Tipología y ciclo de vida de los datos

Práctica 2: Limpieza y validación de los datos

Autores: César Fernández Domínguez, Isabel Fernández Esparza

Junio 2019

# Solución

## Descripción del dataset

Para la realización de esta práctica se ha seleccionado un conjunto de datos relacionado con resultados académicos de estudiantes de dos colegios de Portugal disponible en el repositorio de datos *kaggle*. En concreto este conjunto de datos se ha obtenido del enlace: <https://www.kaggle.com/uciml/student-alcohol-consumption>

Este dataset contiene información de estudiantes de matemáticas en edad de estudios secundarios. Se puede utilizar para analizar cómo afectan a los estudiantes de secundaria sus circunstancias personales a la hora de tener voluntad de continuar con estudios de mayor nivel. Estas circunstancias personales podemos entenderlas desde el punto de vista de: nivel de estudios de los padres, trabajo de los padres, si tienen pareja o no… Por otro lado, también se podría hacer un análisis para estudiar la relación entre el número de suspensos de los estudiantes y el nivel de estudio de los padres, la distancia de los alumnos a los colegios, cómo influye el hecho de disponer de ayuda extraescolar en los resultados escolares, el tiempo de estudio semanal, el consumo de alcohol tanto diario como semanal, el número de ausencias…

Este conjunto de datos se presenta en dos ficheros distintos, en formato CSV: student-mad.csv (asignatura de matemáticas) y student-por.csv (asignatura de portugues), uno por cada asignatura.

El objetivo de esta práctica es limpiar los datos, unificarlos y poder estimar un modelo que pueda predecir el número de suspensos de un estudiante de matemáticas atendiendo a los factores anteriormente descritos. Teniendo en cuenta que en el juego de datos tenemos información de dos colegios diferentes podemos también intentar analizar si las predicciones realizadas están también sesgadas por el colegio al que pertenezcan los alumnos o por el sexo.

A continuación, se presenta una descripción de los atributos, para cada estudiante, contenidos en los dos ficheros:

1. school - colegio al que pertenece el alumno (binario: ‘GP’ - Gabriel Pereira o ‘MS’ - Mousinho da Silveira)
2. sex - sexo (binario: ‘F’ - mujer o ‘M’ - hombre)
3. age - edad (numérico: entre 15 y 22 años)
4. address - tipo de residencia (binario: ‘U’ - urbana o ‘R’ - rural)
5. famsize - tamaño de la familia (binario: ‘LE3’ - menor o igual a 3 o ‘GT3’ - mayor que 3)
6. Pstatus - padres separados o no (binario: ‘T’ - viven juntos o ‘A’ - separados)
7. Medu - nivel educativo de la madre (numérico: 0 - ninguno, 1 - educación primaria (4º grado), 2 – entre 5º a 9º grado, 3 – educación secundaria o 4 – educación superior)
8. Fedu - nivel educativo del padre (numérico: 0 - ninguno, 1 - educación primaria (4º grado), 2 – entre 5º a 9º grado, 3 – educación secundaria o 4 – educación superior)
9. Mjob - trabajo de la madre (nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (p.e. administrativa o policía), ‘at\_home’ o ‘other’)
10. Fjob - trabajo del padre (nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (p.e. administrativa o policía), ‘at\_home’ o ‘other’)
11. reason - razón para elegir la escuela (nominal: cerca de ‘home’, school ‘reputation’, ‘course’ preferencia o ‘other’)
12. guardian - guardian del estudiante (nominal: ‘mother’, ‘father’ o ‘other’)
13. traveltime - tiempo de viaje desde casa a la escuela (numérico: 1 - <15 min., 2 - 15 a 30 min., 3 - 30 min. a 1 hour, o 4 - >1 hora)
14. studytime - tiempo de estudio semanal (numérico: 1 - <2 horas, 2 - 2 a 5 horas, 3 - 5 a 10 horas, o 4 - >10 horas)
15. failures - número de asignaturas suspensas (numérico: n si 1<=n<3, en cualquier otro caso 4)
16. schoolsup - apoyo educativo adicional (binario: yes o no)
17. famsup - ayuda educativa de la familia (binario: yes o no)
18. paid - clases privadas de las asignaturas (Matemáticas o Portugues) (binario: yes o no)
19. activities - actividades extra-escolares (binario: yes o no)
20. nursery - asistió a la guardería (binario: yes o no)
21. higher - el alumno quiere realizar estudios superiores (binario: yes o no)
22. internet - el alumno tiene Internet en casa (binario: yes o no)
23. romantic - el alumno tiene pareja o no (binario: yes o no)
24. famrel - calidad de la relación familiar (numérico: desde 1 - muy mal a 5 - excelente)
25. freetime - tiempo libre después de la escuela (numérico: desde 1 - muy poco a 5 - mucho)
26. goout - el alumno sale con amigo (numérico: desde 1 - muy poco a 5 - mucho)
27. Dalc - consumo de alcohol diario (numérico: desde 1 - muy poco a 5 - mucho)
28. Walc - consumo de alcohol durante el fin de semana (numérico: desde 1 - muy poco a 5 - mucho)
29. health - estado de salud del alumno (numérico: desde 1 - muy mal a 5 - muy bueno)
30. absences - número de ausencias del alumno (numérico: desde 0 a 93)

Además de los siguientes calificaciones relacionadas con las asignaturas de matemáticas y portugues:

1. G1 - calificación primer trimestre (numérico: entre 0 a 20)
2. G2 - calificación segundo trimestre (numérico: entre 0 a 20)
3. G3 - calificación tercer trimestre (numérico: entre 0 a 20)

## Importancia y objetivos de los análisis

Se plantea la necesidad de responder a las siguientes preguntas:

* ¿ Cuales serán las variables que influyen más en la calificación de los estudiantes ?
* ¿ Predecir cuales serán la calificaciones de un estudiante en función de los otros atributos ?
* ¿ Hay alguna propiedad que pueda identificar a parte de la población ?
* Los alumnos que dedican más tiempo al estudio sacan mejores notas.
* Aquellos alumnos que van a clases particulares o reciben ayuda por parte de sus padres sacan mejores notas.
* En general, las chicas son mejores estudiantes que los chicos.

Estos análisis resultan de vital importancia tanto para el profesorado y dirección de una escuela, como para los padres de estudiantes.

