PracticaAnalisisDatos

Alvaro Hernandez

16 de marzo de 2016

ANALISIS DE DATOS R

Descargando y cargando archivos en dataframes

```
fileURL <- "https://archive.ics.uci.edu/ml/machine-learning-databases/00320/student.zip"
setwd("C:/Users/Usuario/Desktop/Experto BigData/Practicas/AnalisisDatosR")
getwd()
```

```
## [1] "C:/Users/Usuario/Desktop/Experto BigData/Practicas/AnalisisDatosR"
```

```
# download.file(fileURL,destfile="./datosAlumnos.zip")

list.files("./datosEstudiantes")
```

```
## [1] "student-mat.csv" "student-merge.R" "student-por.csv" "student.txt"
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.2.4
```

```
studentMat <- read.table("./datosEstudiantes/student-mat.csv",
                         row.names=NULL, sep=";", header=TRUE)

studentPor <- read.table("./datosEstudiantes/student-por.csv",
                         row.names=NULL, sep=";", header=TRUE)
class(studentMat)
```

```
## [1] "data.frame"
```

```
class(studentPor)
```

```
## [1] "data.frame"
```

Preparar los datos

Modificando los headers de los dos datasets

```r
#Cambiando los headers del dataset StudenPor a minusculas
names(studentMat)
```

```
##  [1] "school"     "sex"        "age"        "address"    "famsize"
##  [6] "Pstatus"    "Medu"       "Fedu"       "Mjob"       "Fjob"
## [11] "reason"     "guardian"   "traveltime" "studytime"  "failures"
## [16] "schoolsup"  "famsup"     "paid"       "activities" "nursery"
## [21] "higher"     "internet"   "romantic"   "famrel"     "freetime"
## [26] "goout"      "Dalc"       "Walc"       "health"     "absences"
## [31] "G1"         "G2"         "G3"
```

```r
names(studentMat) <- tolower(names(studentMat))
names(studentMat)
```

```
##  [1] "school"     "sex"        "age"        "address"    "famsize"
##  [6] "pstatus"    "medu"       "fedu"       "mjob"       "fjob"
## [11] "reason"     "guardian"   "traveltime" "studytime"  "failures"
## [16] "schoolsup"  "famsup"     "paid"       "activities" "nursery"
## [21] "higher"     "internet"   "romantic"   "famrel"     "freetime"
## [26] "goout"      "dalc"       "walc"       "health"     "absences"
## [31] "g1"         "g2"         "g3"
```

```r
#Cambiando los headers del dataset studentPor a minusculas
names(studentPor)
```

```
##  [1] "school"     "sex"        "age"        "address"    "famsize"
##  [6] "Pstatus"    "Medu"       "Fedu"       "Mjob"       "Fjob"
## [11] "reason"     "guardian"   "traveltime" "studytime"  "failures"
## [16] "schoolsup"  "famsup"     "paid"       "activities" "nursery"
## [21] "higher"     "internet"   "romantic"   "famrel"     "freetime"
## [26] "goout"      "Dalc"       "Walc"       "health"     "absences"
## [31] "G1"         "G2"         "G3"
```

```r
names(studentPor) <- tolower(names(studentPor))
names(studentPor)
```

```
##  [1] "school"     "sex"        "age"        "address"    "famsize"
##  [6] "pstatus"    "medu"       "fedu"       "mjob"       "fjob"
## [11] "reason"     "guardian"   "traveltime" "studytime"  "failures"
## [16] "schoolsup"  "famsup"     "paid"       "activities" "nursery"
## [21] "higher"     "internet"   "romantic"   "famrel"     "freetime"
## [26] "goout"      "dalc"       "walc"       "health"     "absences"
## [31] "g1"         "g2"         "g3"
```

```r
#Quitando _ de la columna MJOB
length(grep("_",studentMat$mjob))
```

```
## [1] 59
```

```
length(grep("_",studentPor$mjob))
```

```
## [1] 135
```

```
#Sustituimos los _ por espacios con gsub
studentMat$mjob <- gsub("_"," ",studentMat$mjob)
studentPor$mjob <- gsub("_"," ",studentPor$mjob)

#Numero de _ en la columna mjob de los dos datasets
length(grep("_",studentMat$mjob))
```

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

```
length(grep("_",studentPor$mjob))
```

