame(Val3.notas_p$calificacion,model.matrix(calificacion~., data=Val3.notas_p)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn12,Validate)
predictedNN12=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN12<-confusionMatrix(Val3.notas_p$calificacion, predictedNN12)
matrizNN12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado        7      159
##   suspenso       13       11
##
##                Accuracy : 0.0947
##                  95% CI : (0.0571, 0.1456)
##     No Information Rate : 0.8947
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.1387
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.35000
##             Specificity : 0.06471
##          Pos Pred Value : 0.04217
##          Neg Pred Value : 0.45833
##              Prevalence : 0.10526
##          Detection Rate : 0.03684
##    Detection Prevalence : 0.87368
##       Balanced Accuracy : 0.20735
##
##        'Positive' Class : aprobado
##
```

De esta forma tampoco mejora la clasificación.

```
precision_p3<-c(precision_p3, max(precisionNN_p))
names(precision_p3)[2]<-c("Redes Neuronales")
```

Asignatura: Matemáticas

Se prueba primero con una red neuronal de una capa y cinco neuronas.

```
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=5, act.fct = "logistic", linear.output = FALSE)
plot(nn2)
```

```
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val3.notas_m$calificacion, predictedNN2)
matrizNN2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado        8       67
##   suspenso       24        8
##
##                Accuracy : 0.1495
##                  95% CI : (0.088, 0.2314)
##     No Information Rate : 0.7009
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.4644
##
##  Mcnemar's Test P-Value : 1.069e-05
##
##             Sensitivity : 0.25000
##             Specificity : 0.10667
##          Pos Pred Value : 0.10667
##          Neg Pred Value : 0.25000
##              Prevalence : 0.29907
##          Detection Rate : 0.07477
##    Detection Prevalence : 0.70093
##       Balanced Accuracy : 0.17833
##
##        'Positive' Class : aprobado
##
```

```
precisionNN_m<-c(matrizNN2$overall[1])
```

El porcentaje de clasificación correcta en la asignatura de matemáticas casi duplica al de la asignatura de portugués y con el mismo modelo al igual que en el escenario anterior. Sin embargo, sigue siendo bastante bajo.

Se prueba a continuación con distinto número de neuronas.

```
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=10, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val3.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=15, act.fct = "logistic", linear.output = FALSE)
```

```
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val3.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=20, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val3.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn2=neuralnet(calificacion ~., data=Train, hidden=30, act.fct = "logistic", linear.output = FALSE)
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn2,Validate)
predictedNN2=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN2<-confusionMatrix(Val3.notas_m$calificacion, predictedNN2)
precisionNN_m<-c(precisionNN_m, matrizNN2$overall[1])
names(precisionNN_m)<-c("5 neuronas", "10 neuronas", "15 neuronas", "20 neuronas","30 neuronas")
precisionNN_m
```

```
##  5 neuronas 10 neuronas 15 neuronas 20 neuronas 30 neuronas
##   0.1495327   0.1775701   0.1775701   0.1869159   0.1495327
```

El porcentaje de clasificación mediante redes neuronales de una capa, a pesar de ser mayor que en la asignatura de portugués, sigue siendo muy bajo y ni aumentando el número de neuronas se mejora notablemente, solo mejor ligeramente con 20 neuronas.

Se prueba a continuación con una red neuronal de dos capas y cinco neuronas cada una.

```
Train=data.frame(Train3.notas_m$calificacion,model.matrix(calificacion~., data=Train3.notas_m)[,-1])
colnames(Train)[1]="calificacion"
nn21=neuralnet(calificacion ~., data=Train, hidden=c(5,5), act.fct = "logistic", linear.output = FALSE)
plot(nn21)
```

```
Validate=data.frame(Val3.notas_m$calificacion,model.matrix(calificacion~., data=Val3.notas_m)[,-1])
colnames(Validate)[1]="calificacion"
Predict=compute(nn21,Validate)
predictedNN21=factor(Predict$net.result[,1]>0.5, labels = c("aprobado", "suspenso"))
matrizNN21<-confusionMatrix(Val3.notas_m$calificacion, predictedNN21)
matrizNN21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       11       64
##   suspenso       22       10
##
##                Accuracy : 0.1963
```

```
##                  95% CI : (0.1258, 0.2842)
##     No Information Rate : 0.6916
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.393
##
##  Mcnemar's Test P-Value : 9.818e-06
##
##             Sensitivity : 0.3333
##             Specificity : 0.1351
##          Pos Pred Value : 0.1467
##          Neg Pred Value : 0.3125
##              Prevalence : 0.3084
##          Detection Rate : 0.1028
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.2342
##
##        'Positive' Class : aprobado
##
```

Con esta estructura la red neuronal tampoco mejora. Con otras que se ha probado pero no se muestran tampoco mejoró.

```
precision_m3<-c(precision_m3, max(precisionNN_m))
names(precision_m3)[2]<-c("Redes Neuronales")
```

Método 3: Máquina de vector soporte Asignatura: portugués

Se ajusta, a continuación, el modelo para los datos de la asignatura de portugués con el kernel radial.