## Integración y selección de los datos de interés a analizar.

sMat=read.table("data/student-mat.csv",sep=",",header=TRUE)  
sPor=read.table("data/student-por.csv",sep=",",header=TRUE)  
  
# Según el propietario de los datos, los alumnos que están presentes en ambas asignaturas   
# pueden ser identificados por los siguientes atributos  
sBoth=merge(sMat,sPor,by=c("school","sex","age","address","famsize","Pstatus","Medu","Fedu","Mjob","Fjob","reason","nursery","internet"))  
  
# school, sex, age, address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, nursery, internet  
# traveltime, studytime, failures, schoolsup, famsup, paid, activities, higher, romantic, famrel  
# freetime, goout, Dalc, Walc, health, absences, subject

Ambos ficheros de datos de estudiantes contienen 33 atributos (columnas). El fichero de estudiantes de la asignatura de portugues contiene 649 estudiantes y, el de la asignatura de matemáticas 395 estudiantes. Si mezclamos ambos ficheros para obtener los alumnos que están en ambas asignaturas obtenemos un total de 382 estudiantes.

Identificamos cada estudiante mediante los atributos indicados por el propietario del juego de datos. Generamos un identificador con la concatenación de estos atributos para cada estudiante. Después, en otro paso, convertiremos este identificador en un valor numérico que identifique a cada estudiante.

sMat$id = paste(sMat$school,sMat$sex,sMat$age,sMat$address,sMat$famsize,sMat$Pstatus,sMat$Medu,sMat$Fedu,sMat$Mjob,sMat$Fjob,sMat$reason,sMat$nursery,sMat$internet, sep="-")  
sPor$id = paste(sPor$school,sPor$sex,sPor$age,sPor$address,sPor$famsize,sPor$Pstatus,sPor$Medu,sPor$Fedu,sPor$Mjob,sPor$Fjob,sPor$reason,sPor$nursery,sPor$internet, sep="-")

Creamos una variable “score” que contendrá la media de las tres notas de los tres trimestres por cada alumno y asignatura. Luego, a partir de esta variable, creamos una variable categórica que exprese si un alumno ha aprobado o suspendido la asignatura.

sMat$score = rowMeans(subset(sMat, select = c(G1, G2, G3)), na.rm = TRUE)  
sMat$mark<-sMat$score  
sMat$mark[sMat$score<10] <- "fail"  
sMat$mark[sMat$score>=10] <- "pass"  
sMat$mark <- as.factor(sMat$mark)  
  
sPor$score = rowMeans(subset(sPor, select = c(G1, G2, G3)), na.rm = TRUE)  
sPor$mark<-sPor$score  
sPor$mark[sPor$score<10] <- "fail"  
sPor$mark[sPor$score>=10] <- "pass"  
sPor$mark <- as.factor(sPor$mark)

Vamos a crear un par de variables binarias nuevas para después unir los dos ficheros en un solo dataset:

sMat$math = 'yes'  
sMat$port = 'no'  
sPor$math = 'no'  
sPor$port = 'yes'  
students = rbind(sMat,sPor)  
students$port = as.factor(students$port)  
students$math = as.factor(students$math)

Ahora, a partir del identificador que anteriormente habiamos creado para cada estudiante, lo transformamos en un identificador numérico simple.

students = transform(students, id=as.numeric(factor(id)))  
students$id = as.factor(students$id)

Ahora tenemos un dataset con 1044 instancias y 38 atributos para un total de 662 estudiantes de ambas asignaturas.

# Resumen  
glimpse(students)

## Observations: 1,044  
## Variables: 38  
## $ school <fct> GP, GP, GP, GP, GP, GP, GP, GP, GP, GP, GP, GP, GP,...  
## $ sex <fct> F, F, F, F, F, M, M, F, M, M, F, F, M, M, M, F, F, ...  
## $ age <int> 18, 17, 15, 15, 16, 16, 16, 17, 15, 15, 15, 15, 15,...  
## $ address <fct> U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, ...  
## $ famsize <fct> GT3, GT3, LE3, GT3, GT3, LE3, LE3, GT3, LE3, GT3, G...  
## $ Pstatus <fct> A, T, T, T, T, T, T, A, A, T, T, T, T, T, A, T, T, ...  
## $ Medu <int> 4, 1, 1, 4, 3, 4, 2, 4, 3, 3, 4, 2, 4, 4, 2, 4, 4, ...  
## $ Fedu <int> 4, 1, 1, 2, 3, 3, 2, 4, 2, 4, 4, 1, 4, 3, 2, 4, 4, ...  
## $ Mjob <fct> at\_home, at\_home, at\_home, health, other, services,...  
## $ Fjob <fct> teacher, other, other, services, other, other, othe...  
## $ reason <fct> course, course, other, home, home, reputation, home...  
## $ guardian <fct> mother, father, mother, mother, father, mother, mot...  
## $ traveltime <int> 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 3, 1, 2, 1, 1, 1, ...  
## $ studytime <int> 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 1, 2, 3, 1, 3, ...  
## $ failures <int> 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...  
## $ schoolsup <fct> yes, no, yes, no, no, no, no, yes, no, no, no, no, ...  
## $ famsup <fct> no, yes, no, yes, yes, yes, no, yes, yes, yes, yes,...  
## $ paid <fct> no, no, yes, yes, yes, yes, no, no, yes, yes, yes, ...  
## $ activities <fct> no, no, no, yes, no, yes, no, no, no, yes, no, yes,...  
## $ nursery <fct> yes, no, yes, yes, yes, yes, yes, yes, yes, yes, ye...  
## $ higher <fct> yes, yes, yes, yes, yes, yes, yes, yes, yes, yes, y...  
## $ internet <fct> no, yes, yes, yes, no, yes, yes, no, yes, yes, yes,...  
## $ romantic <fct> no, no, no, yes, no, no, no, no, no, no, no, no, no...  
## $ famrel <int> 4, 5, 4, 3, 4, 5, 4, 4, 4, 5, 3, 5, 4, 5, 4, 4, 3, ...  
## $ freetime <int> 3, 3, 3, 2, 3, 4, 4, 1, 2, 5, 3, 2, 3, 4, 5, 4, 2, ...  
## $ goout <int> 4, 3, 2, 2, 2, 2, 4, 4, 2, 1, 3, 2, 3, 3, 2, 4, 3, ...  
## $ Dalc <int> 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...  
## $ Walc <int> 1, 1, 3, 1, 2, 2, 1, 1, 1, 1, 2, 1, 3, 2, 1, 2, 2, ...  
## $ health <int> 3, 3, 3, 5, 5, 5, 3, 1, 1, 5, 2, 4, 5, 3, 3, 2, 2, ...  
## $ absences <int> 6, 4, 10, 2, 4, 10, 0, 6, 0, 0, 0, 4, 2, 2, 0, 4, 6...  
## $ G1 <int> 5, 5, 7, 15, 6, 15, 12, 6, 16, 14, 10, 10, 14, 10, ...  
## $ G2 <int> 6, 5, 8, 14, 10, 15, 12, 5, 18, 15, 8, 12, 14, 10, ...  
## $ G3 <int> 6, 6, 10, 15, 10, 15, 11, 6, 19, 15, 9, 12, 14, 11,...  
## $ id <fct> 184, 123, 39, 25, 72, 341, 336, 121, 279, 260, 32, ...  
## $ score <dbl> 5.666667, 5.333333, 8.333333, 14.666667, 8.666667, ...  
## $ mark <fct> fail, fail, fail, pass, fail, pass, pass, fail, pas...  
## $ math <fct> yes, yes, yes, yes, yes, yes, yes, yes, yes, yes, y...  
## $ port <fct> no, no, no, no, no, no, no, no, no, no, no, no, no,...