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

```
#Quitando _ de la columna FJOB
length(grep("_",studentMat$fjob))
```

```
## [1] 20
```

```
length(grep("_",studentPor$fjob))
```

```
## [1] 42
```

```
#Sustituimos los _ por espacios con gsub
studentMat$fjob <- gsub("_"," ",studentMat$fjob)
studentPor$fjob <- gsub("_"," ",studentPor$fjob)

#Numero de _ en la columna fjob de los dos datasets
length(grep("_",studentMat$fjob))
```

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

```
length(grep("_",studentPor$fjob))
```

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

Creando un nuevo dataFrame a partir de los anteriores y ordenandolo por diferentes campos para ver mejor los datos

```
#Hacemos un nuevo dataFrame con los datos de los otros dos siempre que coincidan los campos
#"school","sex","age","address","famsize","pstatus","medu","fedu","mjob","fjob","reason","nursery","int
#a los que no sean iguales les aniadimos los sufijos mat y por segun corresponda
studentMatPor <- merge(studentMat,studentPor,
                       by=c("school","sex","age","address","famsize","pstatus",
                            "medu","fedu","mjob","fjob","reason","nursery","internet"),
                       all=FALSE, suffixes=c("mat","por"))

kable(studentMatPor[1:10,1:7])
```

school	sex	age	address	famsize	pstatus	medu
GP	F	15	R	GT3	T	1
GP	F	15	R	GT3	T	1
GP	F	15	R	GT3	T	2
GP	F	15	R	GT3	T	2
GP	F	15	R	GT3	T	3
GP	F	15	R	GT3	T	3
GP	F	15	R	GT3	T	3
GP	F	15	R	LE3	T	2
GP	F	15	R	LE3	T	3
GP	F	15	U	GT3	A	3

```
dim(studentMatPor)[1]
```

```
## [1] 382
```

```
#Vemos los encabezados de studentMatPor
names(studentMatPor)
```

```
##  [1] "school"        "sex"           "age"           "address"
##  [5] "famsize"       "pstatus"       "medu"          "fedu"
##  [9] "mjob"          "fjob"          "reason"        "nursery"
## [13] "internet"      "guardianmat"   "traveltimemat" "studytimemat"
## [17] "failuresmat"   "schoolsupmat"  "famsupmat"     "paidmat"
## [21] "activitiesmat" "highermat"     "romanticmat"   "famrelmat"
## [25] "freetimemat"   "gooutmat"      "dalcmat"       "walcmat"
## [29] "healthmat"     "absencesmat"   "g1mat"         "g2mat"
## [33] "g3mat"         "guardianpor"   "traveltimepor" "studytimepor"
## [37] "failurespor"   "schoolsuppor"  "famsuppor"     "paidpor"
## [41] "activitiespor" "higherpor"     "romanticpor"   "famrelpor"
## [45] "freetimepor"   "gooutpor"      "dalcpor"       "walcpor"
## [49] "healthpor"     "absencespor"   "g1por"         "g2por"
## [53] "g3por"
```

```
#y lo ordenamos por sexo, edad, tamaño de familia
studentMatPor <- studentMatPor

#indicamos la variable por la cual ordenar famsize ya que lo ordenaba por
#GT3 como el valor menor al no haberselo indicado
studentMatPor$famsize <- relevel(studentMatPor$famsize,ref="LE3")
```

```
studentMatPor <- studentMatPor[order(
        (studentMatPor[,"sex"]), (studentMatPor[,"age"]), (studentMatPor[,"famsize"])
                                    ),
                                ]
#Podemos ver como queda ordenado
kable(studentMatPor[1:10,1:7])
```

	school	sex	age	address	famsize	pstatus	medu
8	GP	F	15	R	LE3	T	2
9	GP	F	15	R	LE3	T	3
32	GP	F	15	U	LE3	A	3
33	GP	F	15	U	LE3	A	3
34	GP	F	15	U	LE3	A	4
35	GP	F	15	U	LE3	T	1
36	GP	F	15	U	LE3	T	3
37	GP	F	15	U	LE3	T	4
38	GP	F	15	U	LE3	T	4
1	GP	F	15	R	GT3	T	1