```
fitsvm11 <-svm(calificacion ~., data = Train3.notas_p)
summary(fitsvm11)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train3.notas_p)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  133
##
##  ( 73 60 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm11,Val3.notas_p)
matrizSVM11<-confusionMatrix(Val3.notas_p$calificacion, predictedSVM)
matrizSVM11
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       16        8
##
##                Accuracy : 0.9158
##                  95% CI : (0.8668, 0.9511)
##     No Information Rate : 0.9579
##     P-Value [Acc > NIR] : 0.9970053
##
##                   Kappa : 0.4663
##
##  Mcnemar's Test P-Value : 0.0001768
##
##             Sensitivity : 0.9121
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.3333
##              Prevalence : 0.9579
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.9560
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(matrizSVM11$overall[1])
names(precisionSVM_p)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm12 <-svm(calificacion ~., data = Train3.notas_p, kernel="polynomial")
summary(fitsvm12)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train3.notas_p, kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  142
##
##  ( 82 60 )
##
##
## Number of Classes:  2
##
```

```
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm12,Val3.notas_p)
matrizSVM12<-confusionMatrix(Val3.notas_p$calificacion, predictedSVM)
matrizSVM12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       24        0
##
##                Accuracy : 0.8737
##                  95% CI : (0.8179, 0.9174)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.668e-06
##
##             Sensitivity : 0.8737
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM12$overall[1])
names(precisionSVM_p)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm13 <-svm(calificacion ~., data = Train3.notas_p, kernel="sigmoid")
summary(fitsvm13)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train3.notas_p, kernel = "sigmoid")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  sigmoid
##        cost:  1
##      coef.0:  0
##
## Number of Support Vectors:  128
##
```

```
##  ( 68 60 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm13,Val3.notas_p)
matrizSVM13<-confusionMatrix(Val3.notas_p$calificacion, predictedSVM)
matrizSVM13
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      166        0
##   suspenso       20        4
##
##                Accuracy : 0.8947
##                  95% CI : (0.8421, 0.9345)
##     No Information Rate : 0.9789
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.259
##
##  Mcnemar's Test P-Value : 2.152e-05
##
##             Sensitivity : 0.8925
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.1667
##              Prevalence : 0.9789
##          Detection Rate : 0.8737
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.9462
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM13$overall[1])
names(precisionSVM_p)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm14 <-svm(calificacion ~., data = Train3.notas_p, kernel="linear")
summary(fitsvm14)
```

```
##
## Call:
## svm(formula = calificacion ~ ., data = Train3.notas_p, kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
```

```
##        cost:  1
##
## Number of Support Vectors:  75
##
##  ( 42 33 )
##
##
## Number of Classes:  2
##
## Levels:
##  aprobado suspenso
```

```
predictedSVM = predict(fitsvm14,Val3.notas_p)
matrizSVM14<-confusionMatrix(Val3.notas_p$calificacion, predictedSVM)
matrizSVM14
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      158        8
##   suspenso        8       16
##
##                Accuracy : 0.9158
##                  95% CI : (0.8668, 0.9511)
##     No Information Rate : 0.8737
##     P-Value [Acc > NIR] : 0.04519
##
##                   Kappa : 0.6185
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.9518
##             Specificity : 0.6667
##          Pos Pred Value : 0.9518
##          Neg Pred Value : 0.6667
##              Prevalence : 0.8737
##          Detection Rate : 0.8316
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.8092
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_p<-c(precisionSVM_p, matrizSVM14$overall[1])
names(precisionSVM_p)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenidos de los distintos kernel.

```
precisionSVM_p
```

```
##     radial polinomial  sigmoidal     lineal
##  0.9157895  0.8736842  0.8947368  0.9157895
```

La predicción de los SVM de kernel radial y lineal es la misma. La clasificación de estos kernels es la mejor de los cuatros. La peor es el SVM de kernel polinomial.

```
precision_p3<-c(precision_p3, max(precisionSVM_p))
names(precision_p3)[3]<-c("SVM")
```

Asignatura: matemáticas

Se prueba primero con el kernel radial.

```
fitsvm21 <-svm(calificacion ~., data = Train3.notas_m)
predictedSVM = predict(fitsvm21,Val3.notas_m)
matrizSVM21<-confusionMatrix(Val3.notas_m$calificacion, predictedSVM)
matrizSVM21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       74        1
##   suspenso       15       17
##
##                Accuracy : 0.8505
##                  95% CI : (0.7686, 0.912)
##     No Information Rate : 0.8318
##     P-Value [Acc > NIR] : 0.358429
##
##                   Kappa : 0.5922
##
##  Mcnemar's Test P-Value : 0.001154
##
##             Sensitivity : 0.8315
##             Specificity : 0.9444
##          Pos Pred Value : 0.9867
##          Neg Pred Value : 0.5313
##              Prevalence : 0.8318
##          Detection Rate : 0.6916
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8880
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(matrizSVM21$overall[1])
names(precisionSVM_m)<-c("radial")
```

Se prueba a continuación con el kernel polinomial.

```
fitsvm22 <-svm(calificacion ~., data = Train3.notas_m, kernel="polynomial")
predictedSVM = predict(fitsvm22,Val3.notas_m)
matrizSVM22<-confusionMatrix(Val3.notas_m$calificacion, predictedSVM)
matrizSVM22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       75        0
##   suspenso       32        0
##
```

```
##                Accuracy : 0.7009
##                  95% CI : (0.6048, 0.7856)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 0.7009
##             Specificity :     NA
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 1.0000
##          Detection Rate : 0.7009
##    Detection Prevalence : 0.7009
##       Balanced Accuracy :     NA
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM22$overall[1])
names(precisionSVM_m)[2]<-c("polinomial")
```