# Estadísticas básicas  
summary(students)

## school sex age address famsize Pstatus  
## GP:772 F:591 Min. :15.00 R:285 GT3:738 A:121   
## MS:272 M:453 1st Qu.:16.00 U:759 LE3:306 T:923   
## Median :17.00   
## Mean :16.73   
## 3rd Qu.:18.00   
## Max. :22.00   
##   
## Medu Fedu Mjob Fjob   
## Min. :0.000 Min. :0.000 at\_home :194 at\_home : 62   
## 1st Qu.:2.000 1st Qu.:1.000 health : 82 health : 41   
## Median :3.000 Median :2.000 other :399 other :584   
## Mean :2.603 Mean :2.388 services:239 services:292   
## 3rd Qu.:4.000 3rd Qu.:3.000 teacher :130 teacher : 65   
## Max. :4.000 Max. :4.000   
##   
## reason guardian traveltime studytime   
## course :430 father:243 Min. :1.000 Min. :1.00   
## home :258 mother:728 1st Qu.:1.000 1st Qu.:1.00   
## other :108 other : 73 Median :1.000 Median :2.00   
## reputation:248 Mean :1.523 Mean :1.97   
## 3rd Qu.:2.000 3rd Qu.:2.00   
## Max. :4.000 Max. :4.00   
##   
## failures schoolsup famsup paid activities nursery   
## Min. :0.0000 no :925 no :404 no :824 no :528 no :209   
## 1st Qu.:0.0000 yes:119 yes:640 yes:220 yes:516 yes:835   
## Median :0.0000   
## Mean :0.2644   
## 3rd Qu.:0.0000   
## Max. :3.0000   
##   
## higher internet romantic famrel freetime   
## no : 89 no :217 no :673 Min. :1.000 Min. :1.000   
## yes:955 yes:827 yes:371 1st Qu.:4.000 1st Qu.:3.000   
## Median :4.000 Median :3.000   
## Mean :3.936 Mean :3.201   
## 3rd Qu.:5.000 3rd Qu.:4.000   
## Max. :5.000 Max. :5.000   
##   
## goout Dalc Walc health   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:3.000   
## Median :3.000 Median :1.000 Median :2.000 Median :4.000   
## Mean :3.156 Mean :1.494 Mean :2.284 Mean :3.543   
## 3rd Qu.:4.000 3rd Qu.:2.000 3rd Qu.:3.000 3rd Qu.:5.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
##   
## absences G1 G2 G3   
## Min. : 0.000 Min. : 0.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 0.000 1st Qu.: 9.00 1st Qu.: 9.00 1st Qu.:10.00   
## Median : 2.000 Median :11.00 Median :11.00 Median :11.00   
## Mean : 4.435 Mean :11.21 Mean :11.25 Mean :11.34   
## 3rd Qu.: 6.000 3rd Qu.:13.00 3rd Qu.:13.00 3rd Qu.:14.00   
## Max. :75.000 Max. :19.00 Max. :19.00 Max. :20.00   
##   
## id score mark math port   
## 317 : 4 Min. : 1.333 fail:321 no :649 no :395   
## 326 : 4 1st Qu.: 9.333 pass:723 yes:395 yes:649   
## 360 : 4 Median :11.333   
## 408 : 4 Mean :11.267   
## 158 : 3 3rd Qu.:13.333   
## 185 : 3 Max. :19.333   
## (Other):1022

# Tipo de dato asignado a cada campo  
sapply(students, class)

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

## Limpieza de los datos

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Identificación y tratamiento de valores extremos.

# Con datos nulos  
colSums(is.na(students))

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

# Con datos ""  
colSums(students=="")

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

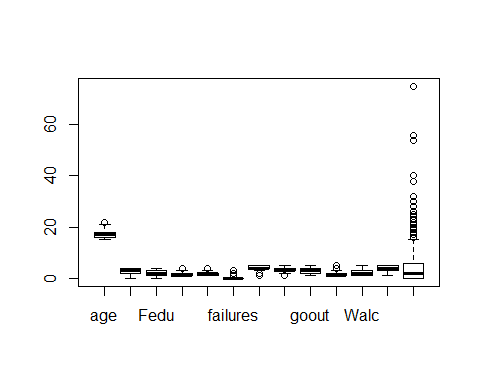
# Con datos "?"  
colSums(students=="?")