Utilizando cast sobre los dataframes para explorar los datos

```
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.2.4
```

```
#media de nota final por trabajo del padre y de la madre
jobG3 <- cast(studentMat, mjob~fjob, mean, value=c("g3"))
jobG3
```

```
##       mjob   at home health     other services  teacher
## 1  at home 12.285714  11.50  8.878788  8.80000  3.00000
## 2   health       NaN  13.50 11.588235 12.40000 11.00000
## 3    other  9.200000  12.00  9.798077  9.50000 11.33333
## 4 services  8.166667  10.25 11.357143 10.76744 13.12500
## 5  teacher 11.000000  10.00 10.761905 10.31579 13.08333
```

```
#tiempo libre medio por edad y colegio
ftAgeSchool <- cast(studentMat, age~school, mean, value=c("freetime"))
ftAgeSchool
```

```
##   age       GP       MS
## 1  15 3.280488      NaN
## 2  16 3.230769      NaN
## 3  17 3.162791 3.666667
## 4  18 3.157895 3.040000
## 5  19 3.277778 3.166667
## 6  20 5.000000 4.500000
## 7  21      NaN 5.000000
## 8  22 4.000000      NaN
```

Transformacion de los datos

Creando variables categoricas

```
#con cut2 creamos una variable categorica para las notas finales
#en matematicas y portugues de el dataframe studentMatPor
#dividimos las notas en intervalos 0-10 10-17 17-valor maximo de la nota.
maxg3mat = max(studentMatPor$g3mat)
maxg3mat
```

```
## [1] 20
```

```
maxg3por = max(studentMatPor$g3por)
maxg3por
```

```
## [1] 19
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.2.4
```

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

```
## Warning: package 'lattice' was built under R version 3.2.4
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.2.3
```

```
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 3.2.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```r
notaG3Mat <-cut2(studentMatPor$g3mat, c(10,17,maxg3mat))

notaG3Por <-cut2(studentMatPor$g3por, c(10,17,maxg3por))

#con levels cambiamos los nombres de las categorias
levels(notaG3Mat) <-c("Suspenso", "Aprobado", "Sobresaliente")

levels(notaG3Por) <-c("Suspenso", "Aprobado", "Sobresaliente")

#Nota final matematicas
table(notaG3Mat)
```

```
## notaG3Mat
##      Suspenso      Aprobado Sobresaliente
##           127           230            25
```

```r
#comprobamos en los datasets que los resultados sean correctos

#Intervalo [0-10)
length(studentMatPor[studentMatPor$g3mat>=0 & studentMatPor$g3mat<10,c("g3mat")])
```

```
## [1] 127
```

```r
#Intervalo [10-17)
length(studentMatPor[studentMatPor$g3mat>=10 & studentMatPor$g3mat<17,c("g3mat")])
```

```
## [1] 230
```

```r
#Intervalo [17-maximo)
length(studentMatPor[studentMatPor$g3mat>=17,c("g3mat")])
```

```
## [1] 25
```

```r
#Nota final portugues
table(notaG3Por)
```

```
## notaG3Por
##      Suspenso      Aprobado Sobresaliente
##            32           320            30
```

```r
#comprobamos en los datasets que los resultados sean correctos

#Intervalo [0-10)
length(studentMatPor[studentMatPor$g3por>=0 & studentMatPor$g3por<10,c("g3por")])
```

```
## [1] 32
```

```r
#Intervalo [10-17)
length(studentMatPor[studentMatPor$g3por>=10 & studentMatPor$g3por<17,c("g3por")])
```

```
## [1] 320
```

```r
#Intervalo [17-maximo)
length(studentMatPor[studentMatPor$g3por>=17,c("g3por")])
```

```
## [1] 30
```

```r
#Añadimos las notas finales con nuestras categorias a los datasets y
#comprobamos que se han añadido correctamente
studentMatPor$finalg3mat <- notaG3Mat

studentMatPor$finalg3por <- notaG3Por

#Nota final categorica mat
kable(studentMatPor$finalg3mat[1:4])
```

Suspenso
Aprobado
Aprobado
Aprobado

```r
#Nota final categorica por
kable(studentMatPor$finalg3por[1:4])
```