Ahora con el kernel sigmoidal.

```
fitsvm23 <-svm(calificacion ~., data = Train3.notas_m, kernel="sigmoid")
predictedSVM = predict(fitsvm23,Val3.notas_m)
matrizSVM23<-confusionMatrix(Val3.notas_m$calificacion, predictedSVM)
matrizSVM23
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       74        1
##   suspenso       15       17
##
##                Accuracy : 0.8505
##                  95% CI : (0.7686, 0.912)
##     No Information Rate : 0.8318
##     P-Value [Acc > NIR] : 0.358429
##
##                   Kappa : 0.5922
##
##  Mcnemar's Test P-Value : 0.001154
##
##             Sensitivity : 0.8315
##             Specificity : 0.9444
##          Pos Pred Value : 0.9867
##          Neg Pred Value : 0.5313
##              Prevalence : 0.8318
##          Detection Rate : 0.6916
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8880
##
```

```
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM23$overall[1])
names(precisionSVM_m)[3]<-c("sigmoidal")
```

Por último, con el kernel lineal.

```
fitsvm24 <-svm(calificacion ~., data = Train3.notas_m, kernel="linear")
predictedSVM = predict(fitsvm24,Val3.notas_m)
matrizSVM24<-confusionMatrix(Val3.notas_m$calificacion, predictedSVM)
matrizSVM24
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       69        6
##   suspenso        7       25
##
##                Accuracy : 0.8785
##                  95% CI : (0.8012, 0.9337)
##     No Information Rate : 0.7103
##     P-Value [Acc > NIR] : 2.932e-05
##
##                   Kappa : 0.7076
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9079
##             Specificity : 0.8065
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.7812
##              Prevalence : 0.7103
##          Detection Rate : 0.6449
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8572
##
##        'Positive' Class : aprobado
##
```

```
precisionSVM_m<-c(precisionSVM_m, matrizSVM24$overall[1])
names(precisionSVM_m)[4]<-c("lineal")
```

Se compara a continuación los porcentajes de clasificación correcta obtenidos de los distintos kernel.

```
precisionSVM_m
```

```
##     radial polinomial  sigmoidal     lineal
##  0.8504673  0.7009346  0.8504673  0.8785047
```

La mejor clasificación es la del kernel lineal, seguida por la del kernel radial y sigmoidal que es la misma, y por último el kernel polinomial.

```
precision_m3<-c(precision_m3, max(precisionSVM_m))
names(precision_m3)[3]<-c("SVM")
```

Método 4: Naive Bayes Asignatura: portugués

```
fitbayes1 <-naiveBayes(calificacion ~., data = Train3.notas_p)
predictedBayes= predict(fitbayes1,Val3.notas_p)
matrizNB1<-confusionMatrix(Val3.notas_p$calificacion, predictedBayes)
matrizNB1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      144       22
##   suspenso        5       19
##
##                Accuracy : 0.8579
##                  95% CI : (0.8, 0.9042)
##     No Information Rate : 0.7842
##     P-Value [Acc > NIR] : 0.006652
##
##                   Kappa : 0.5059
##
##  Mcnemar's Test P-Value : 0.002076
##
##             Sensitivity : 0.9664
##             Specificity : 0.4634
##          Pos Pred Value : 0.8675
##          Neg Pred Value : 0.7917
##              Prevalence : 0.7842
##          Detection Rate : 0.7579
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.7149
##
##        'Positive' Class : aprobado
##
```

```
precision_p3<-c(precision_p3, matrizNB1$overall[1])
names(precision_p3)[4]<-c("Naive Bayes")
```

Asignatura: matemáticas

```
fitbayes2 <-naiveBayes(calificacion ~., data = Train3.notas_m)
predictedBayes= predict(fitbayes2,Val3.notas_m)
matrizNB2<-confusionMatrix(Val3.notas_m$calificacion, predictedBayes)
matrizNB2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       69        6
##   suspenso        7       25
##
##                Accuracy : 0.8785
##                  95% CI : (0.8012, 0.9337)
##     No Information Rate : 0.7103
##     P-Value [Acc > NIR] : 2.932e-05
##
```

```
##                   Kappa : 0.7076
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9079
##             Specificity : 0.8065
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.7812
##              Prevalence : 0.7103
##          Detection Rate : 0.6449
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8572
##
##        'Positive' Class : aprobado
##
```

```
precision_m3<-c(precision_m3, matrizNB2$overall[1])
names(precision_m3)[4]<-c("Naive Bayes")
```