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

# Valores unicos  
apply(students,2, function(x) length(unique(x)))

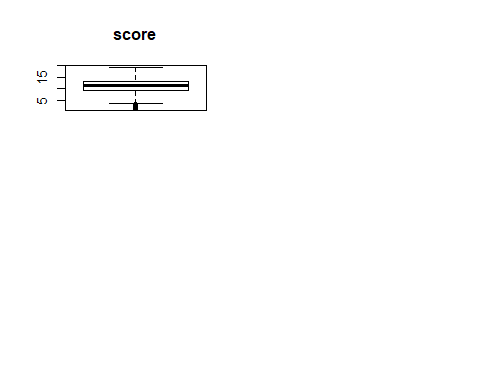
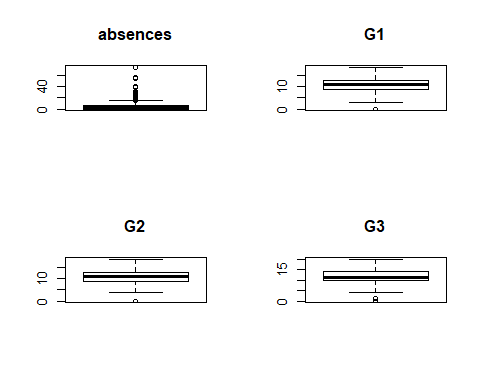
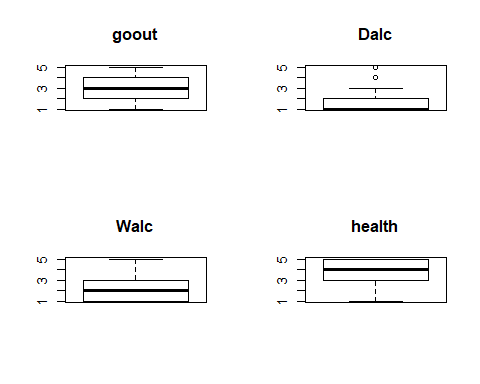
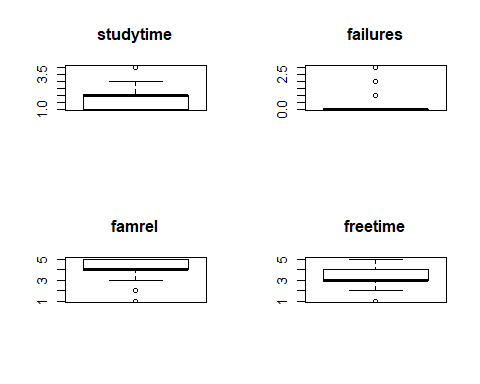
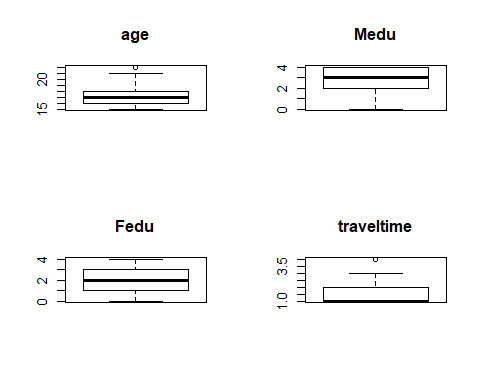
## school sex age address famsize Pstatus   
## 2 2 8 2 2 2   
## Medu Fedu Mjob Fjob reason guardian   
## 5 5 5 5 4 3   
## traveltime studytime failures schoolsup famsup paid   
## 4 4 4 2 2 2   
## activities nursery higher internet romantic famrel   
## 2 2 2 2 2 5   
## freetime goout Dalc Walc health absences   
## 5 5 5 5 5 35   
## G1 G2 G3 id score mark   
## 18 17 19 662 54 2   
## math port   
## 2 2

boxplot(students[, c("age", "Medu", "Fedu", "traveltime", "studytime", "failures", "famrel", "freetime", "goout", "Dalc", "Walc", "health", "absences")])



# Eliminamos outliers  
  
for(i in 1:ncol(students)) {  
 if (is.numeric(students[,i])){  
 outliers = boxplot.stats(students[,i])$out  
 index = which( students[,i] %in% outliers)  
 if(length(index)>0) {  
 #students=students[-index,]  
 }  
 }  
}

par(mfrow=c(2,2))  
for(i in 1:ncol(students)) {  
 if (is.numeric(students[,i])){  
 boxplot(students[,i], main = colnames(students)[i], width = 100)  
 }  
}



## Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

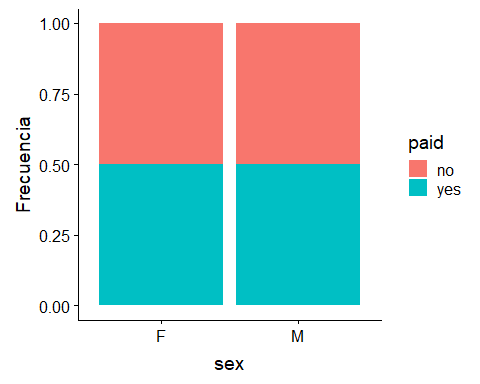
Comprobación de la normalidad y homogeneidad de la varianza.

A continuación, se seleccionan los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar. No obstante, como se verá en el apartado consistente en la realización de pruebas estadísticas, no todos se utilizarán.

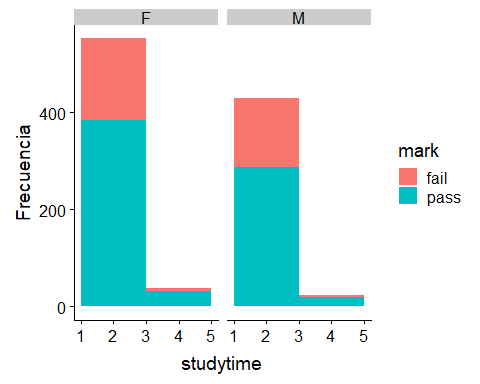
# school, sex, age, address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, nursery, internet  
# traveltime, studytime, failures, schoolsup, famsup, paid, activities, higher, romantic, famrel  
# freetime, goout, Dalc, Walc, health, absences, subject  
  
# Agrupación por sexo de los estudiantes  
students.male <- students[students$sex == "M",]  
students.female <- students[students$sex == "F",]  
  
# Agrupación por si reciben clases particules pagadas o no  
students.paid <- students[students$paid == "yes",]  
students.nopaid <- students[students$paid == "no",]  
  
# Agrupación por si reciben soporte por parte de la familia  
students.famsup <- students[students$famsup == "yes",]  
students.nofamsup <- students[students$famsup == "no",]  
  
# Agrupación por si reciben ayuda extra escolar  
students.schoolsup <- students[students$schoolsup == "yes",]  
students.noschoolsup <- students[students$schoolsup == "no",]  
  