Aprobado
Aprobado
Aprobado
Aprobado

Exploracion datos apply dplyr

```r
#Media de notas, en matematicas y en portugues
mediaG3 <- list(g3mat=c(studentMatPor$g3mat), g3por=c(studentMatPor$g3por))
a <- lapply(mediaG3,mean)
a
```

```
## $g3mat
## [1] 10.38743
##
## $g3por
## [1] 12.51571
```

```r
class(a)
```

```
## [1] "list"
```

```r
#Media de ausencias en cada nota de matematicas
x <- tapply(studentMatPor$absencesmat,studentMatPor$g3mat,mean)
x
```

```
##         0         4         5         6         7         8         9
##  0.000000 22.000000 11.428571  8.066667  7.428571  8.806452 10.851852
##        10        11        12        13        14        15        16
##  4.607143  6.604651  3.900000  6.160000  4.000000  2.937500  3.058824
##        17        18        19        20
##  3.666667  5.923077  4.200000  4.000000
```

```r
class(x)
```

```
## [1] "array"
```

```r
#Media de ausencias en cada nota de portugues
y <- tapply(studentMatPor$absencespor,studentMatPor$g3por,mean)
y
```

```
##         0         1         5         6         7         8         9
##  0.000000  0.000000 12.000000 16.000000  4.333333  9.750000  3.454545
##        10        11        12        13        14        15        16
##  5.195652  3.851852  3.654545  2.444444  4.121951  2.212121  3.857143
##        17        18        19
##  2.315789  1.555556  2.000000
```

```r
class(y)
```

```
## [1] "array"
```

```r
#Media de horas de estudio de cada dataframe
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     combine, src, summarize
```

```
## The following object is masked from 'package:reshape':
##
##     rename
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
studentMatPor %>%
        summarise(mediaStudyTimeMat=mean(studytimemat),
                mediaStudyTimePor=mean(studytimepor))
```

```
##   mediaStudyTimeMat mediaStudyTimePor
## 1          2.034031          2.039267
```