Método 5: Árboles de clasificación Asignatura: portugués

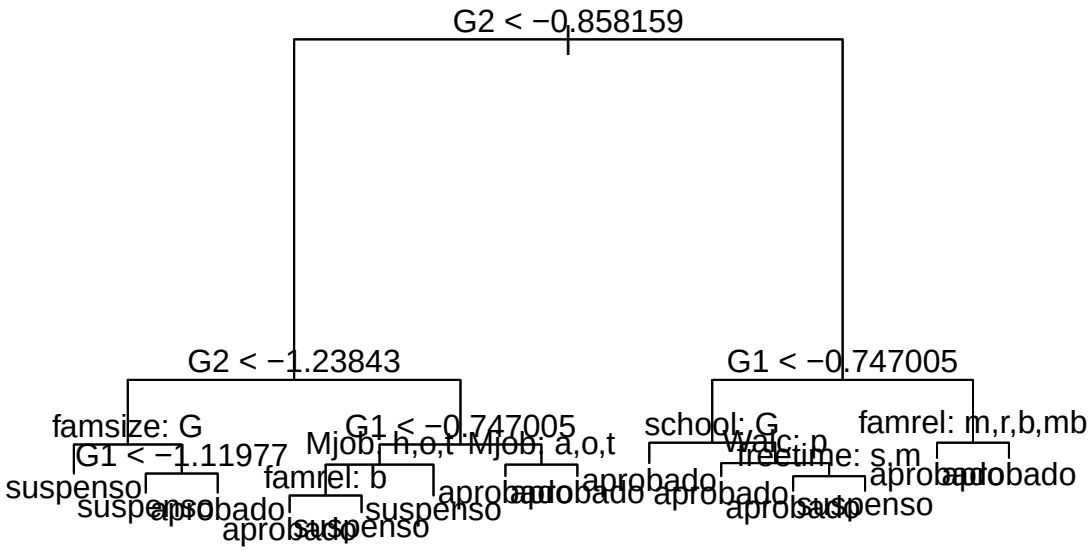
```
tree11 = tree(calificacion~., data = Train3.notas_p)
summary(tree11)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train3.notas_p)
## Variables actually used in tree construction:
## [1] "G2"       "famsize"  "G1"       "Mjob"     "famrel"   "school"   "Walc"
## [8] "freetime"
## Number of terminal nodes:  14
## Residual mean deviance:  0.1008 = 43.22 / 429
## Misclassification error rate: 0.02257 = 10 / 443
```

```
plot(tree11)
text(tree11, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII

## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

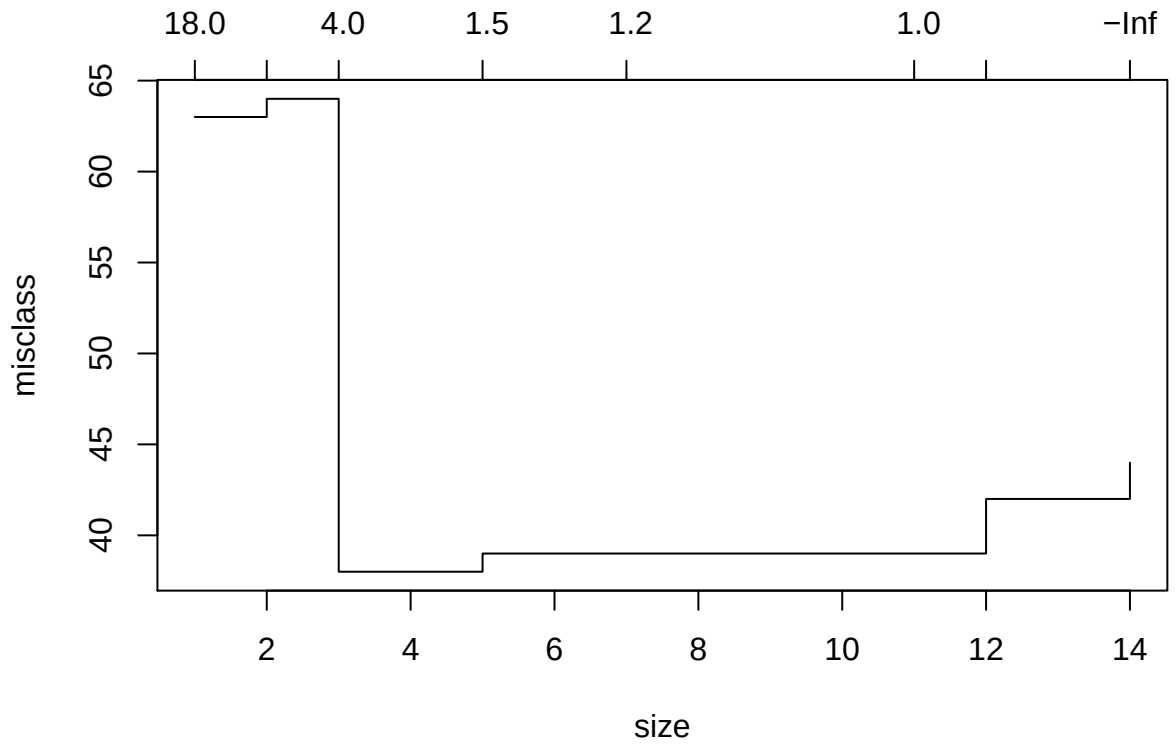
```
predicedtree = predict(tree11, Val3.notas_p, type="class")
matriztree11<-confusionMatrix(Val3.notas_p$calificacion, predicedtree)
matriztree11
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      157        9
##   suspenso        5       19
##
##                Accuracy : 0.9263
##                  95% CI : (0.8795, 0.9591)
##     No Information Rate : 0.8526
##     P-Value [Acc > NIR] : 0.00147
##
##                   Kappa : 0.6884
##
##  Mcnemar's Test P-Value : 0.42268
##
##             Sensitivity : 0.9691
##             Specificity : 0.6786
##          Pos Pred Value : 0.9458
##          Neg Pred Value : 0.7917
##              Prevalence : 0.8526
##          Detection Rate : 0.8263
```