# Agrupación por tiempo dedicado al estudio  
students.studytime <- students[students$studytime >= 3,]  
students.nostudytime <- students[students$studytime < 3,]  
  
# Agrupación por edad  
students.mayores <- students[students$age >= 16,]  
students.menores <- students[students$age < 16,]  
  
# Agrupación por asignatura  
students.port <- students[students$port == "yes",]  
students.math <- students[students$math == "yes",]

### Análisis visual

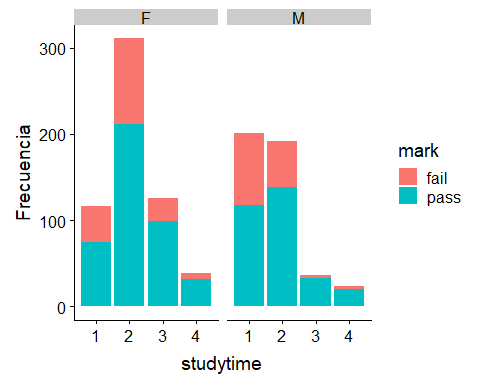
ggplot(data = students[1:length(students),],aes(x=sex,fill=paid))+geom\_bar(position="fill")+ylab("Frecuencia")



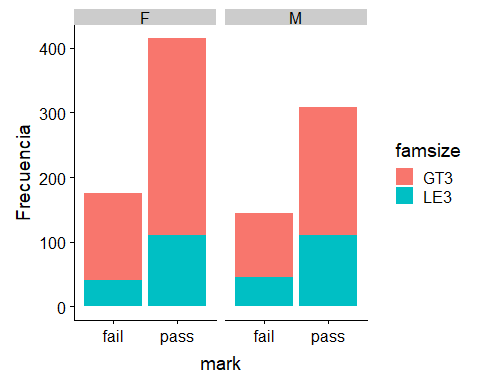
ggplot(students,aes(x=studytime,fill=mark))+geom\_histogram(binwidth = 2)+ylab("Frecuencia")+facet\_wrap(~sex )



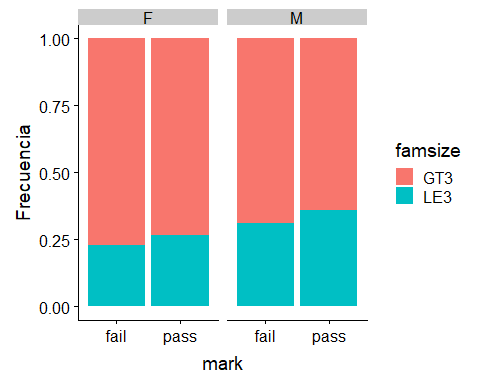
ggplot(students,aes(x=studytime,fill=mark))+geom\_bar()+ylab("Frecuencia")+facet\_wrap(~sex )



ggplot(students,aes(x=mark,fill=famsize))+geom\_bar()+ylab("Frecuencia")+facet\_wrap(~sex )



ggplot(students,aes(x=mark,fill=famsize))+geom\_bar(position='fill')+ylab("Frecuencia")+facet\_wrap(~sex )

 ##Comprobación de la normalidad y homogeneidad de la varianza Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de Anderson-Darling. Así, se comprueba que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado α = 0,05. Si esto se cumple, entonces se considera que la variable en cuestión sigue una distribución normal.

# Tests de normalidad   
library(nortest)  
  
# Función que aplica distintos test de homogeneidad sobre los datos de entrada  
normTest <- function(data, name) {  
 ad\_val = (ad.test(data)$p.value > alpha) # Anderson-Darling test  
 ks\_val = (ks.test(data, pnorm, mean(data), sd(data))$p.value > alpha) # Kolmogorov-Smirnov test  
 sh\_val = (shapiro.test(data)$p.value > alpha) # Shapiro test  
 cat(name)  
 cat("\t")  
 cat(ad\_val,ks\_val,sh\_val,"\t")  
 cat("\n")  
}  
  
alpha = 0.05  
col.names = colnames(students)  
cat("Distribucion normal: \n")

## Distribucion normal:

for (i in 1:ncol(students)) {  
 if (is.integer(students[,i]) | is.numeric(students[,i])) {  
 normTest(students[,i], col.names[i])  
 }  
}

## age FALSE FALSE FALSE   
## Medu FALSE FALSE FALSE   
## Fedu FALSE FALSE FALSE   
## traveltime FALSE FALSE FALSE   
## studytime FALSE FALSE FALSE   
## failures FALSE FALSE FALSE   
## famrel FALSE FALSE FALSE   
## freetime FALSE FALSE FALSE   
## goout FALSE FALSE FALSE   
## Dalc FALSE FALSE FALSE   
## Walc FALSE FALSE FALSE   
## health FALSE FALSE FALSE   
## absences FALSE FALSE FALSE   
## G1 FALSE FALSE FALSE   
## G2 FALSE FALSE FALSE   
## G3 FALSE FALSE FALSE   
## score FALSE FALSE FALSE

normTest(students.female[,c("goout")], "goout~female")

## goout~female FALSE FALSE FALSE

normTest(students.male[,c("goout")], "goout~male")

## goout~male FALSE FALSE FALSE

normTest(students.female[,c("Walc")], "Walc~female")

## Walc~female FALSE FALSE FALSE

normTest(students.male[,c("Walc")], "Walc~male")

## Walc~male FALSE FALSE FALSE

normTest(students.female[,c("score")], "score~female")

## score~female FALSE TRUE FALSE

normTest(students.male[,c("score")], "score~male")

## score~male FALSE FALSE FALSE

normTest(students.studytime[,c("score")], "score~studytime")

## score~studytime TRUE TRUE FALSE

normTest(students.nostudytime[,c("score")], "score~nostudytime")

## score~nostudytime FALSE FALSE FALSE

normTest(students.mayores[,c("score")], "score~mayores")

## score~mayores FALSE FALSE FALSE

normTest(students.menores[,c("score")], "score~menores")

## score~menores TRUE TRUE TRUE

normTest(students.paid[,c("score")], "score~paid")

## score~paid TRUE TRUE TRUE

normTest(students.nopaid[,c("score")], "score~nopaid")

## score~nopaid FALSE FALSE FALSE

normTest(students.schoolsup[,c("score")], "score~schoolsup")