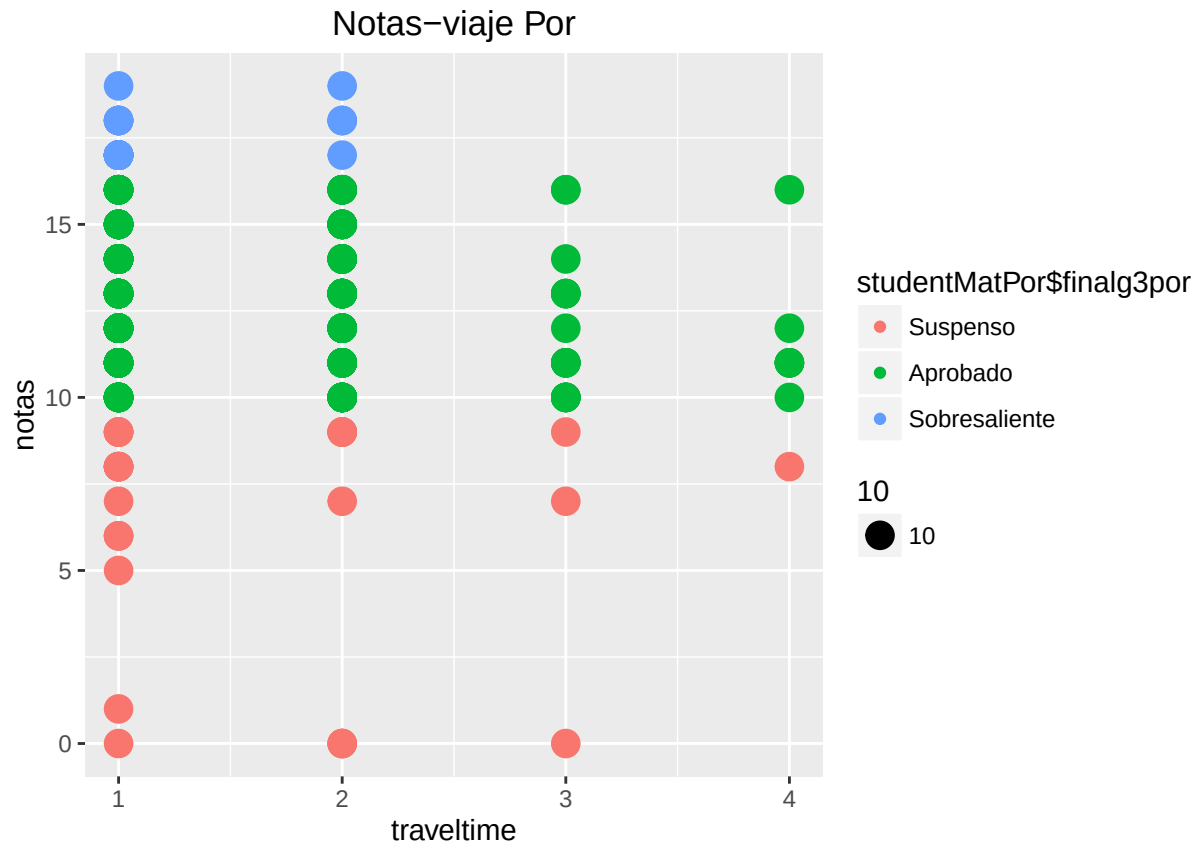
Analisis exploratorio, graficos

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.3
```

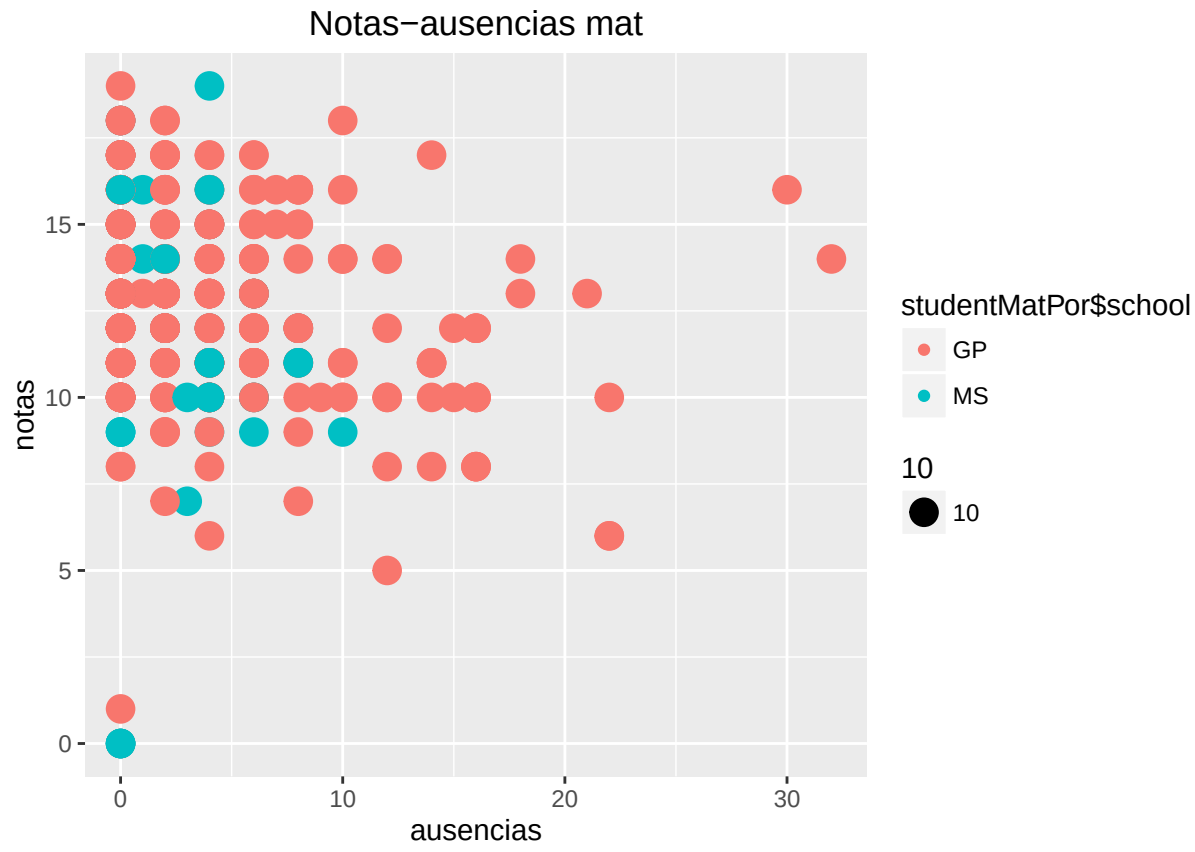
```
library(ggplot2)