```
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.8239
##
##        'Positive' Class : aprobado
##
```

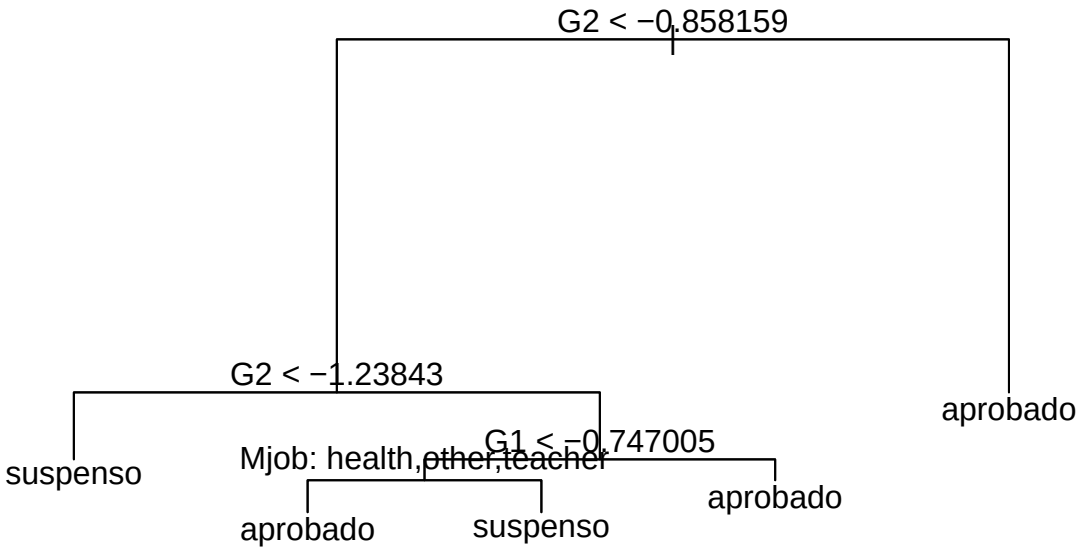
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree11 = cv.tree(tree11, FUN = prune.misclass)
plot(cv.tree11)
```



Se observa como o al tener muy pocas ramas o al aumentar el tamaño del árbol a más de cinco el error aumenta. Por ello, se elige que tenga 5 ramas.

```
prune.tree11 = prune.misclass(tree11, best = 5)
plot(prune.tree11)
text(prune.tree11, pretty=0)
```



Se observa que las ramas corresponden a las variables: G2, G1 y Mjob.

```
predicedtree12 = predict(prune.tree11, Val3.notas_p, type="class")
matriztree12<-confusionMatrix(Val3.notas_p$calificacion, predicedtree12)
matriztree12
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado      161        5
##   suspenso        8       16
##
##                Accuracy : 0.9316
##                  95% CI : (0.8858, 0.9631)
##     No Information Rate : 0.8895
##     P-Value [Acc > NIR] : 0.03512
##
##                   Kappa : 0.6725
##
##  Mcnemar's Test P-Value : 0.57910
##
##             Sensitivity : 0.9527
##             Specificity : 0.7619
##          Pos Pred Value : 0.9699
##          Neg Pred Value : 0.6667
##              Prevalence : 0.8895
```

```
##          Detection Rate : 0.8474
##    Detection Prevalence : 0.8737
##       Balanced Accuracy : 0.8573
##
##        'Positive' Class : aprobado
##
```

```
precision_p3<-c(precision_p3, matriztree12$overall[1])
names(precision_p3)[5]<-c("Arbol de clasificación")
```

Asignatura: matemáticas

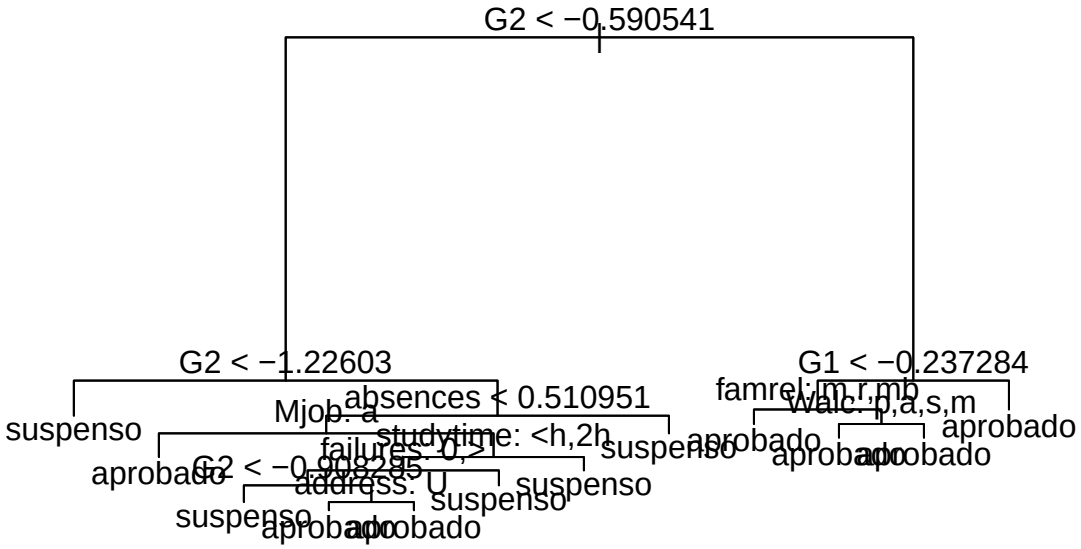
```
tree21 = tree(calificacion~., data = Train3.notas_m)
summary(tree21)
```

```
##
## Classification tree:
## tree(formula = calificacion ~ ., data = Train3.notas_m)
## Variables actually used in tree construction:
## [1] "G2"        "absences"  "Mjob"      "studytime" "failures"  "address"
## [7] "G1"        "famrel"    "Walc"
## Number of terminal nodes:  12
## Residual mean deviance:  0.1506 = 35.84 / 238
## Misclassification error rate: 0.036 = 9 / 250
```

```
plot(tree21)
text(tree21, pretty = 1)
```

```
## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII

## Warning in FUN(X[[i]], ...): abreviatura utilizada con caracteres no ASCII
```