## score~schoolsup TRUE TRUE TRUE

normTest(students.noschoolsup[,c("score")], "score~noschoolsup")

## score~noschoolsup FALSE FALSE FALSE

normTest(students.famsup[,c("score")], "score~famsup")

## score~famsup FALSE TRUE FALSE

normTest(students.nofamsup[,c("score")], "score~nofamsup")

## score~nofamsup FALSE TRUE FALSE

Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos esta homogeneidad en cuanto a los grupos conformados por los vehículos que presentan un motor turbo frente a un motor estándar. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

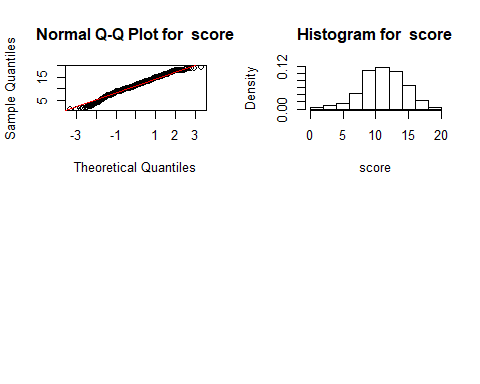
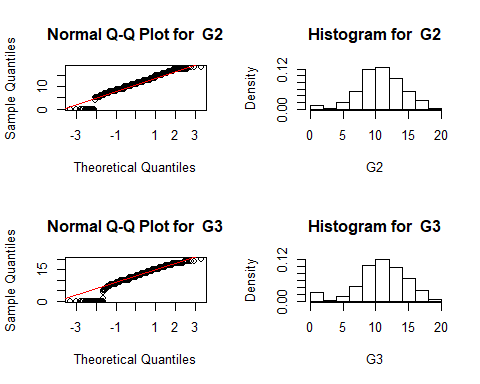
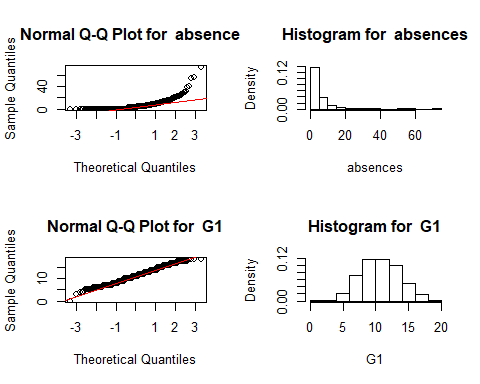
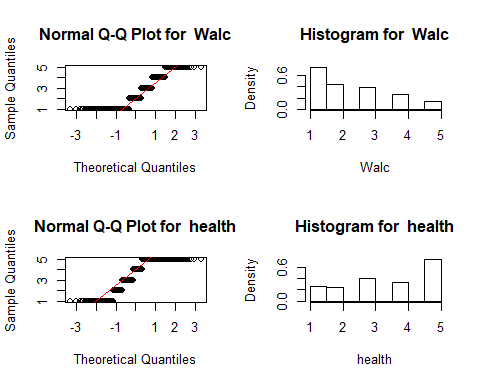
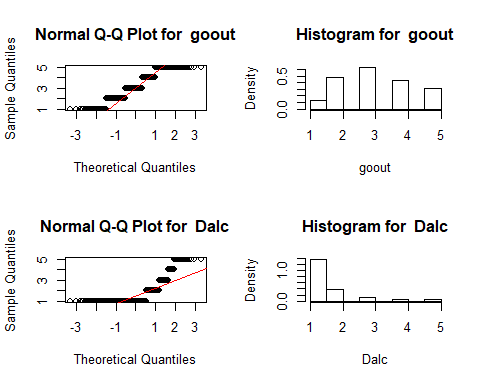
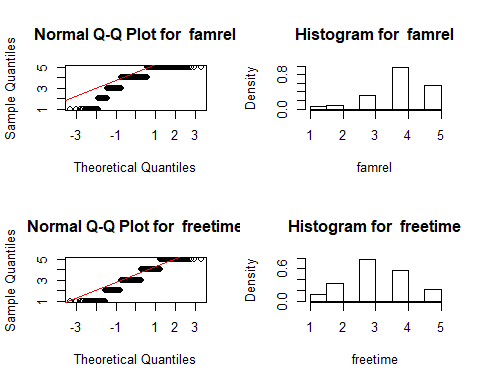
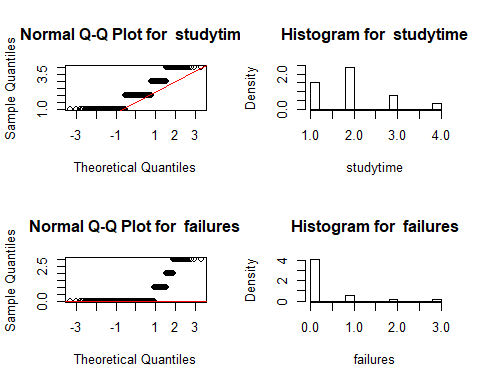
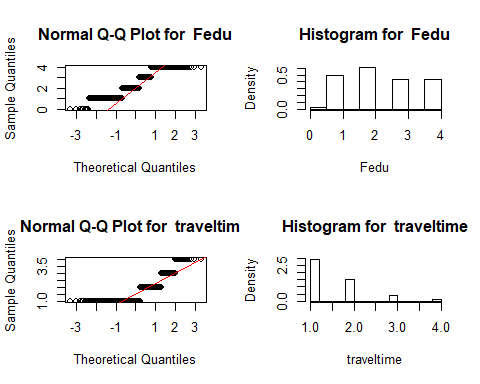
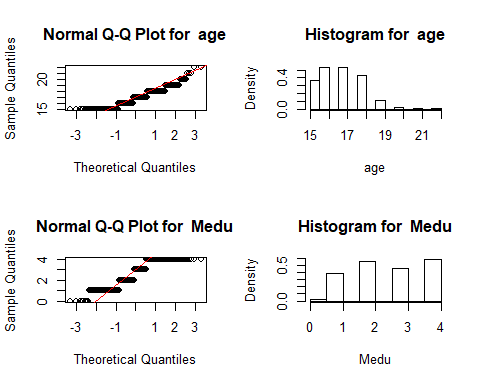
fligner.test(score ~ age, data = students)