#Primero veremos como influye en la nota el tiempo de viaje en portugues
plotTraveltimePor = qplot(data=studentMatPor,x=traveltimepor ,y=g3por, xlab="traveltime", ylab="notas",
plotTraveltimePor
```

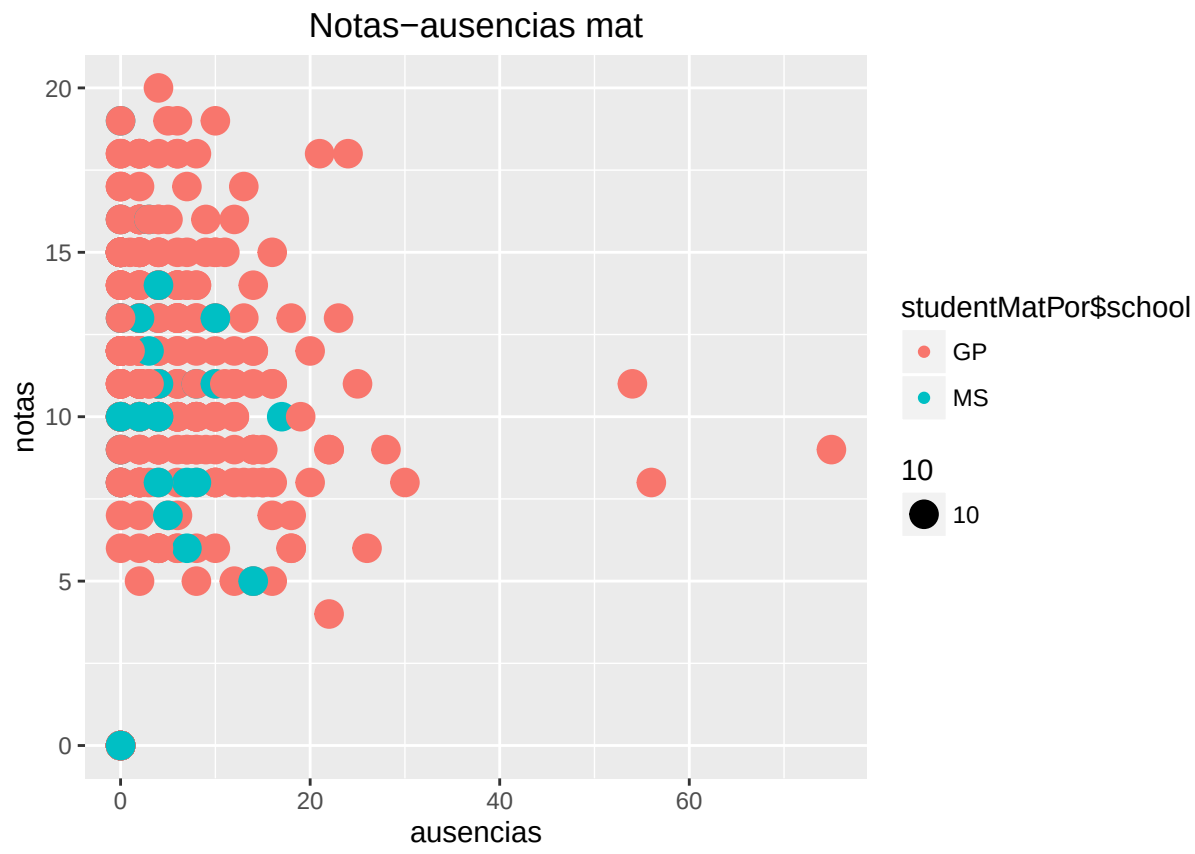



```
#vemos que los alumnos con notas mas altas tienen menos tiempo de viaje hasta el colegio


plotAbsencesPor = qplot(data=studentMatPor ,x=absencespor ,y=g3por,
                        xlab="ausencias", ylab="notas", color=studentMatPor$school,
                        main="Notas-ausencias mat", size = 10)
plotAbsencesPor
```




```
plotAbsencesMat = qplot(data=studentMatPor ,x=absencesmat ,y=g3mat,
                        xlab="ausencias", ylab="notas", color=studentMatPor$school,
                        main="Notas-ausencias mat", size = 10)
plotAbsencesMat
```




```
#En ambos casos el numero de ausencias de MS es menor que el de GP
```