Debido a la superposición de las etiquetas, el gráfico no es claro.

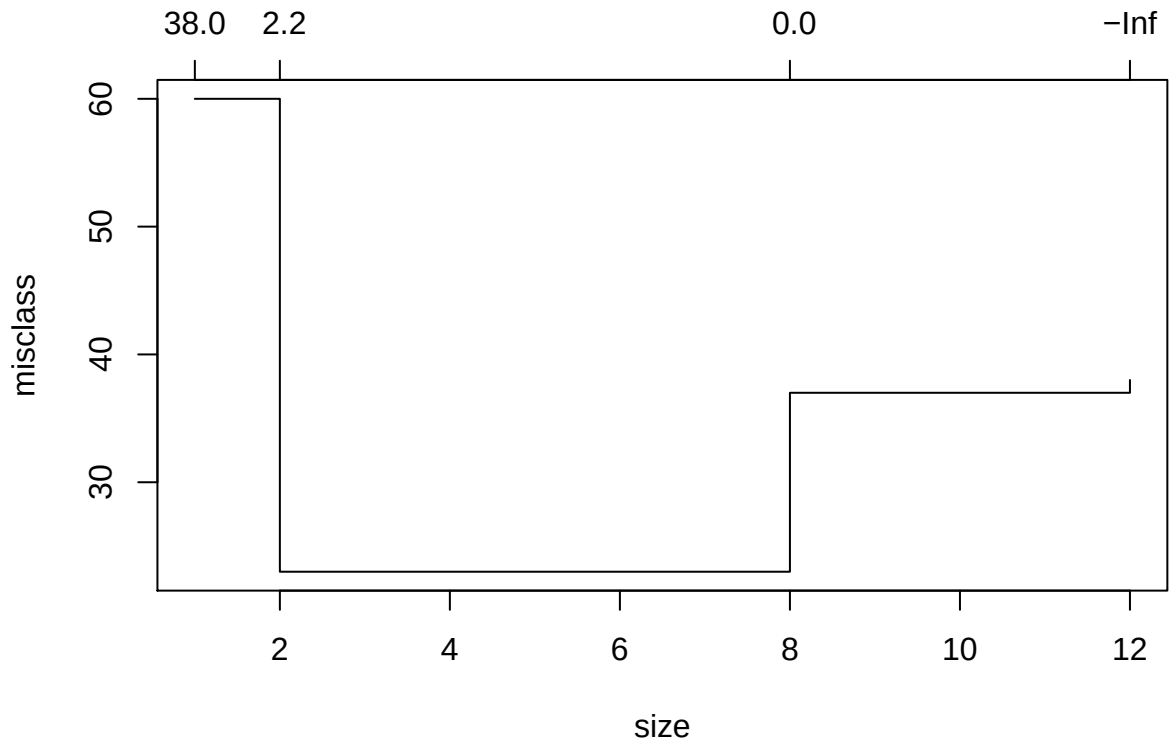
```
predicedtree = predict(tree21, Val3.notas_m, type="class")
matriztree21<-confusionMatrix(Val3.notas_m$calificacion, predicedtree)
matriztree21
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       73        2
##   suspenso       10       22
##
##                Accuracy : 0.8879
##                  95% CI : (0.8123, 0.9407)
##     No Information Rate : 0.7757
##     P-Value [Acc > NIR] : 0.002251
##
##                   Kappa : 0.7118
##
##  Mcnemar's Test P-Value : 0.043308
##
##             Sensitivity : 0.8795
##             Specificity : 0.9167
##          Pos Pred Value : 0.9733
##          Neg Pred Value : 0.6875
##              Prevalence : 0.7757
##          Detection Rate : 0.6822
```

```
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8981
##
##        'Positive' Class : aprobado
##
```