##   
## Fligner-Killeen test of homogeneity of variances  
##   
## data: score by age  
## Fligner-Killeen:med chi-squared = 23.529, df = 7, p-value =  
## 0.001378

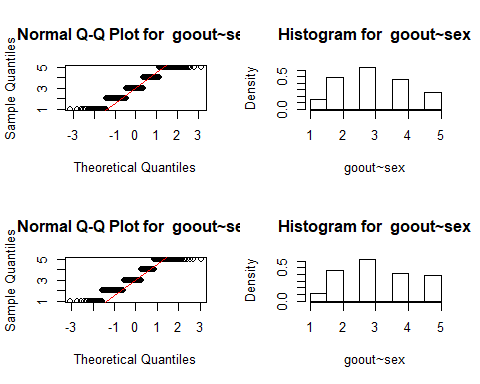
fligner.test(score ~ sex, data = students)

##   
## Fligner-Killeen test of homogeneity of variances  
##   
## data: score by sex  
## Fligner-Killeen:med chi-squared = 0.19607, df = 1, p-value =  
## 0.6579

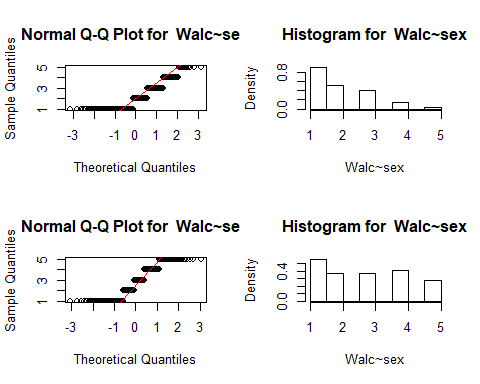
plotNormHistogram <- function(data, name) {  
 qqnorm(data,main = paste("Normal Q-Q Plot for ",name))  
 qqline(data,col="red")  
 hist(data,   
 main=paste("Histogram for ", name),   
 xlab=name, freq = FALSE)  
}  
  
par(mfrow=c(2,2))  
for(i in 1:ncol(students)) {  
 if (is.numeric(students[,i])){  
 plotNormHistogram(students[,i], colnames(students)[i])  
 }  
}



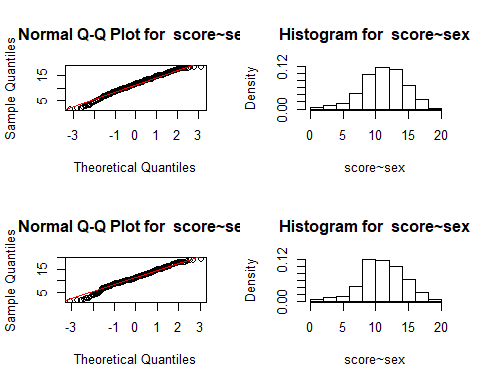
par(mfrow=c(2,2))  
plotNormHistogram(students[(students$sex == "F"),c("goout")], "goout~sex")  
plotNormHistogram(students[(students$sex == "M"),c("goout")], "goout~sex")



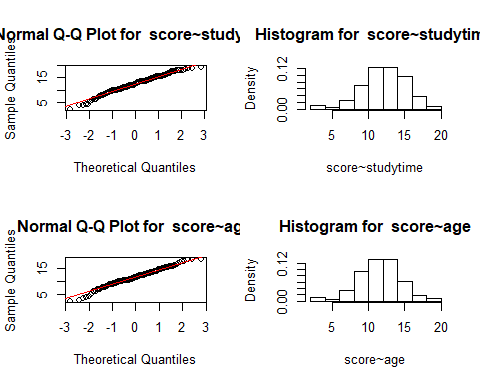
plotNormHistogram(students[(students$sex == "F"),c("Walc")], "Walc~sex")  
plotNormHistogram(students[(students$sex == "M"),c("Walc")], "Walc~sex")



plotNormHistogram(students[(students$sex == "F"),c("score")], "score~sex")  
plotNormHistogram(students[(students$sex == "M"),c("score")], "score~sex")



plotNormHistogram(students[(students$studytime > 2),c("score")], "score~studytime")  
plotNormHistogram(students[(students$age < 16),c("score")], "score~age")



## Pruebas estadísticas

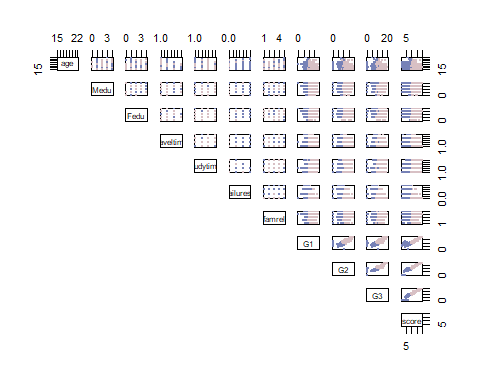
Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

### Correlaciones

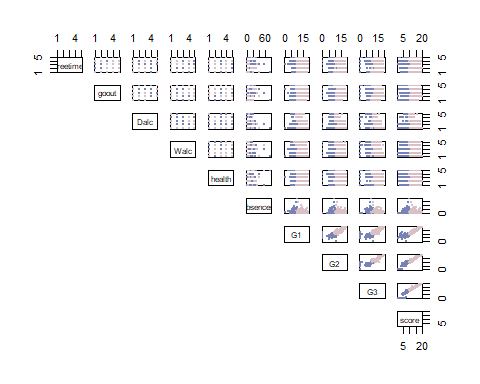
# Análisis de correlación  
corr\_matrix <- matrix(nc = 2, nr = 0)  
colnames(corr\_matrix) <- c("estimate", "p-value")  
# Calcular el coeficiente de correlación para cada variable cuantitativa  
# con respecto al campo "precio"  
for (i in 1:(ncol(students))) {  
 if (colnames(students)[i] != "score") {  
 if (is.integer(students[,i]) | is.numeric(students[,i])) {  
 spearman\_test = cor.test(students[,i],  
 students$score,  
 method = "spearman")  
 corr\_coef = spearman\_test$estimate  
 p\_val = spearman\_test$p.value  
   
 # Add row to matrix  
 pair = matrix(ncol = 2, nrow = 1)  
 pair[1][1] = corr\_coef  
 pair[2][1] = p\_val  
 corr\_matrix <- rbind(corr\_matrix, pair)  
 rownames(corr\_matrix)[nrow(corr\_matrix)] <- colnames(students)[i]  
 }  
 }  
}  
print(corr\_matrix)

## estimate p-value  
## age -0.12285038 6.898902e-05  
## Medu 0.24518910 9.298302e-16  
## Fedu 0.20340201 3.273473e-11  
## traveltime -0.12895585 2.925874e-05  
## studytime 0.20004048 6.947297e-11  
## failures -0.41737795 2.890952e-45  
## famrel 0.03959414 2.011456e-01  
## freetime -0.07551607 1.466446e-02  
## goout -0.11679110 1.555327e-04  
## Dalc -0.16924285 3.763900e-08  
## Walc -0.14355324 3.209590e-06  
## health -0.07083599 2.208430e-02  
## absences -0.10393033 7.705005e-04  
## G1 0.95083179 0.000000e+00  
## G2 0.97620215 0.000000e+00  
## G3 0.97637729 0.000000e+00

color=diverge\_hcl(length(students$mark))[rank(students$mark)]  
pairs(~ age + Medu + Fedu + traveltime + studytime + failures + famrel + G1 + G2 + G3 + score, data = students, pch = 19, cex = 0.5, lower.panel = NULL, col = color)



pairs(~ freetime + goout + Dalc + Walc + health + absences + G1 + G2 + G3 + score, data = students, pch = 19, cex = 0.5, lower.panel = NULL, col = color)



### ¿Las chicas sacan mejores notas que los chicos?

t.test(students.male$score, students.female$score, alternative = "less")

##   
## Welch Two Sample t-test  
##   
## data: students.male$score and students.female$score  
## t = -0.93229, df = 972.51, p-value = 0.1757  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf 0.1435574  
## sample estimates:  
## mean of x mean of y   
## 11.16115 11.34856

### ¿Quien más tiempo dedica al estudio saca mejores notas?

t.test(students.nostudytime$score, students.studytime$score, alternative = "less")

##   
## Welch Two Sample t-test  
##   
## data: students.nostudytime$score and students.studytime$score  
## t = -5.8354, df = 361.6, p-value = 5.964e-09  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf -0.9859761  
## sample estimates:  
## mean of x mean of y   
## 10.97236 12.34673

### Modelo lineal

smp\_siz = floor(0.75\*nrow(students))  
  
set.seed(123) # set seed to ensure you always have same random numbers generated  
train\_ind = sample(seq\_len(nrow(students)),size = smp\_siz) # Randomly identifies the rows equal to sample size ( defined in previous instruction) from all the rows of Smarket dataset and stores the row number in train\_ind  
train = students[train\_ind,] #creates the training dataset with row numbers stored in train\_ind  
test = students[-train\_ind,] # creates the test dataset excluding the row numbers mentioned in train\_ind

# Generación de varios modelos  
modelo1 <- lm(score ~ G1, data = train)  
modelo2 <- lm(score ~ G1 + G2, data = train)  
modelo3 <- lm(score ~ G1 + G2 + G3, data = train)  
modelo4 <- lm(score ~ G1 + G3, data = train)  
modelo5 <- lm(score ~ G1 + G3 + studytime, data = train)  
modelo6 <- lm(score ~ studytime + sex + absences, data = train)  
modelo7 <- lm(score ~ studytime + paid, data = train)  
modelo8 <- lm(score ~ G1 + G3 + studytime + paid, data = train)  
  
tabla.coeficientes <- matrix(  
 c(1, summary(modelo1)$r.squared,  
 2, summary(modelo2)$r.squared,  
 3, summary(modelo3)$r.squared,  
 4, summary(modelo4)$r.squared,  
 5, summary(modelo5)$r.squared,  
 6, summary(modelo6)$r.squared,  
 7, summary(modelo7)$r.squared,  
 8, summary(modelo8)$r.squared),  
 ncol = 2, byrow = TRUE)  
colnames(tabla.coeficientes) <- c("Modelo", "R^2")  
tabla.coeficientes

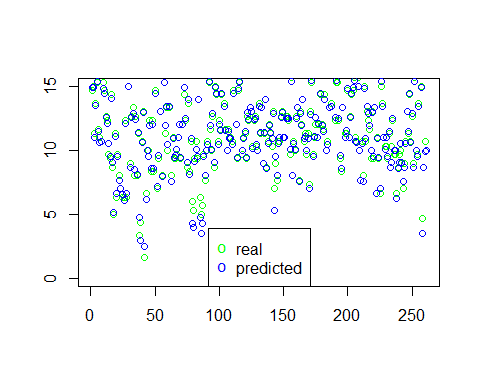
## Modelo R^2  
## [1,] 1 0.85213764  
## [2,] 2 0.97557027  
## [3,] 3 1.00000000  
## [4,] 4 0.98579767  
## [5,] 5 0.98580377  
## [6,] 6 0.03740947  
## [7,] 7 0.03915323  
## [8,] 8 0.98588380

summary(modelo1)

##   
## Call:  
## lm(formula = score ~ G1, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.0202 -0.3733 -0.0083 0.6564 7.3608   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.02746 0.17378 -0.158 0.875   
## G1 1.00397 0.01496 67.089 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.269 on 781 degrees of freedom  
## Multiple R-squared: 0.8521, Adjusted R-squared: 0.8519   
## F-statistic: 4501 on 1 and 781 DF, p-value: < 2.2e-16

y\_predict = predict(modelo8, test)

plot(test$score, col = "green", xlab = "", ylab = "", ylim = range(0,15))   
points(y\_predict, col = "blue")   
legend("bottom", c("real", "predicted"), pch = "o", col = c("green", "blue"), trace = TRUE)



## xchar= 11.2 ; (yextra,ychar)= 0 1.5   
## rect2(91.33,3.9, w=79.35, h=4.5, ...)  
## points2( 102.5 102.5 , 2.4 0.9 , pch= o o , ...)

## Conclusiones

# Recursos

Los siguientes recursos son de utilidad para la realización de la práctica: ● Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC. ● Megan Squire (2015). Clean Data. Packt Publishing Ltd. ● Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann. ● Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369. ● Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media. ● Wes McKinney (2012). Python for Data Analysis. O’Reilley Media, Inc. ● Tutorial de Github <https://guides.github.com/activities/hello-world>.