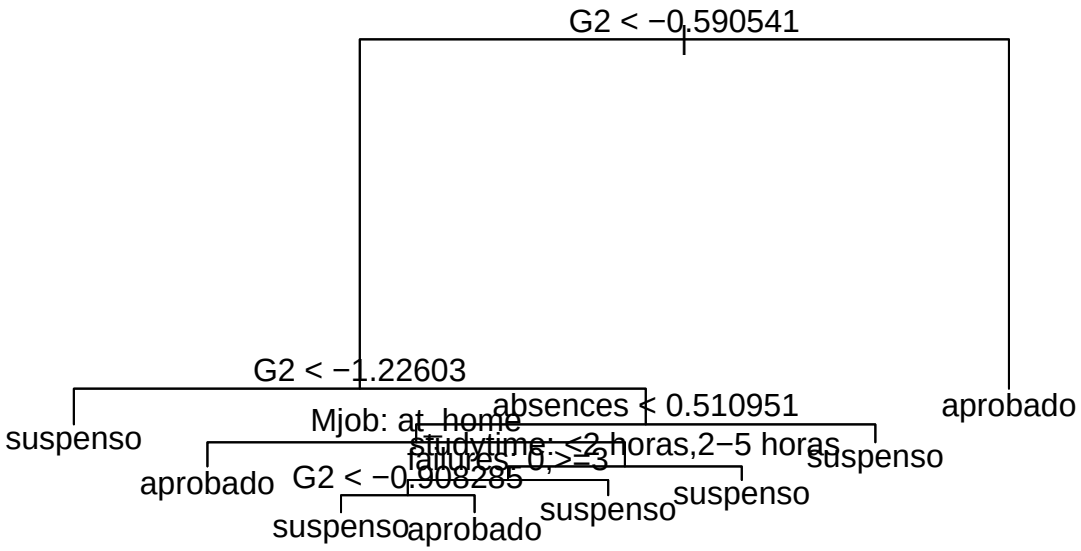
Se procede a podarlo para reducir su alta varianza al tener muchas ramas.

```
cv.tree21 = cv.tree(tree21, FUN = prune.misclass)
plot(cv.tree21)
```



Se observa como al tener muy pocas ramas, el porcentaje de error aumenta. Se elige que tenga 3 ramas que es de los números de ramas con menor errores y a partir del cual el error vuelve a crecer.

```
prune.tree21 = prune.misclass(tree21, best = 3)
plot(prune.tree21)
text(prune.tree21, pretty=0)
```



Se observa que las ramas corresponden a G2, Mjob, absences, studytime y failures.

```
predicedtree22 = predict(prune.tree21, Val3.notas_m, type="class")
matriztree22<-confusionMatrix(Val3.notas_m$calificacion, predicedtree22)
matriztree22
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction aprobado suspenso
##   aprobado       73        2
##   suspenso       10       22
##
##                Accuracy : 0.8879
##                  95% CI : (0.8123, 0.9407)
##     No Information Rate : 0.7757
##     P-Value [Acc > NIR] : 0.002251
##
##                   Kappa : 0.7118
##
##  Mcnemar's Test P-Value : 0.043308
##
##             Sensitivity : 0.8795
##             Specificity : 0.9167
##          Pos Pred Value : 0.9733
##          Neg Pred Value : 0.6875
##              Prevalence : 0.7757
```

```
##          Detection Rate : 0.6822
##    Detection Prevalence : 0.7009
##       Balanced Accuracy : 0.8981
##
##        'Positive' Class : aprobado
##
```

```
precision_m3<-c(precision_m3, matriztree22$overall[1])
names(precision_m3)[5]<-c("Arbol de clasificación")
```

```
precision_p<-rbind(precision_p1, precision_p2, precision_p3)
rownames(precision_p)<-c("Sin G1 y G2", "Con G1 y sin G2", "Con G1 y G2")
precision_m<-rbind(precision_m1, precision_m2, precision_m3)
rownames(precision_m)<-c("Sin G1 y G2", "Con G1 y sin G2", "Con G1 y G2")
```

Discusión

El rendimiento académico de los estudiantes se mide y se cuantifica mediante las notas. Estas notas son de alta importancia en los últimos cursos previos a la universidad ya que pueden restringir la futura educación del estudiante, como por ejemplo las carreraras universitarias o instituciones en las que pueda estudiar. Por ello, es de suma importancia el poder predecir las notas de los estudiantes para en el caso de mal rendimiento proporcionarles la ayuda necesaria antes del examen final.

En la predicción númerica de la nota final mediante regresión múltiple se observa como al no incluir en el análisis las notas de los trimestres previos las predicción explica muy poco porcentaje de la varianza, es decir, no es altamente fiable. Sin embargo en este caso puede servir de ayuda premiliminar antes de conocer la nota del primer trimestre y posteriormente confirmar si necesita ayuda con un mayor porcentaje de fiabilidad.

```
R_Cuadrado
```

```
##             Sin G1 y G2 Con G1 y sin G2 Con G1 y G2
## Portugués        0.4429          0.8118      0.9116
## Matemáticas      0.4089          0.8492      0.9485
```

En este escenario en el que no se tiene ninguna nota previa, en la asignatura de portugués, se observa como el colegio Mousinho da Silveira, el género masculino, los suspensos, el apoyo del colegio, la salud regular o muy bien y las ausencias son factores significativos que influyen de forma negativa en la nota final, especialmente los suspensos. Sin embargo, la edad, el tiempo de estudio superior a diez horas, el querer continuar con su educación, una buena o muy buena relación de familia, salir poco y tener poco tiempo libre repercuten de manera positiva, especialmente el querer continuar con su educación. En la asigntura de matemáticas son menos las variables significativas pero sus coeficientes repercuten en la nota final de manera similar que en la asignatura de portugués.

Que un estudiante que tenga suspensos previos repercuta de manera negativa en la nota final tiene sentido ya que tiene tendencias previas de suspender. El apoyo del colegio puede mostrar las capacidades de un estudiante, es decir, que si necesita apoyo en la asigntura puede ser que vaya peor y por ello su predicción de nota final sea peor. El que tener salud regular o muy bien repercuta de manera negativa puede deberse a que al tener mejor salud prefieren no preocuparse por sus estudios y centrarse en otras cosas como en salir, jugar, ... Las variables que afectan de forma positiva se ve su relación directamente a excepción de la de tener poco tiempo libre, que considero que puede ser por dedicarle bastante tiempo a los estudios, o la de la edad que puede ser que al repetir la asignatura ya que tenga conocimientos del año anterior y le resulte más fácil.

Al ya incluir la nota del primer trimestre, en la predicción númerica de la nota final el coeficiente de determinación se dispara drásticamente hacia arriba, pasando a ser alrededos del 80%. Esto se debe a la alta correlación entre G1 y G3. En este escenario pasa a ser especialmente significativa la edad, tienendo un efecto negativo. Además se vuelven significativas afectando también negativamente a G3 que los padres de los estudiantes trabajen en trabajos que se han denominado "otros" o "servicios" en la asignatura de portugués y

en la de matemáticas el consumo de un poco de alcohol diario. En este análisis existe colinealidad por lo que tampoco es altamente fiable y no se profundizará su análisis del trabajo del padre o cosumno de alcohol.

El escenario 3, es decir, contando con G1 y G2, es totalmente similar al segundo escenario añadiendole unas decimas al coeficiente de determinación por la nueva variable introducida.

En cuanto al análisis binario de la nota final, clasificando a los alumnos con aprobado o suspenso, los porcentajes de clasificación correcta son los siguientes:

Para la asignatura de portugués:

```
precision_p
```

```
##                 Regresion Logistica Redes Neuronales       SVM Naive Bayes
## Sin G1 y G2               0.8210526       0.21052632 0.8736842   0.8473684
## Con G1 y sin G2           0.8842105       0.12631579 0.8894737   0.8789474
## Con G1 y G2               0.8894737       0.08421053 0.9157895   0.8578947
##                 Arbol de clasificación
## Sin G1 y G2                  0.8789474
## Con G1 y sin G2              0.9263158
## Con G1 y G2                  0.9315789
```

Para la asignatura de matemáticas:

```
precision_m
```

```
##                 Regresion Logistica Redes Neuronales       SVM Naive Bayes
## Sin G1 y G2               0.7102804        0.3457944 0.7570093   0.7196262
## Con G1 y sin G2           0.7476636        0.2616822 0.8037383   0.7850467
## Con G1 y G2               0.8317757        0.1869159 0.8785047   0.8785047
##                 Arbol de clasificación
## Sin G1 y G2                  0.7383178
## Con G1 y sin G2              0.8411215
## Con G1 y G2                  0.8878505
```

Para el escenario 1, sin contar con G1 y G2, el mejor método de predicción en la asignatura de portugués es el árbol de clasificación y en la asignatura de matemáticas es el SVM. En la asignatura de portugués el SVM también es especialmente bueno siendo el segundo mejor.

Para el escenario 2 y el escenario 3, el mejor método de predicción en ambas asignaturas es el árbol de clasificación.

En general, el mejor método de predicción para este estudio son los árboles de clasificación. El peor método de clasificación son las redes neuronales. Esto se puede deber a no haber encontrado una correcta función de activicación.

Se puede concluir que las notas de los alumnos pueden estar influidas por ya no solo la propia inteligencia del alumno, si no por factores del colegio, sociales y demográficos; pero que el mayor factor significativo en la nota final, son las notas previas. Es decir, la nota final está altamente influenciada por la notas previas.

Bibliografía

Diapositivas de Studium.

P. Cortez y A. Silva. Using Data Mining to Predict Secondary School Student Performance. En A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data mining: Practical machine learning tools and techniques. Burlington, MA: Morgan Kaufmann.

Ye, N., 2014. Data mining: theories, algorithms, and examples. Boca Raton: Taylor & Francis.

Agresti, A. (1990). Categorical data analysis. New York [u.a.]: Wiley

El modelo de redes neuronales. (s. f.). www.ibm.com. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model

L Breiman, JH Friedman, RA Olshen, and CJ Stone.Classification and Regression Trees. Wadsworth Inc, 1984

Álvarez Rodríguez, R. (2020). Predicción del rendimiento académico en las Matemáticas de la educación Secundaria mediante Redes Neuronales. Universidad Nacional de Educación a Distancia. http://e-spacio.uned.es/fez/eserv/bibliuned:master-Ciencias-FSC-Ralvarez/Alvarez_Rodriguez_Roi_TFM.pdf

Regresión Lineal. En *Wikipedia*. https://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal#Regresi%C3%B3n_lineal_simple

Regresión logística En *Wikipedia*. https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica

Máxima verosimilitud. En *Wikipedia*. https://es.wikipedia.org/wiki/M%C3%A1xima_verosimilitud

Máquinas de vectores de soporte. En *Wikipedia*. https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte