DECISION TREES and LOGISTIC REGRESSION to predict the GRADE (no pass/pass)

Bogdan Tanasa

1. INTRODUCTION

We are using the data from **UCI** : !(https://archive.ics.uci.edu/ml/datasets/Student+Performance)

We are reading a file about **STUDENTS**, and we aim to predict whether they have passed or not the exams **(PASS/no_PASS)**;

The attributes in the **INPUT FILE** are the following :

- 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
- 2 sex - student's sex (binary: "F" - female or "M" - male)
- 3 age - student's age (numeric: from 15 to 22)
- 4 address - student's home address type (binary: "U" - urban or "R" - rural)
- 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
- 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
- 7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 11 reason - reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
- 12 guardian - student's guardian (nominal: "mother", "father" or "other")
- 13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
- 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- 16 schoolsup - extra educational support (binary: yes or no)
- 17 famsup - family educational support (binary: yes or no)
- 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- 19 activities - extra-curricular activities (binary: yes or no)
- 20 nursery - attended nursery school (binary: yes or no)
- 21 higher - wants to take higher education (binary: yes or no)
- 22 internet - Internet access at home (binary: yes or no)
- 23 romantic - with a romantic relationship (binary: yes or no)
- 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

- 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 29 health - current health status (numeric: from 1 - very bad to 5 - very good)
- 30 absences - number of school absences (numeric: from 0 to 93)

2. DATA EXPLORATION

```
library(ggplot2)
library(reshape2)
library(readxl)
library(dplyr)
library(tidyr)
library(purrr)
library(ggpubr)
library(broom)
library(tibble)
library(class)
library(gmodels)
library(caret)
library(e1071)
library(ISLR)
library(pROC)
library(lattice)
library(kknn)
library(multiROC)
library(MLeval)
library(AppliedPredictiveModeling)
library(corrplot)
library(Hmisc)

#########################################################
#########################################################
#########################################################
#########################################################

FILE1="student.mat.txt"

#########################################################
#########################################################
#########################################################
#########################################################

student <- read.delim(FILE1, sep="\t", header=T, stringsAsFactors=F)

#########################################################
#########################################################

summary(student)
```

```
##     school              sex                 age         address
##  Length:395         Length:395         Min.   :15.0   Length:395
##  Class :character   Class :character   1st Qu.:16.0   Class :character
##  Mode  :character   Mode  :character   Median :17.0   Mode  :character
##                                        Mean   :16.7
##                                        3rd Qu.:18.0
##                                        Max.   :22.0
```

```
##    famsize            Pstatus               Medu            Fedu
##  Length:395         Length:395         Min.   :0.000   Min.   :0.000
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000
##  Mode  :character   Mode  :character   Median :3.000   Median :2.000
##                                        Mean   :2.749   Mean   :2.522
##                                        3rd Qu.:4.000   3rd Qu.:3.000
##                                        Max.   :4.000   Max.   :4.000
##      Mjob               Fjob              reason            guardian
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    traveltime      studytime        failures       schoolsup
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:395
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##     famsup              paid            activities          nursery
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     higher             internet           romantic             famrel
##  Length:395         Length:395         Length:395         Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:4.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                           Mean   :3.944
##                                                           3rd Qu.:5.000
##                                                           Max.   :5.000
##     freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##      health         absences            G1              G2
##  Min.   :1.000   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00
##  1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00
##  Median :4.000   Median : 4.000   Median :11.00   Median :11.00
##  Mean   :3.554   Mean   : 5.709   Mean   :10.91   Mean   :10.71
##  3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00
##  Max.   :5.000   Max.   :75.000   Max.   :19.00   Max.   :19.00
##        G3
##  Min.   : 0.00
##  1st Qu.: 8.00
##  Median :11.00
##  Mean   :10.42
```

```
##  3rd Qu.:14.00
##  Max.   :20.00
```

```
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```
class(student)
```

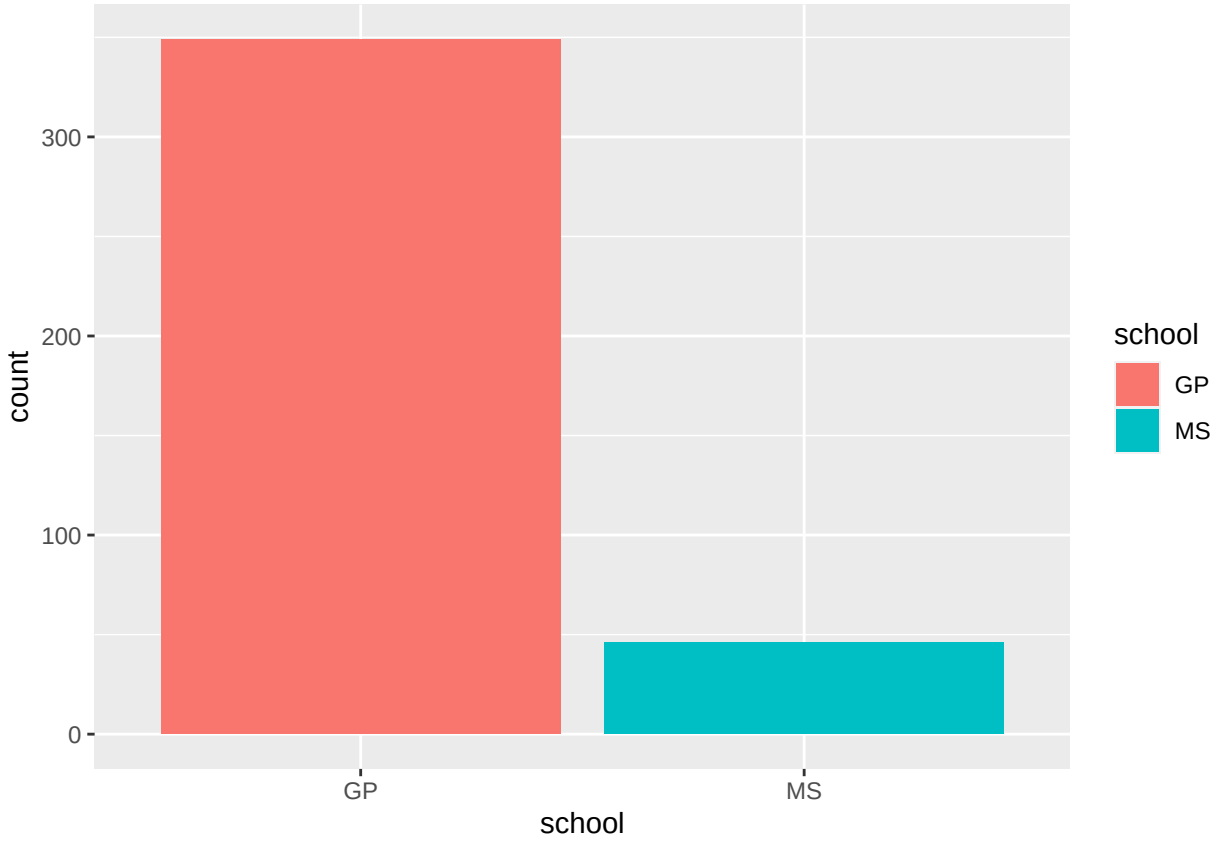
```
## [1] "data.frame"
```

Here we are starting to display the data for visual exploration.

```
##############################################################################################################
##############################################################################################################
# 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)

unique(student$school)
```

```
## [1] "GP" "MS"
```

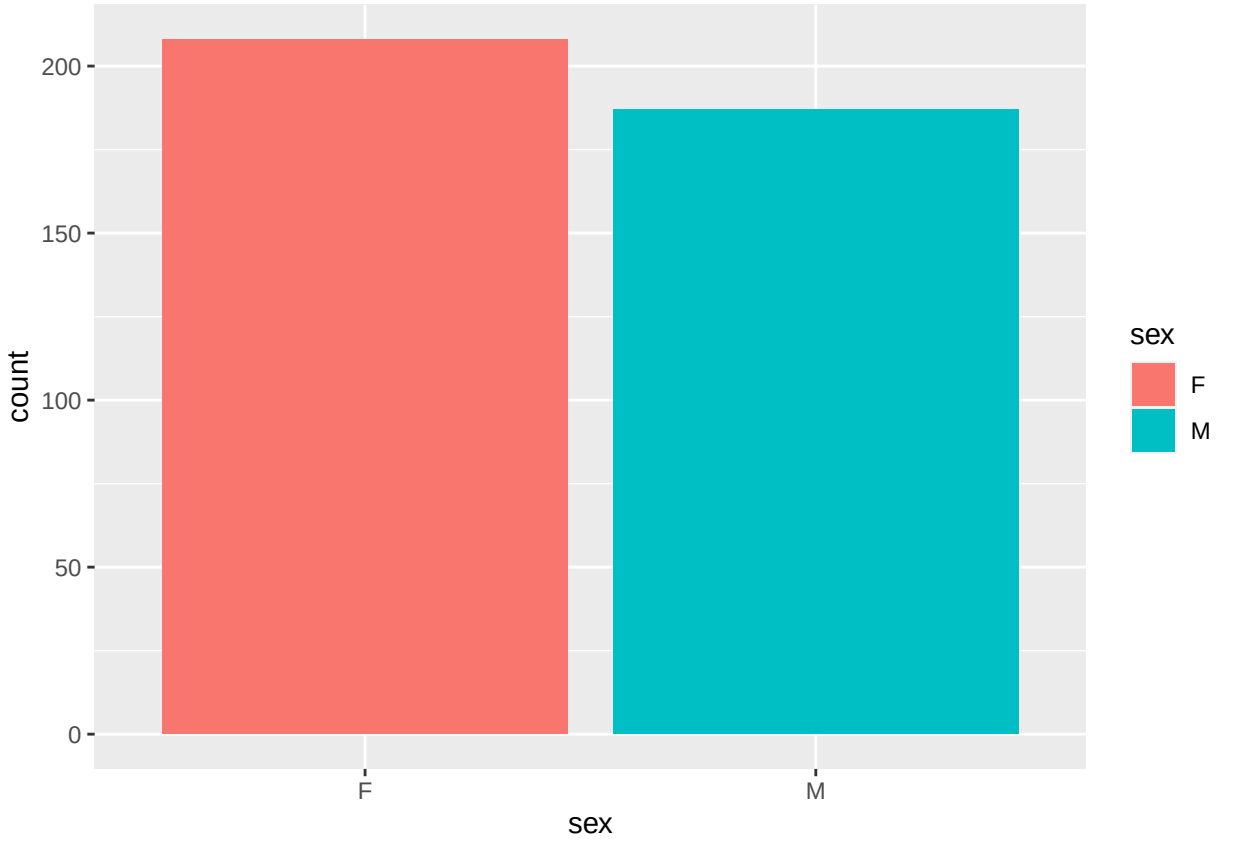
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=school, fill=school))
```




```
# ggsave("display.1.school.png")
student$school = as.factor(student$school)

##############################################################################################################
##############################################################################################################
# 2 sex - student's sex (binary: "F" - female or "M" - male)

unique(student$sex)
```

```
## [1] "F" "M"
```

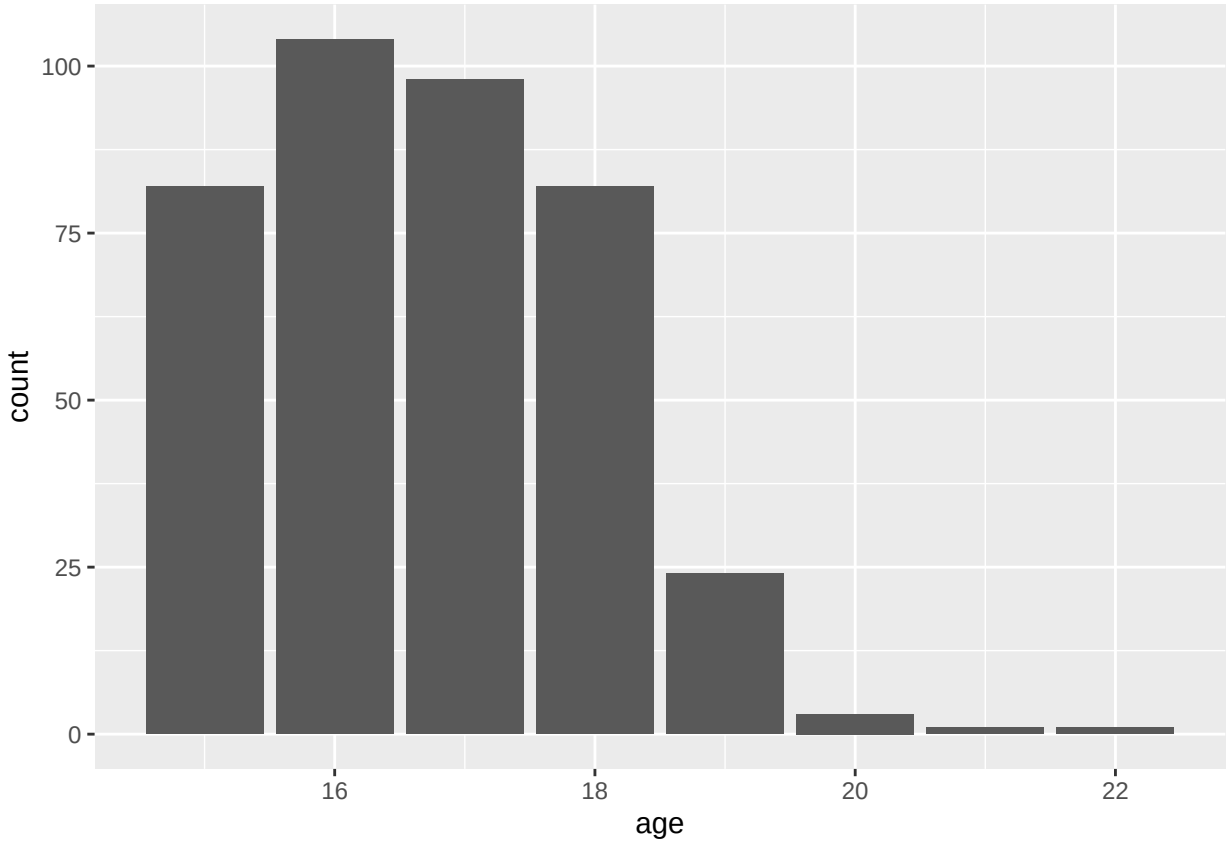
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=sex , fill=sex))
```




```
# ggsave("display.2.sex.png")
student$sex = as.factor(student$sex)

##############################################################################################################
##############################################################################################################
# 3 age - student's age (numeric: from 15 to 22)

unique(student$age)
```

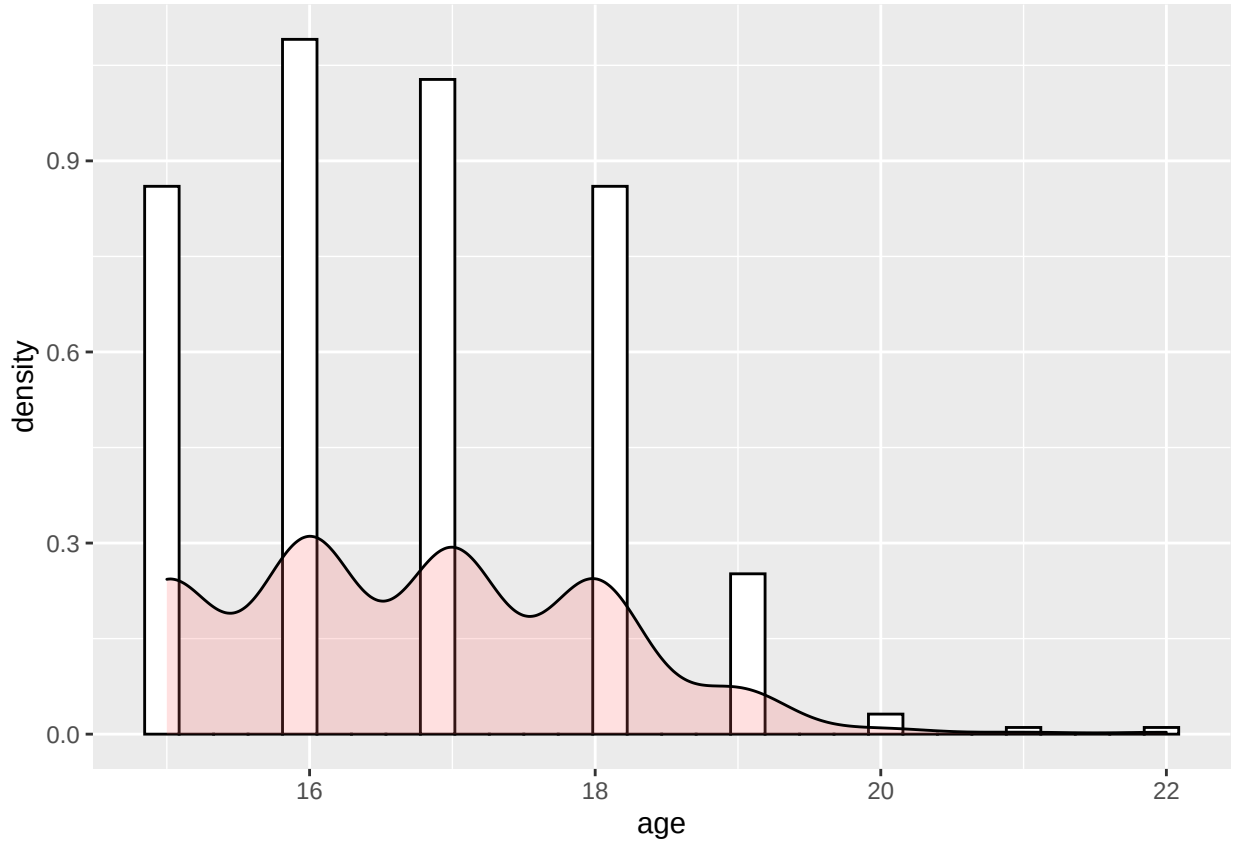
```
## [1] 18 17 15 16 19 22 20 21
```

```
ggplot(data = student) +
       geom_bar(mapping = aes(x=age , fill=age))
```




```
ggplot(data=student, aes(x=age)) +
      geom_histogram(aes(y=..density..), colour="black", fill="white")+
      geom_density(alpha=.2, fill="#FF6666")
```

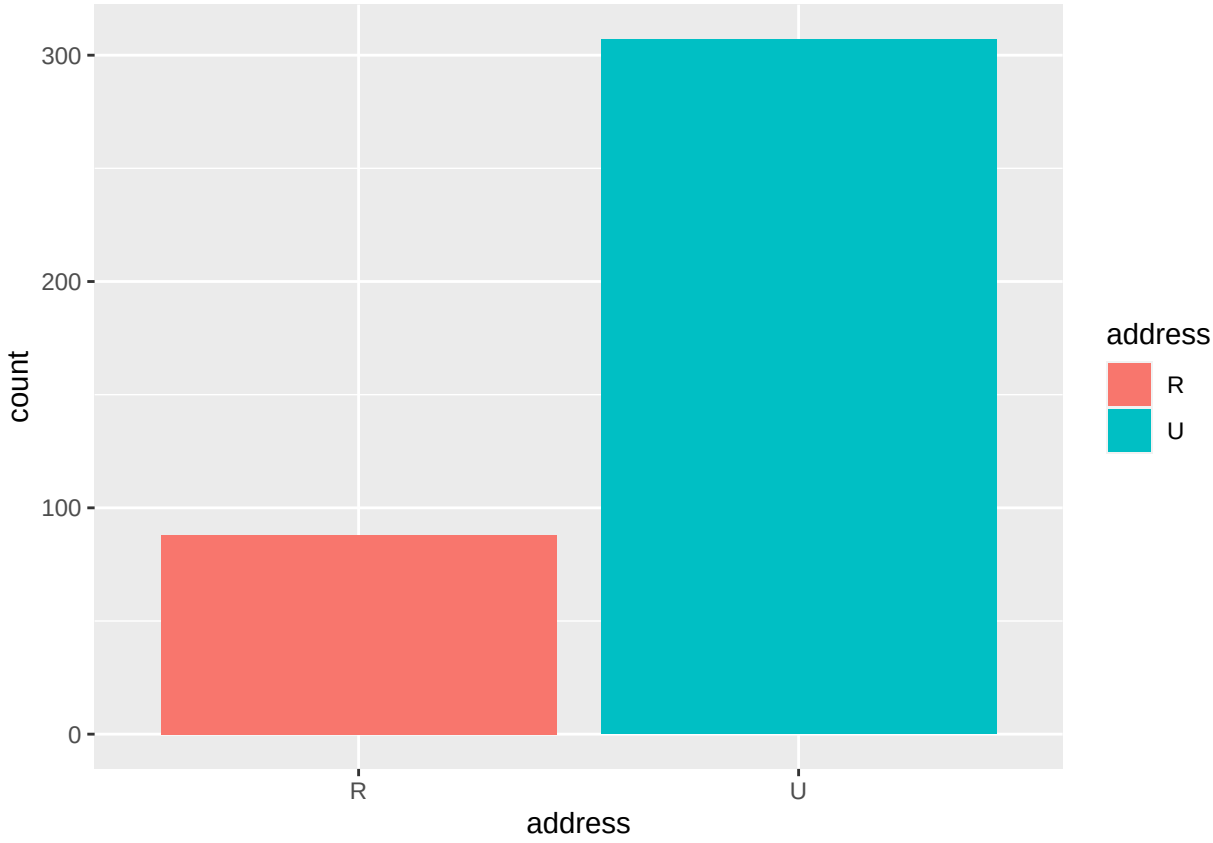
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggsave("display.3.age.png")
# AGE is already on the numerical scale !!
student$age = as.integer(student$age)

##########################################################################################################
##########################################################################################################
# 4 address - student's home address type (binary: "U" - urban or "R" - rural)

unique(student$address) ## [1] "U" "R"
```

```
## [1] "U" "R"
```

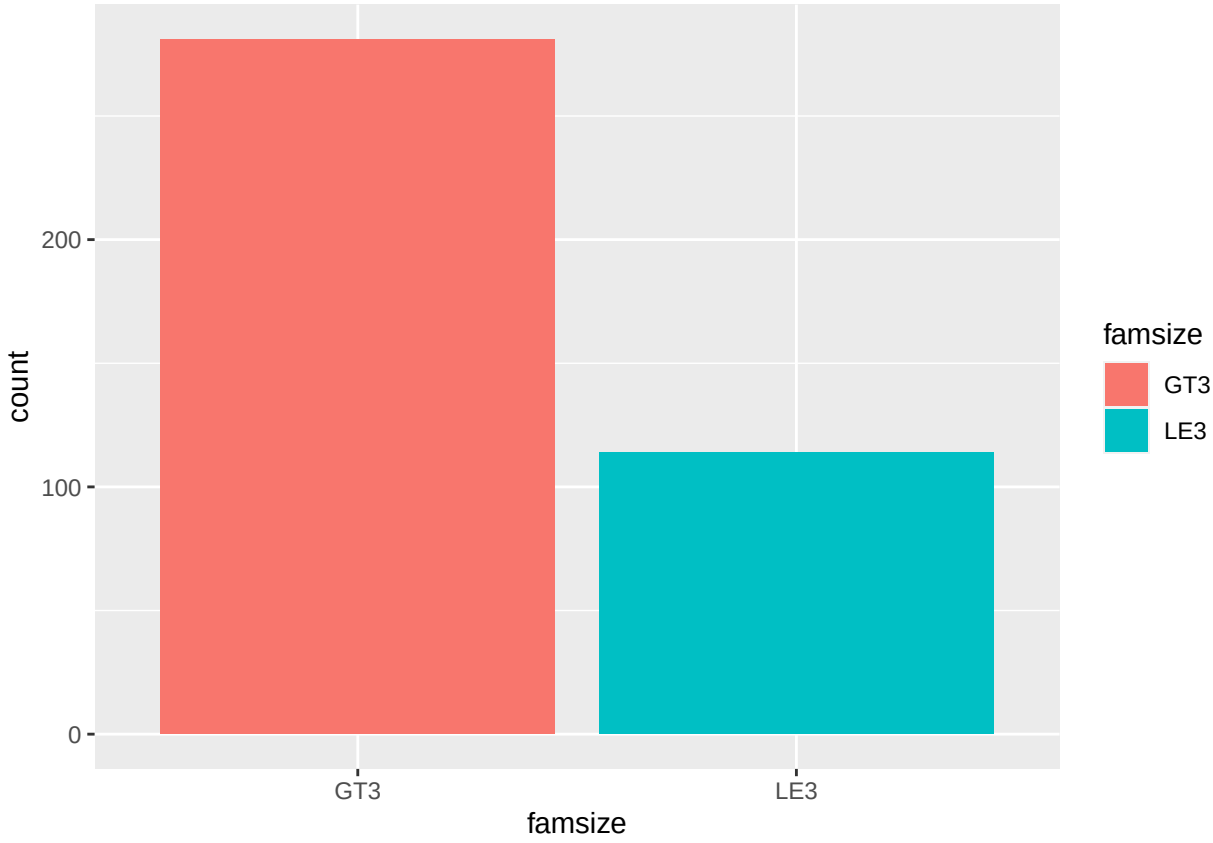
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=address, fill=address))
```




```
# ggsave("display.4.address.png")
student$address = as.factor(student$address)

##############################################################################################################
##############################################################################################################
# 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)

unique(student$famsize)
```

```
## [1] "GT3" "LE3"
```

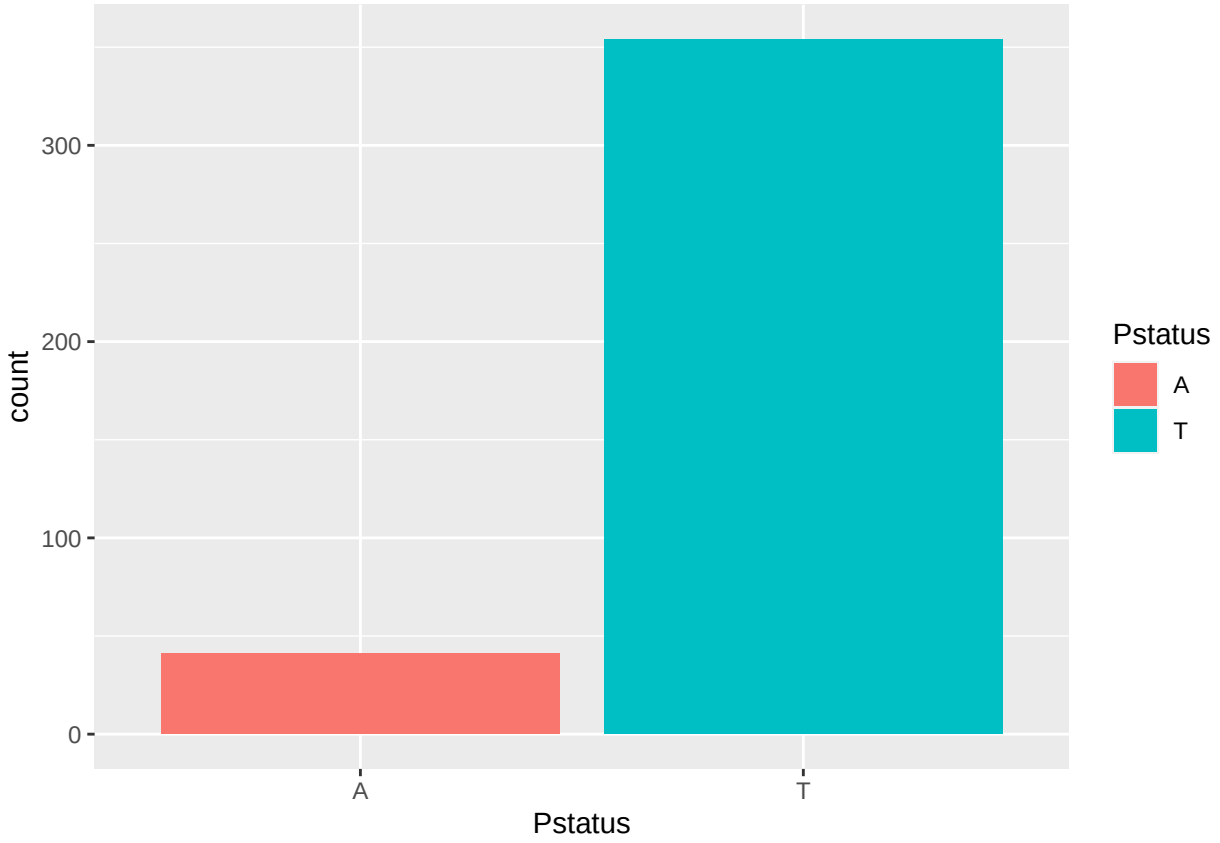
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=famsize, fill=famsize))
```




```
# ggsave("display.5.famsize.png")
student$famsize = as.factor(student$famsize)

##############################################################################################################
##############################################################################################################
# 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)

unique(student$Pstatus)
```

```
## [1] "A" "T"
```

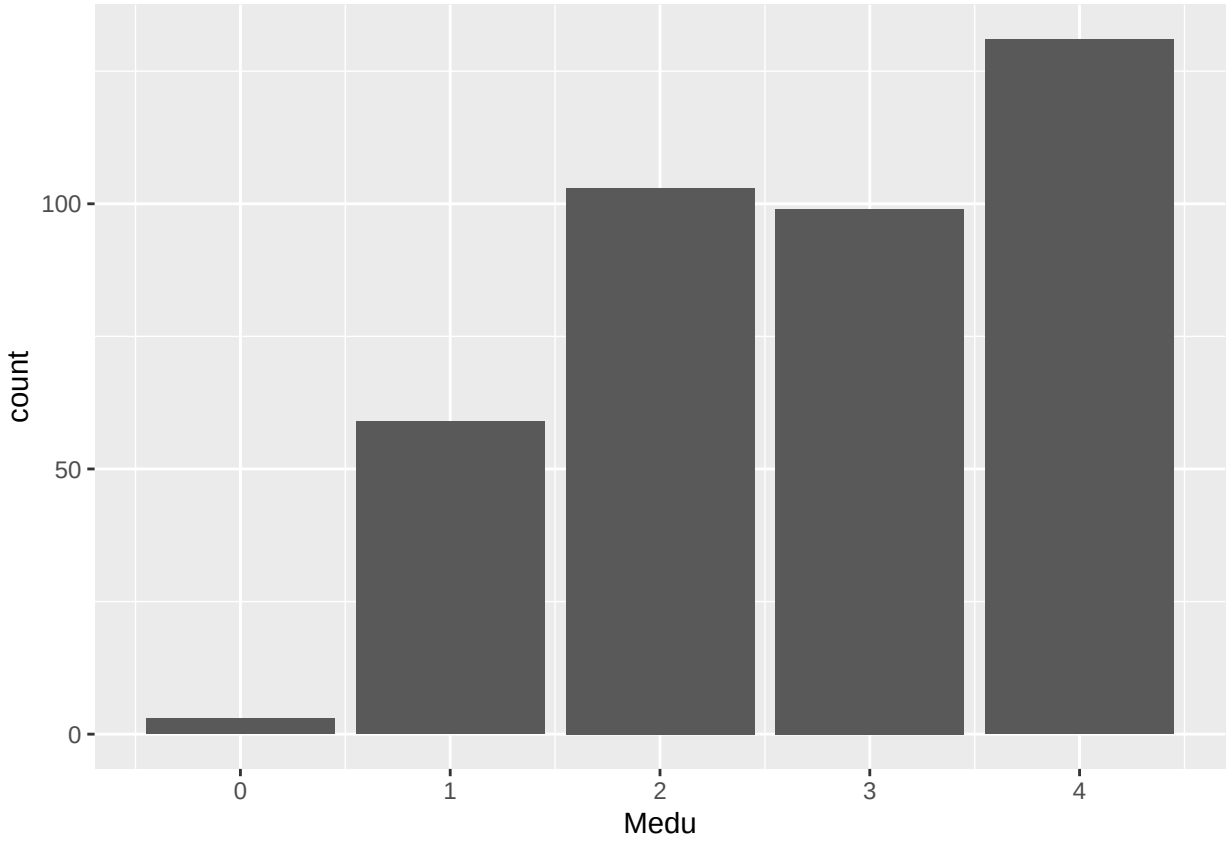
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Pstatus, fill=Pstatus))
```




```
# ggsave("display.6.Pstatus.png")
student$Pstatus = as.factor(student$Pstatus)

##############################################################################################################
##############################################################################################################
# 7 Medu - mother's education (numeric: 0 - none,  1 - primary education (4th grade), 2 â€" 5th to 9th

unique(student$Medu)
```

```
## [1] 4 1 3 2 0
```

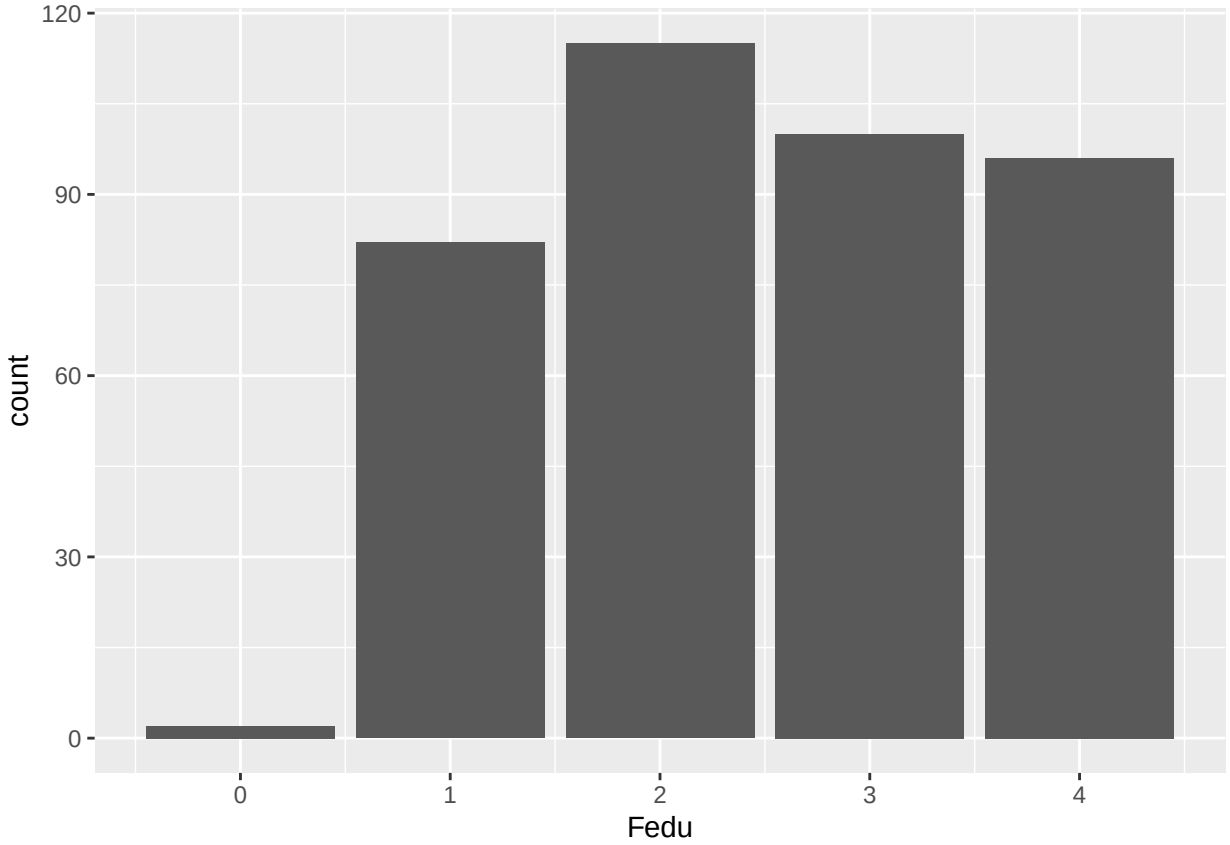
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Medu, fill=Medu))
```




```
# ggsave("display.7.Medu.png")
# we may wanna use the numerical values in various regression models
student$Medu = as.integer(student$Medu)

##############################################################################################################
##############################################################################################################
# 8 Fedu - father's education (numeric: 0 - none,  1 - primary education (4th grade), 2 â€" 5th to 9th

unique(student$Fedu)
```

```
## [1] 4 1 2 3 0
```

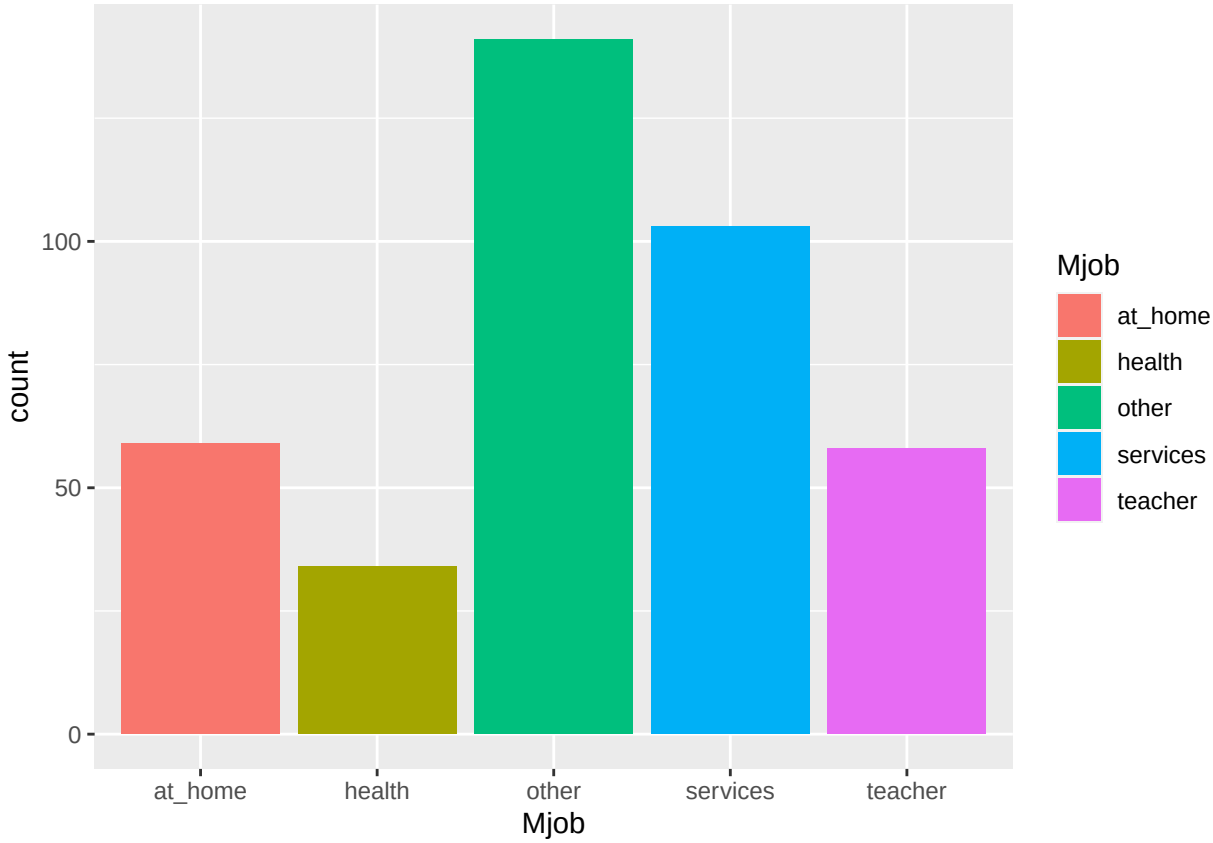
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Fedu, fill=Fedu))
```




```
# ggsave("display.8.Fedu.png")
# we may wanna use the numerical values in various regression models
student$Fedu = as.integer(student$Fedu)

##############################################################################################################
##############################################################################################################
# 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrati

unique(student$Mjob)
```

```
## [1] "at_home"  "health"   "other"    "services" "teacher"
```

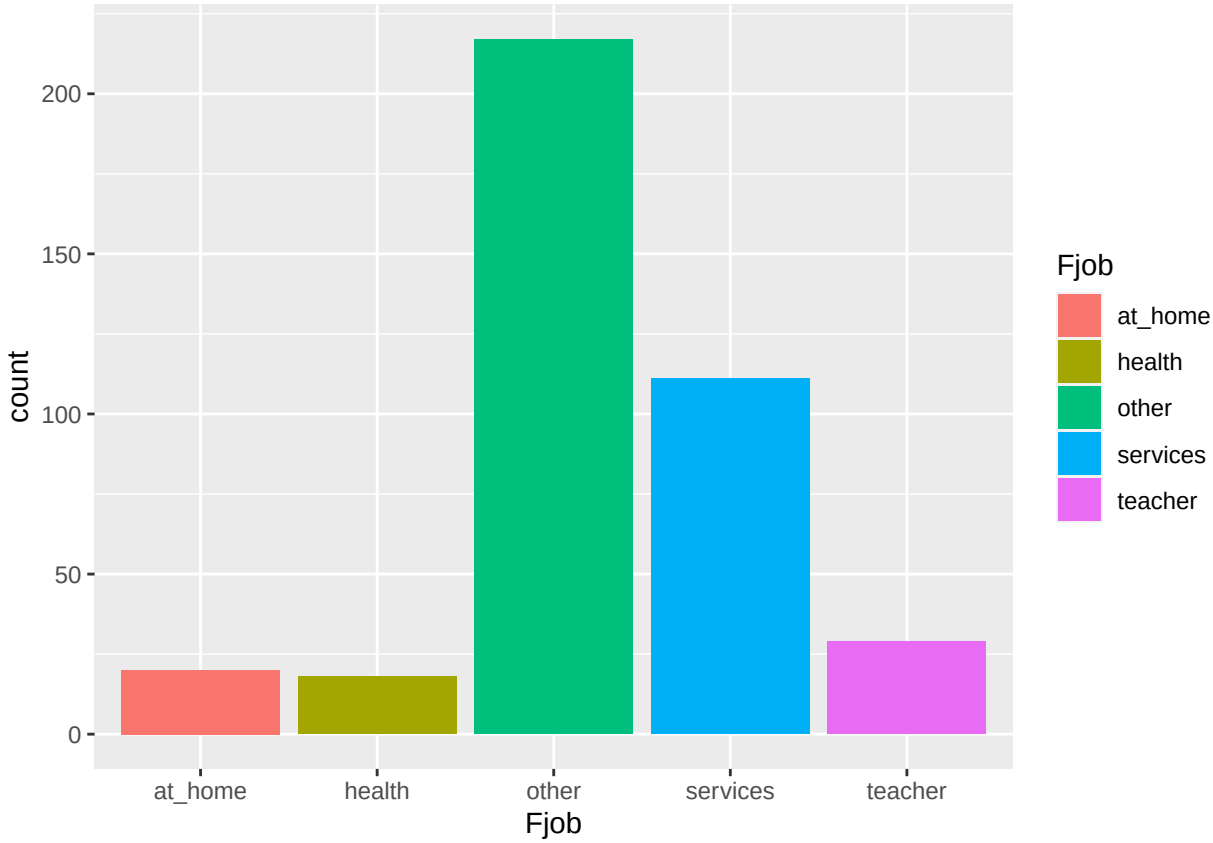
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Mjob, fill=Mjob))
```




```
# ggsave("display.9.Mjob.png")
student$Mjob = as.factor(student$Mjob)

##############################################################################################################
##############################################################################################################
# 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrat

unique(student$Fjob)
```

```
## [1] "teacher"  "other"    "services" "health"   "at_home"
```

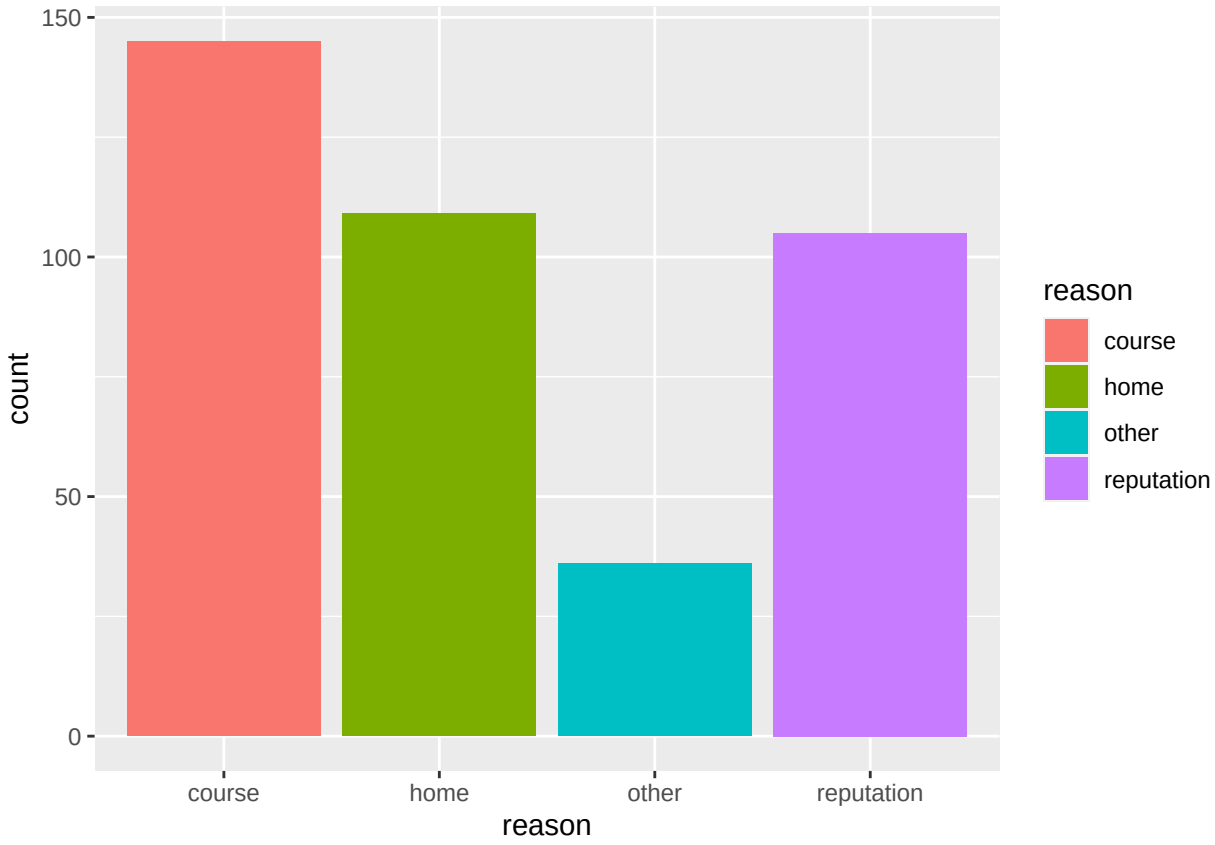
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Fjob, fill=Fjob))
```




```
# ggsave("display.10.Fjob.png")
student$Fjob = as.factor(student$Fjob)

##############################################################################################################
##############################################################################################################
# 11 reason - reason to choose this school (nominal: close to "home", school "reputation", "course" pre

unique(student$reason)
```

```
## [1] "course"     "other"      "home"       "reputation"
```

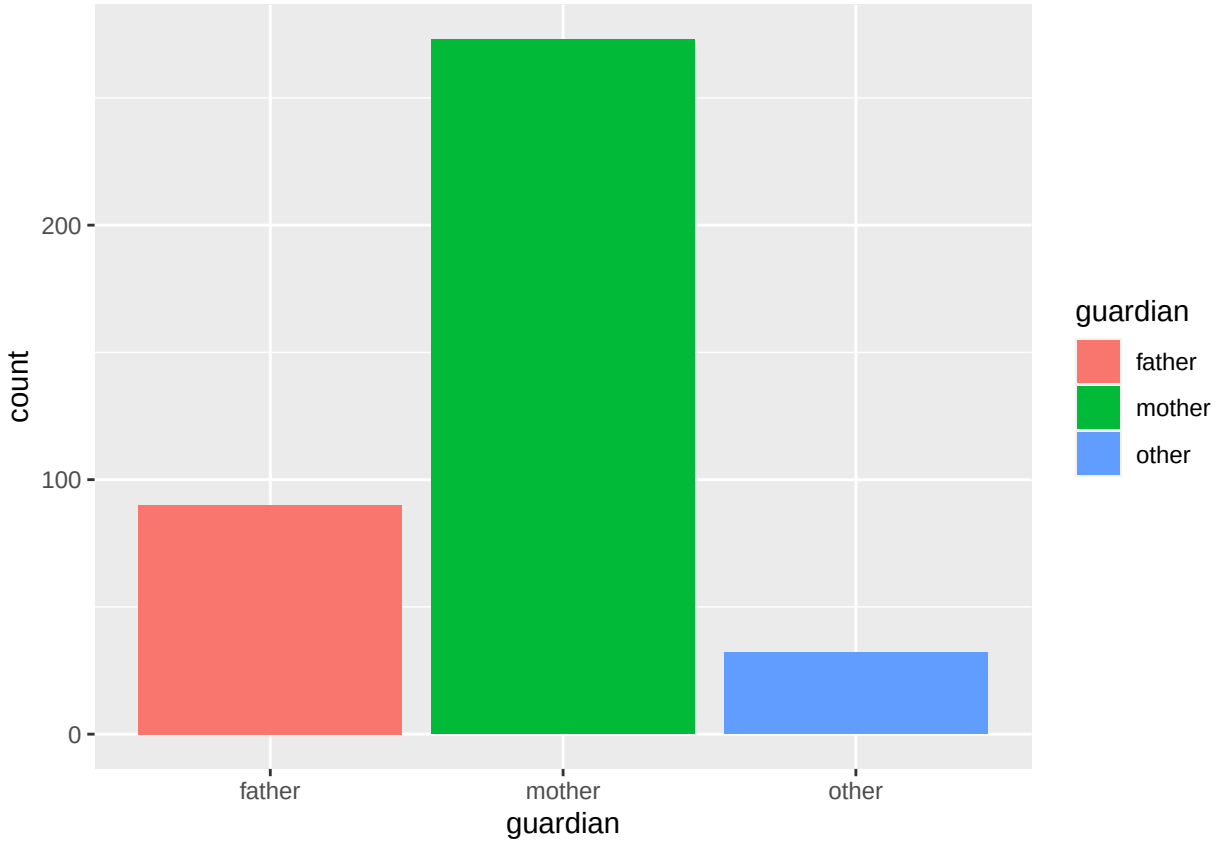
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=reason, fill=reason))
```




```
# ggsave("display.11.reason.png")
student$reason = as.factor(student$reason)

##############################################################################################################
##############################################################################################################
# 12 guardian - student's guardian (nominal: "mother", "father" or "other")

unique(student$guardian)
```

```
## [1] "mother" "father" "other"
```

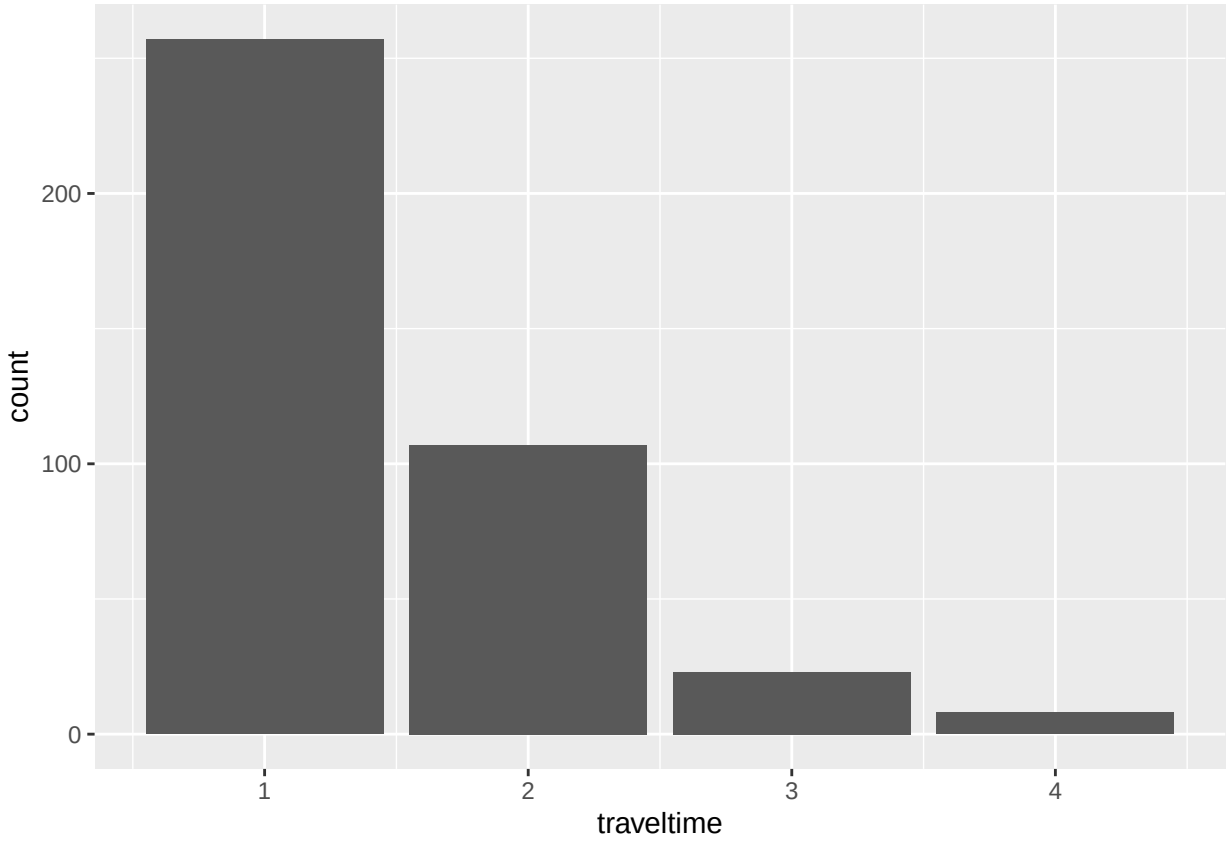
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=guardian, fill=guardian))
```




```
# ggsave("display.12.guardian.png")
student$guardian = as.factor(student$guardian)

##############################################################################################################
##############################################################################################################
# 13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to

unique(student$traveltime)
```

```
## [1] 2 1 3 4
```

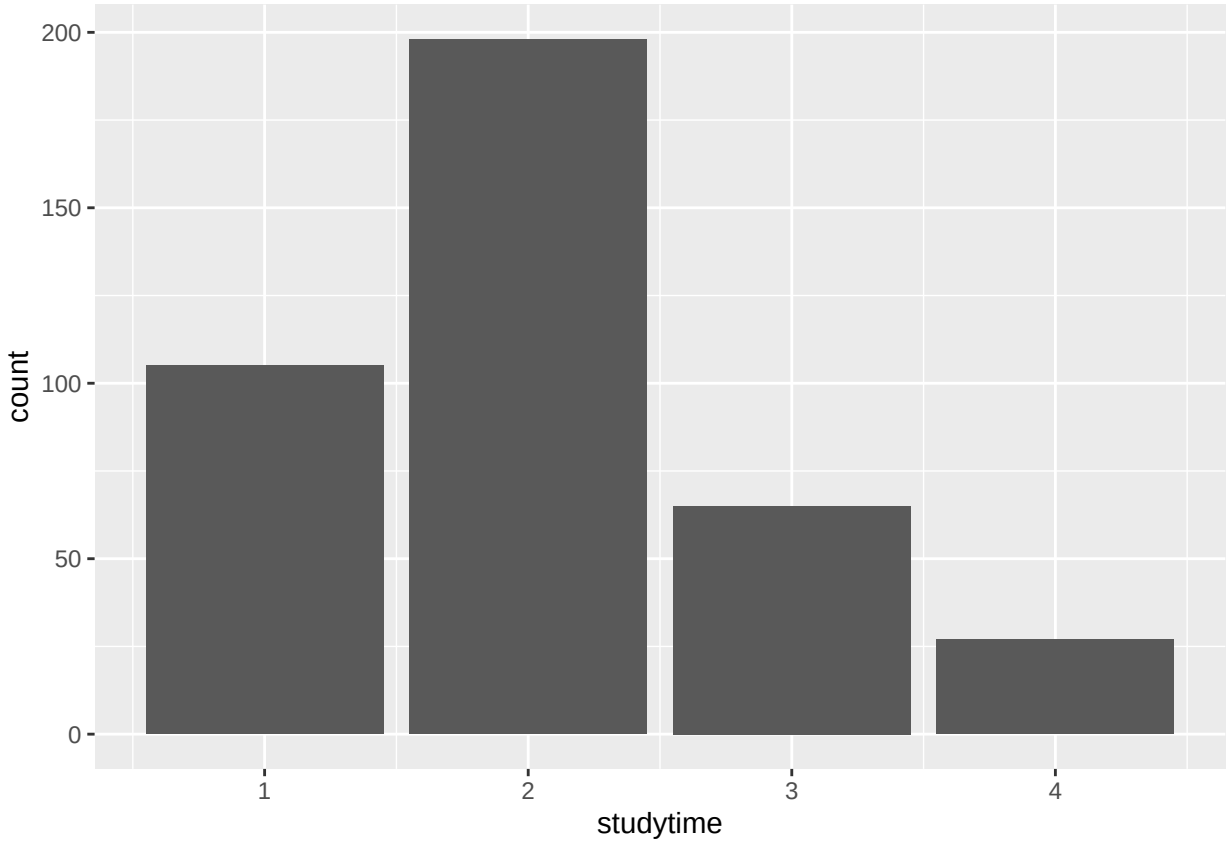
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=traveltime, fill=traveltime))
```




```
# ggsave("display.13.traveltime.png")
# we may wanna use the NUMERICAL VALUES :
student$traveltime = as.integer(student$traveltime)

##########################################################################################################
##########################################################################################################
# 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - 

unique(student$studytime)
```

```
## [1] 2 3 1 4
```

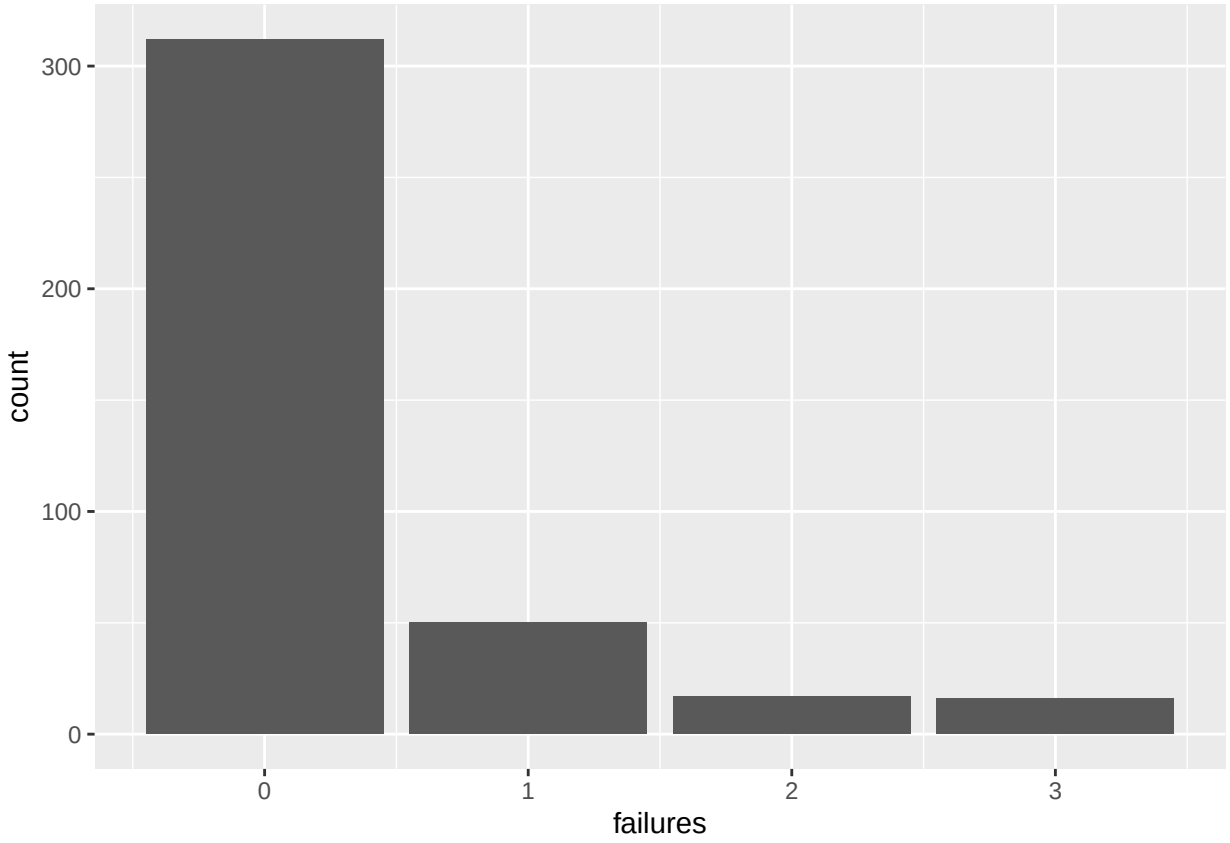
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=studytime, fill=studytime))
```




```
# ggsave("display.14.studytime.png")
# we may wanna use the NUMERICAL VALUES :
student$studytime = as.integer(student$studytime)

##########################################################################################################
##########################################################################################################
# 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)

unique(student$failures)
```

```
## [1] 0 3 2 1
```

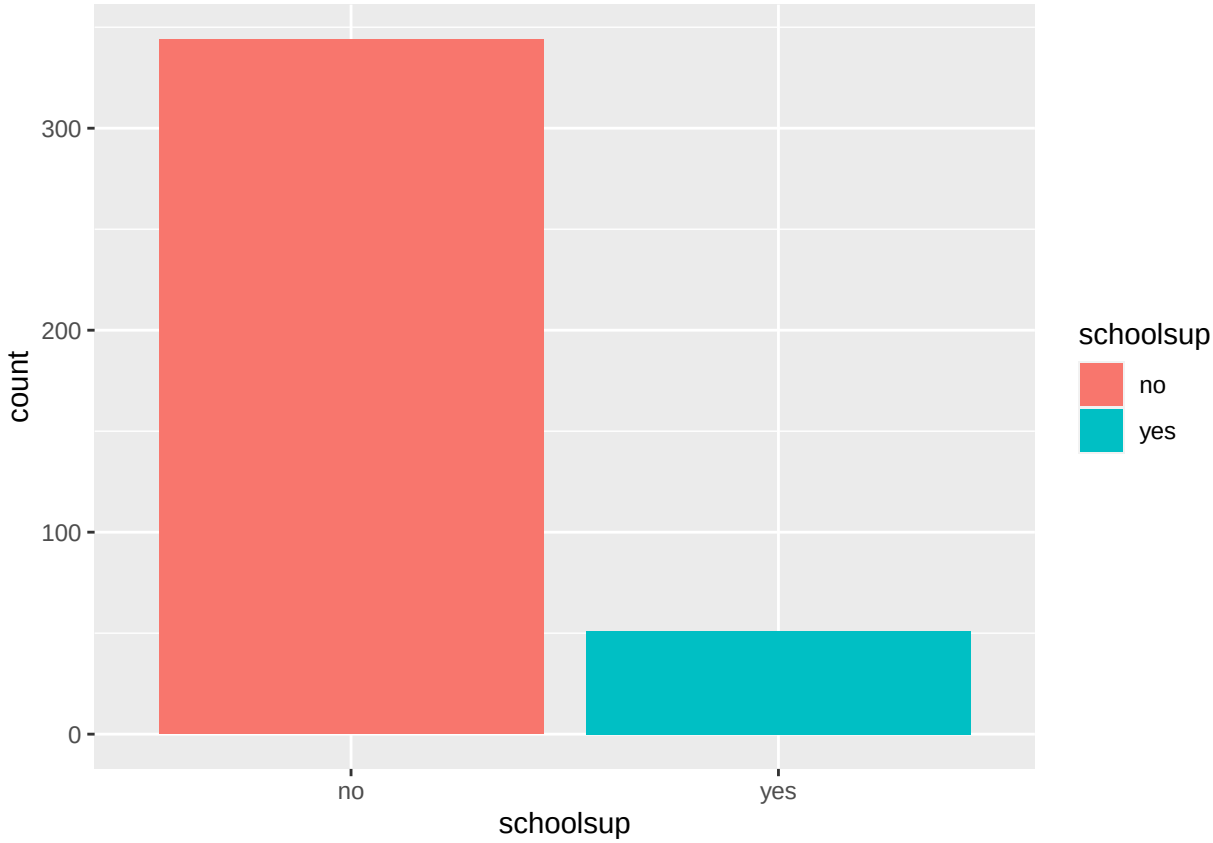
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=failures, fill=failures))
```




```
# ggsave("display.15.failures.png")
# we may wanna use the NUMERICAL VALUES :
student$failures = as.integer(student$failures)

##############################################################################################################
##############################################################################################################
# 16 schoolsup - extra educational support (binary: yes or no)

unique(student$schoolsup)
```

```
## [1] "yes" "no"
```

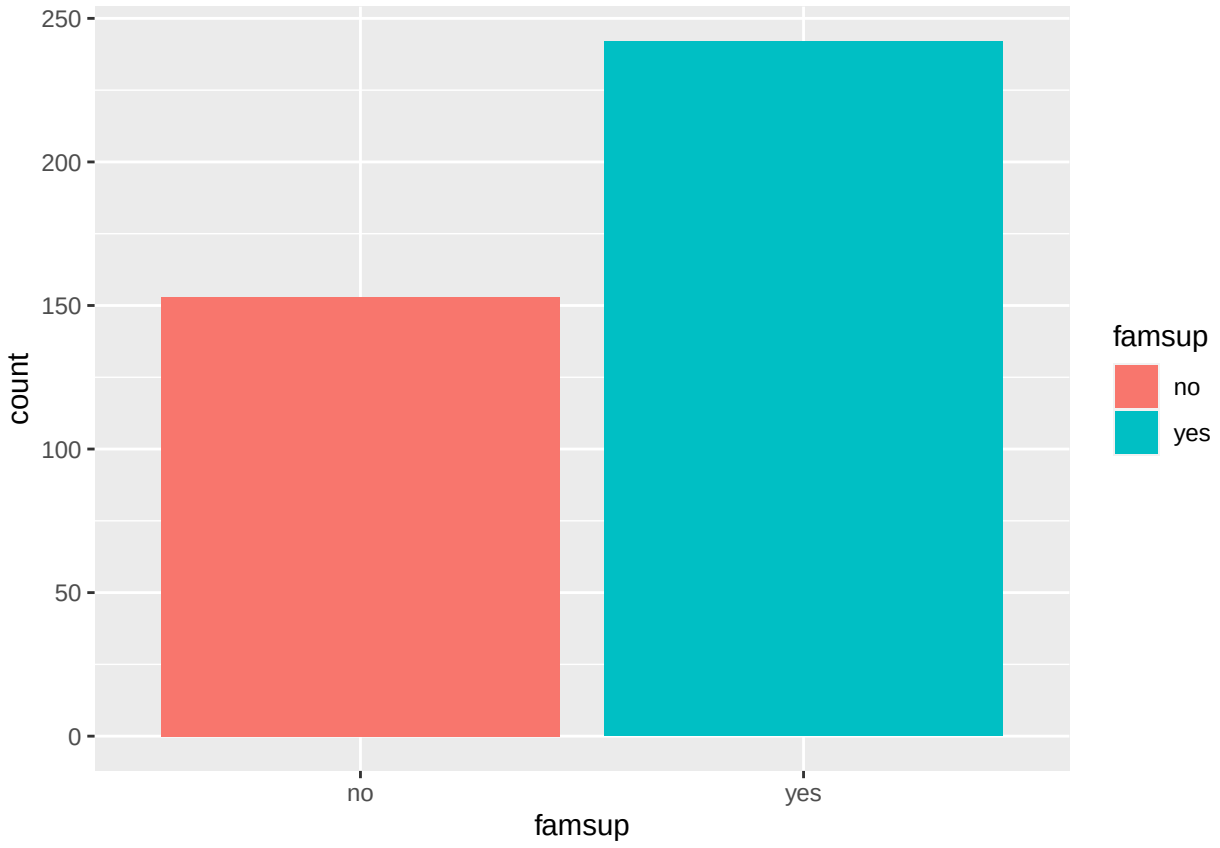
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=schoolsup, fill=schoolsup))
```




```
# ggsave("display.16.schoolsup.png")
student$schoolsup = as.factor(student$schoolsup)

##############################################################################################################
##############################################################################################################
# 17 famsup - family educational support (binary: yes or no)

unique(student$famsup)
```

```
## [1] "no"  "yes"
```

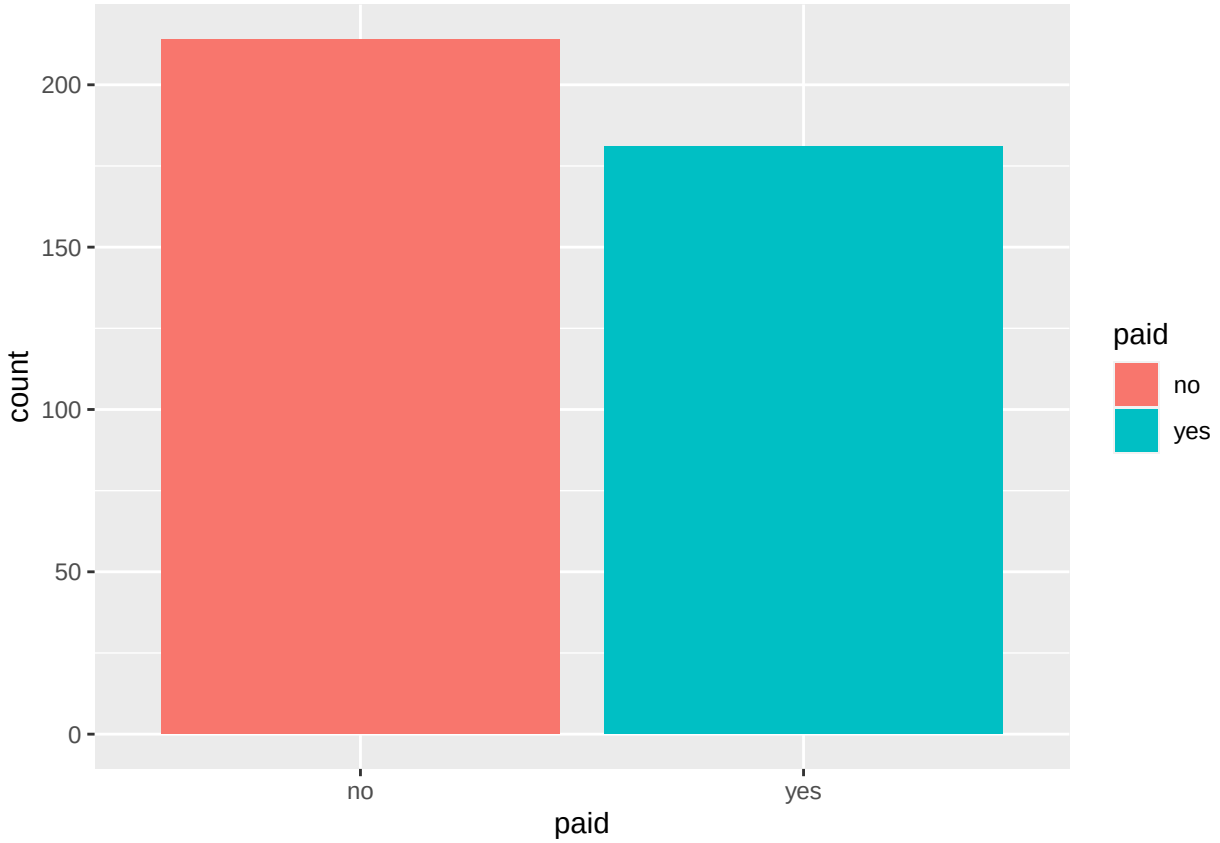
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=famsup, fill=famsup))
```




```
# ggsave("display.17.famsup.png")
student$famsup = as.factor(student$famsup)

##############################################################################################################
##############################################################################################################
# 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)

unique(student$paid)
```

```
## [1] "no"  "yes"
```

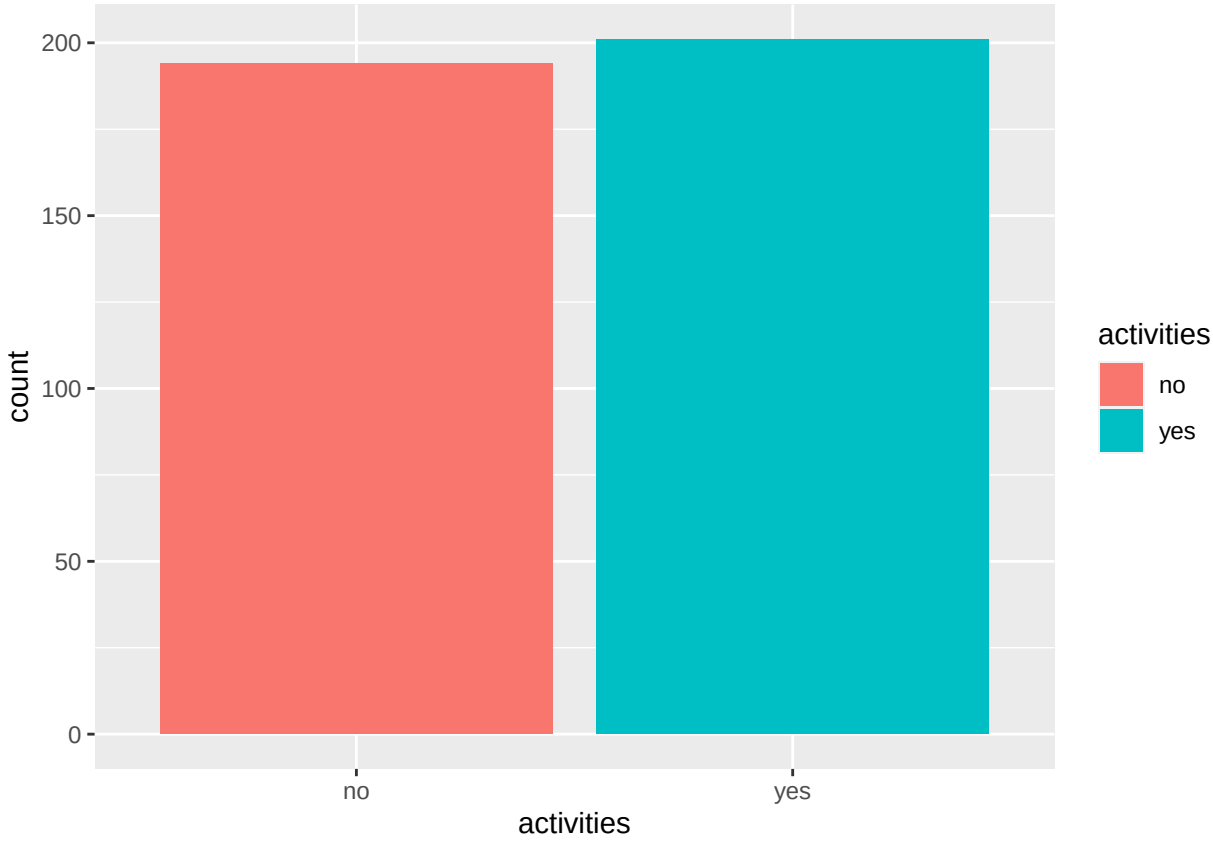
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=paid, fill=paid))
```




```
# ggsave("display.18.paid.png")
student$paid = as.factor(student$paid)

##############################################################################################################
##############################################################################################################
# 19 activities - extra-curricular activities (binary: yes or no)

unique(student$activities)
```

```
## [1] "no"  "yes"
```

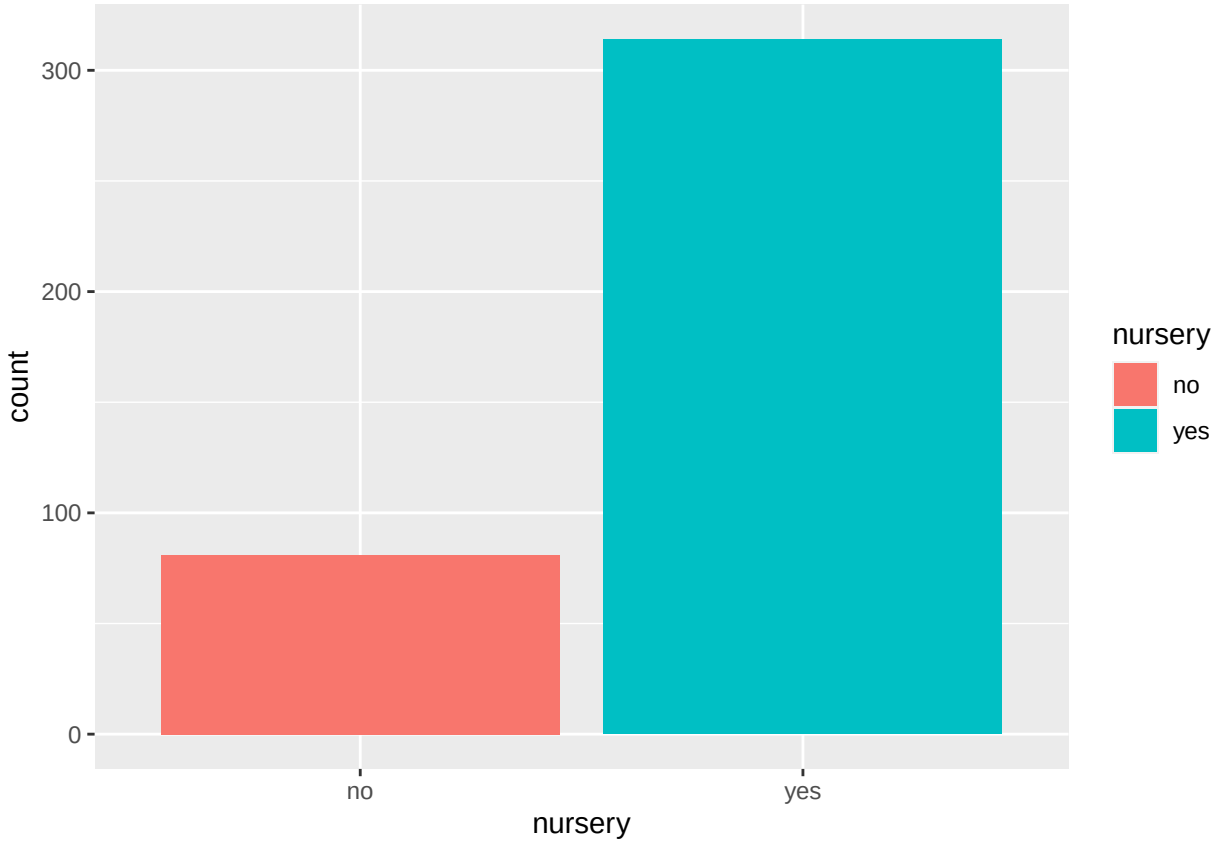
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=activities, fill=activities))
```




```
# ggsave("display.19.activities.png")
student$activities = as.factor(student$activities)

##############################################################################################################
##############################################################################################################
# 20 nursery - attended nursery school (binary: yes or no)

unique(student$nursery)
```

```
## [1] "yes" "no"
```

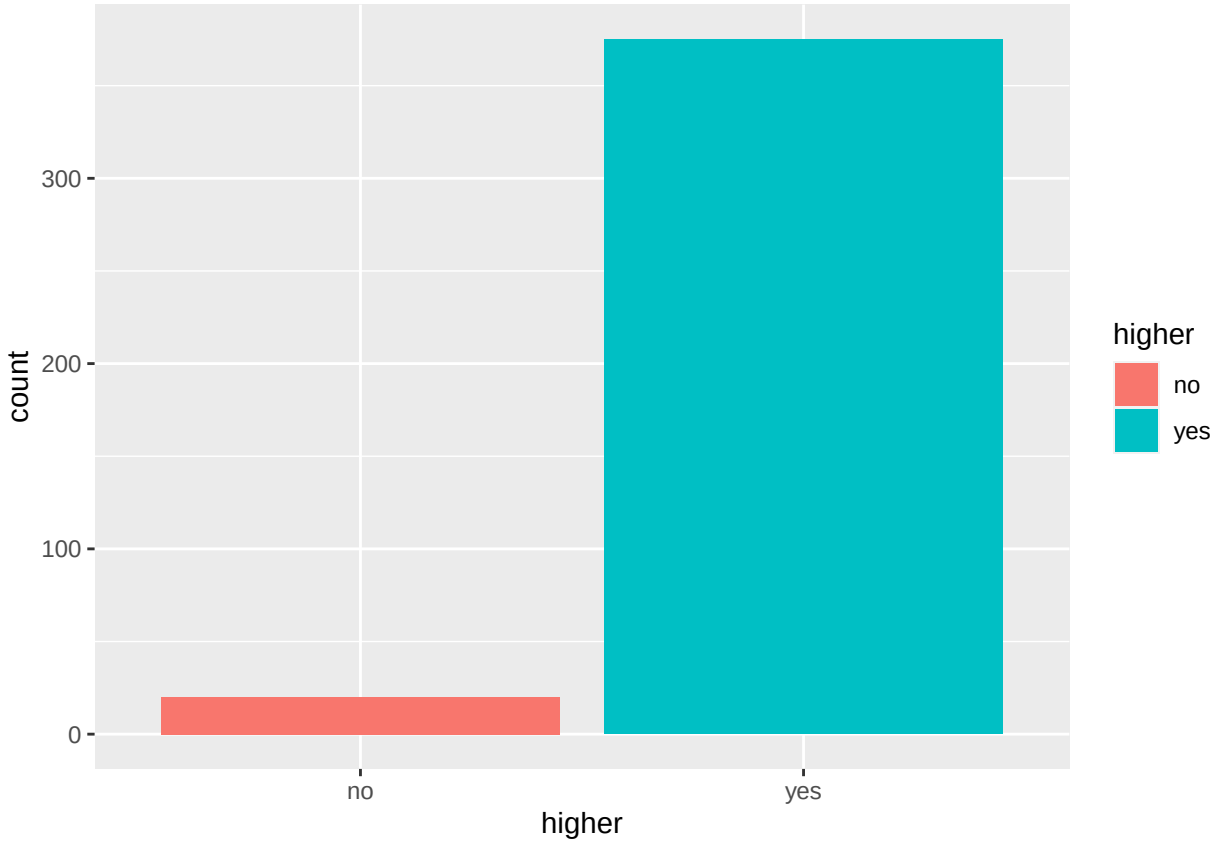
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=nursery, fill=nursery))
```




```
# ggsave("display.20.nursery.png")
student$nursery = as.factor(student$nursery)

##############################################################################################################
##############################################################################################################
# 21 higher - wants to take higher education (binary: yes or no)

unique(student$higher)
```

```
## [1] "yes" "no"
```

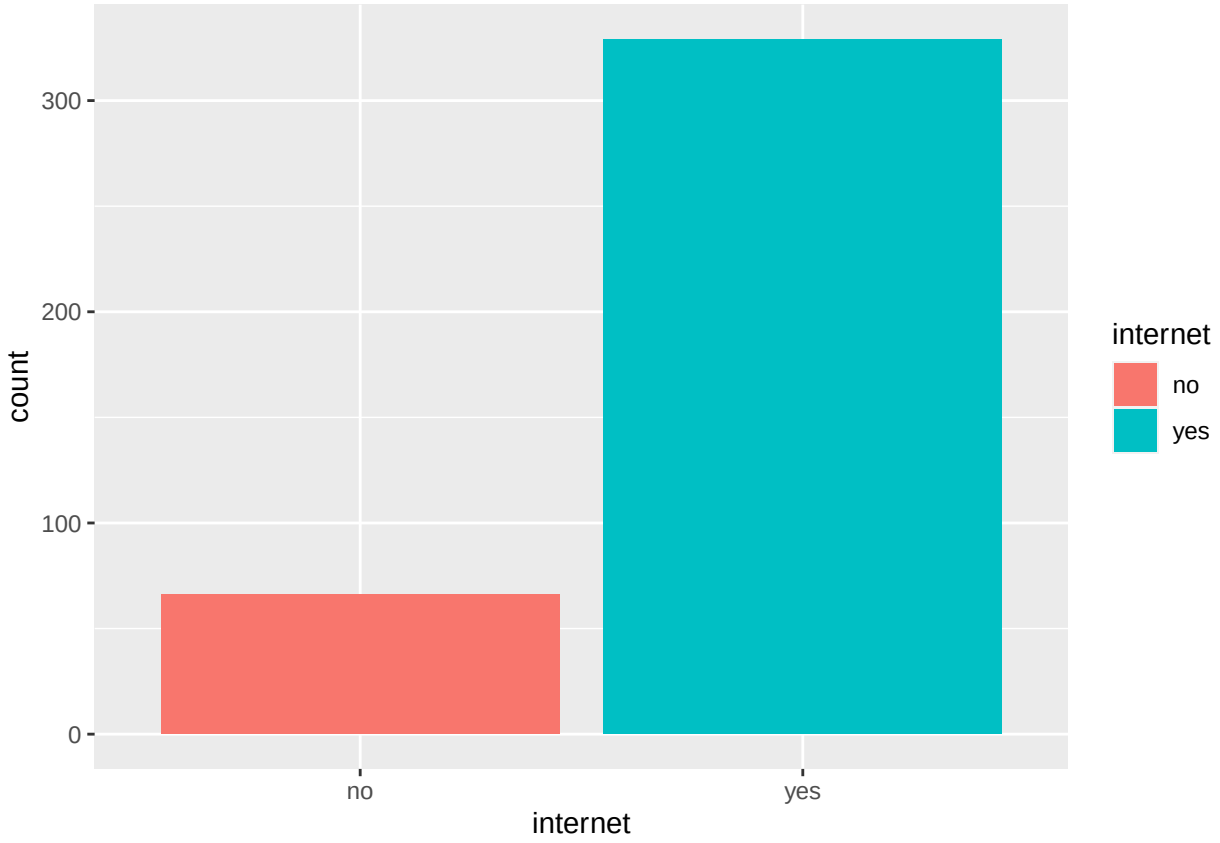
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=higher, fill=higher))
```




```
# ggsave("display.21.higher.png")
student$higher = as.factor(student$higher)

##############################################################################################################
##############################################################################################################
# 22 internet - Internet access at home (binary: yes or no)

unique(student$internet)
```

```
## [1] "no"  "yes"
```

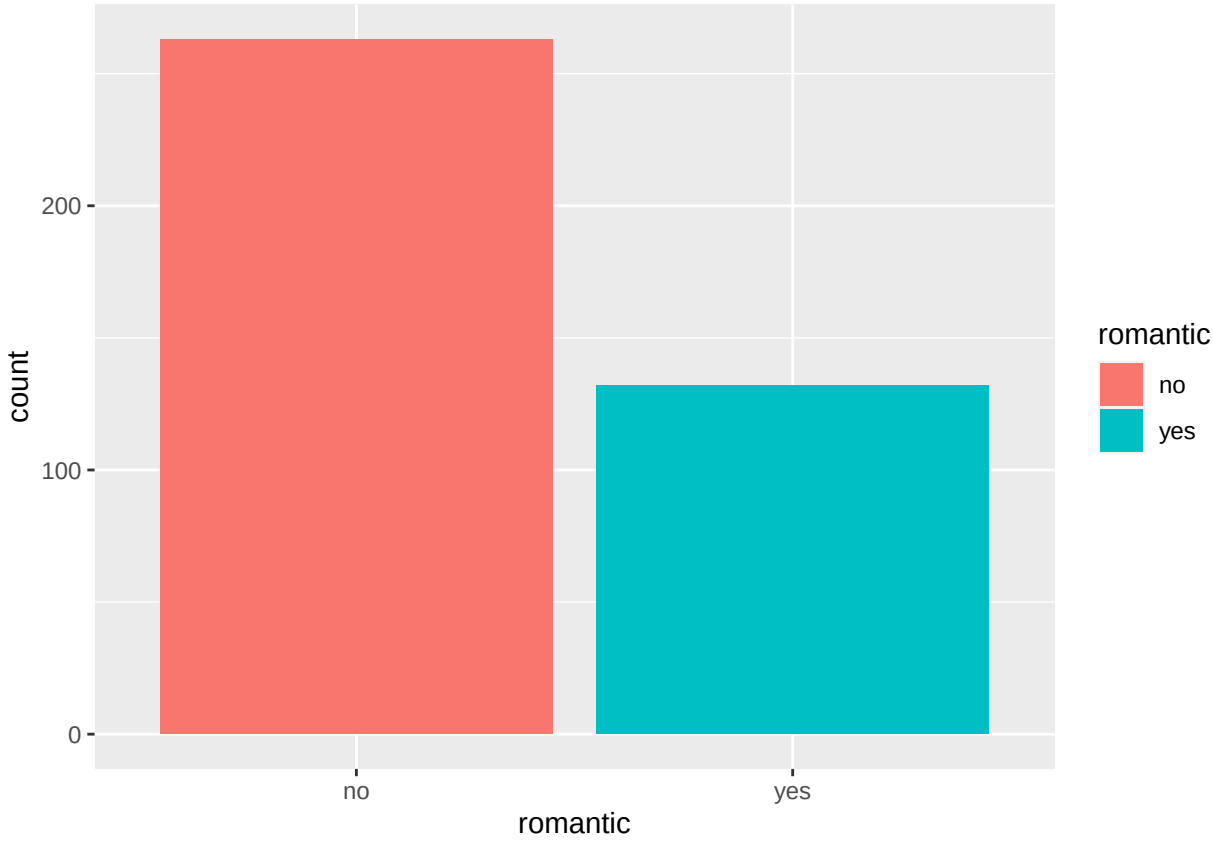
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=internet, fill=internet))
```




```
# ggsave("display.22.internet.png")
student$internet = as.factor(student$internet)

##############################################################################################################
##############################################################################################################
# 23 romantic - with a romantic relationship (binary: yes or no)

unique(student$romantic)
```

```
## [1] "no"  "yes"
```

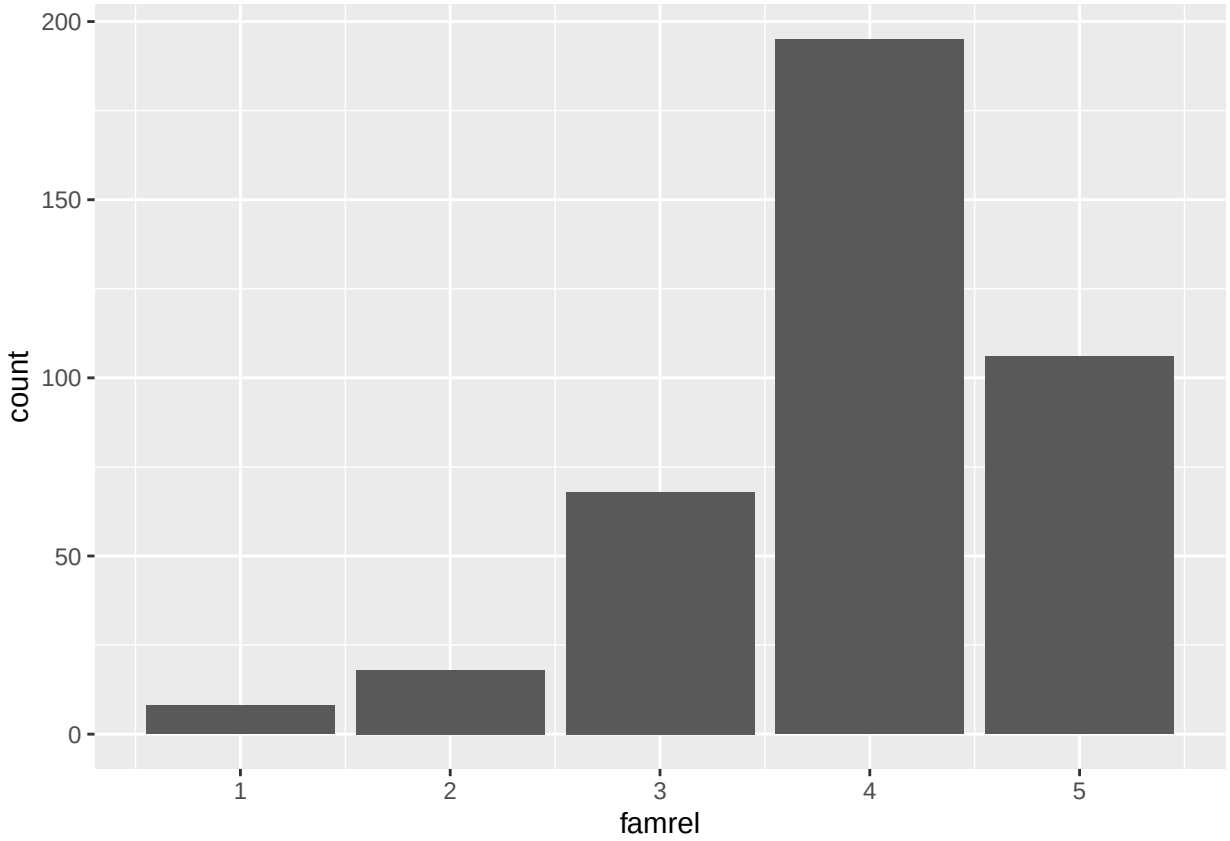
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=romantic, fill=romantic))
```




```
# ggsave("display.23.romantic.png")
student$romantic = as.factor(student$romantic)

##############################################################################################################
##############################################################################################################
# 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

unique(student$famrel)
```

```
## [1] 4 5 3 1 2
```

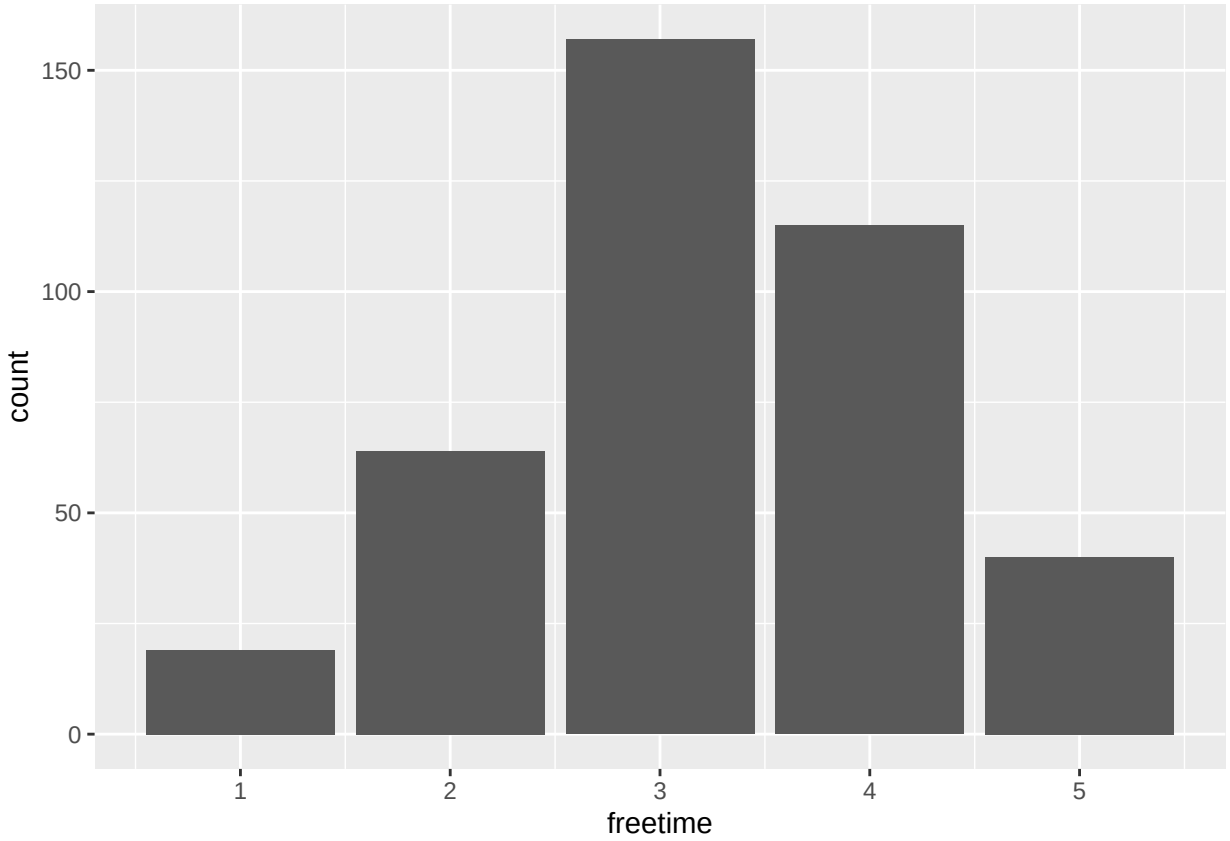
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=famrel, fill=famrel))
```




```
# ggsave("display.24.famrel.png")
# i believe that we can keep these as numerical :
student$famrel = as.integer(student$famrel)

##########################################################################################################
##########################################################################################################
# 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)

unique(student$freetime)
```

```
## [1] 3 2 4 1 5
```

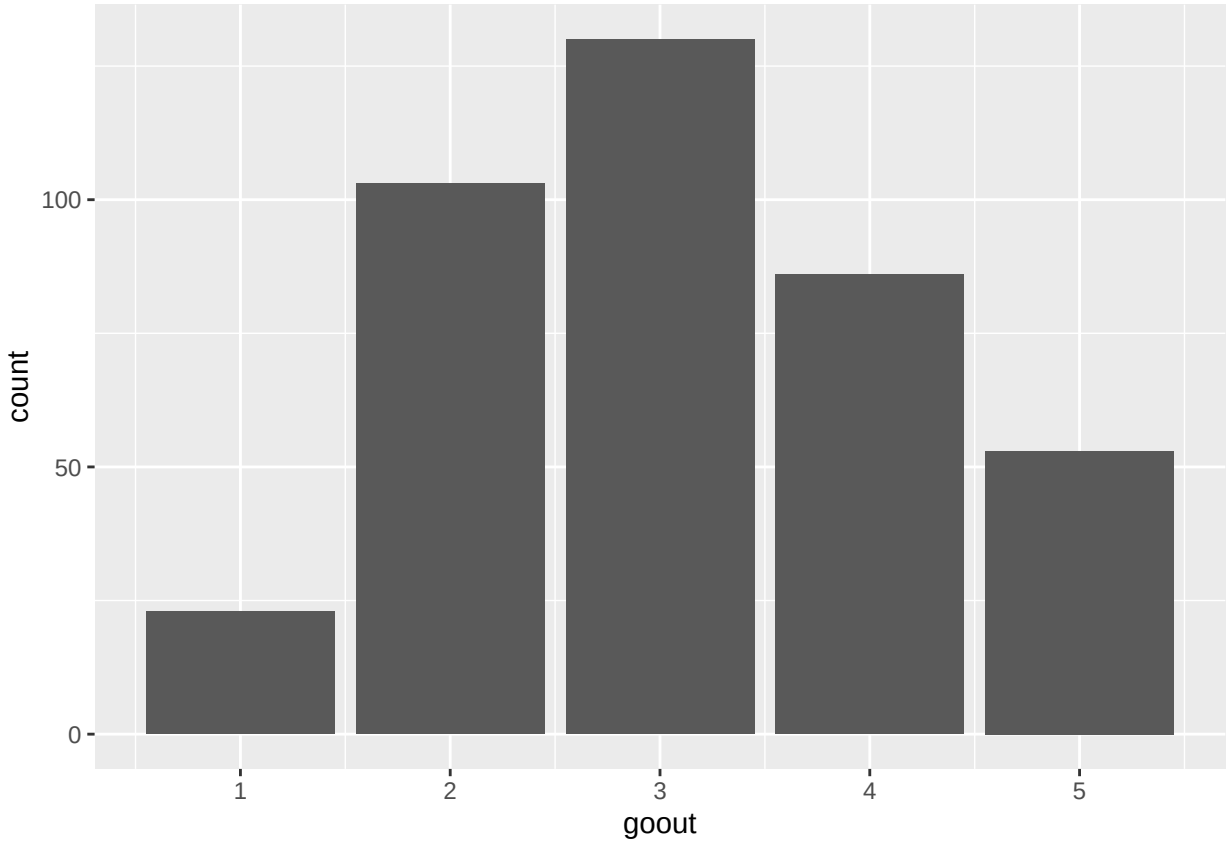
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=freetime, fill=freetime))
```




```
# ggsave("display.25.freetime.png")
# i believe that we can keep these as numerical :
student$freetime = as.integer(student$freetime)

##########################################################################################################
##########################################################################################################
# 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)

unique(student$goout)
```

```
## [1] 4 3 2 1 5
```

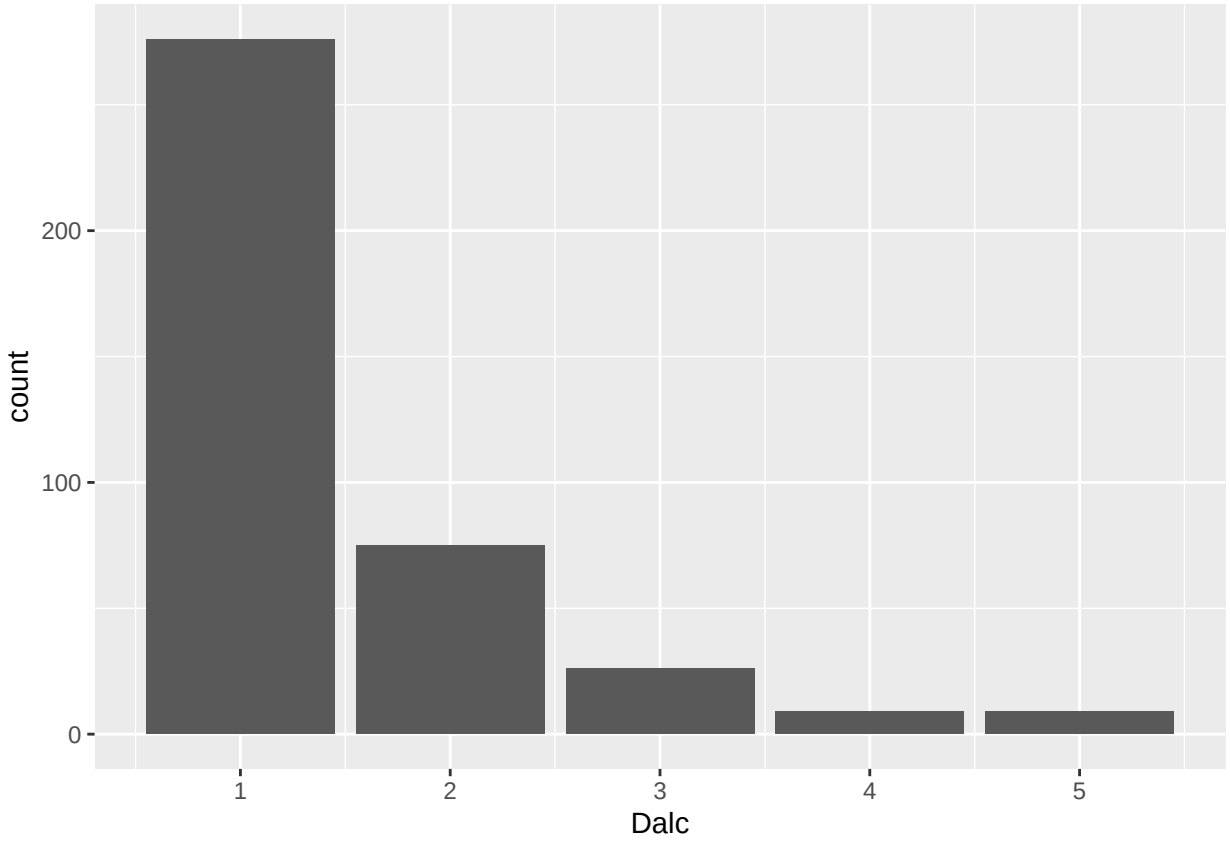
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=goout, fill=goout))
```




```
# ggsave("display.26.goout.png")
# i believe that we can keep these as numerical :
student$goout = as.integer(student$goout)

##########################################################################################################
##########################################################################################################
# 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)

unique(student$Dalc)
```

```
## [1] 1 2 5 3 4
```

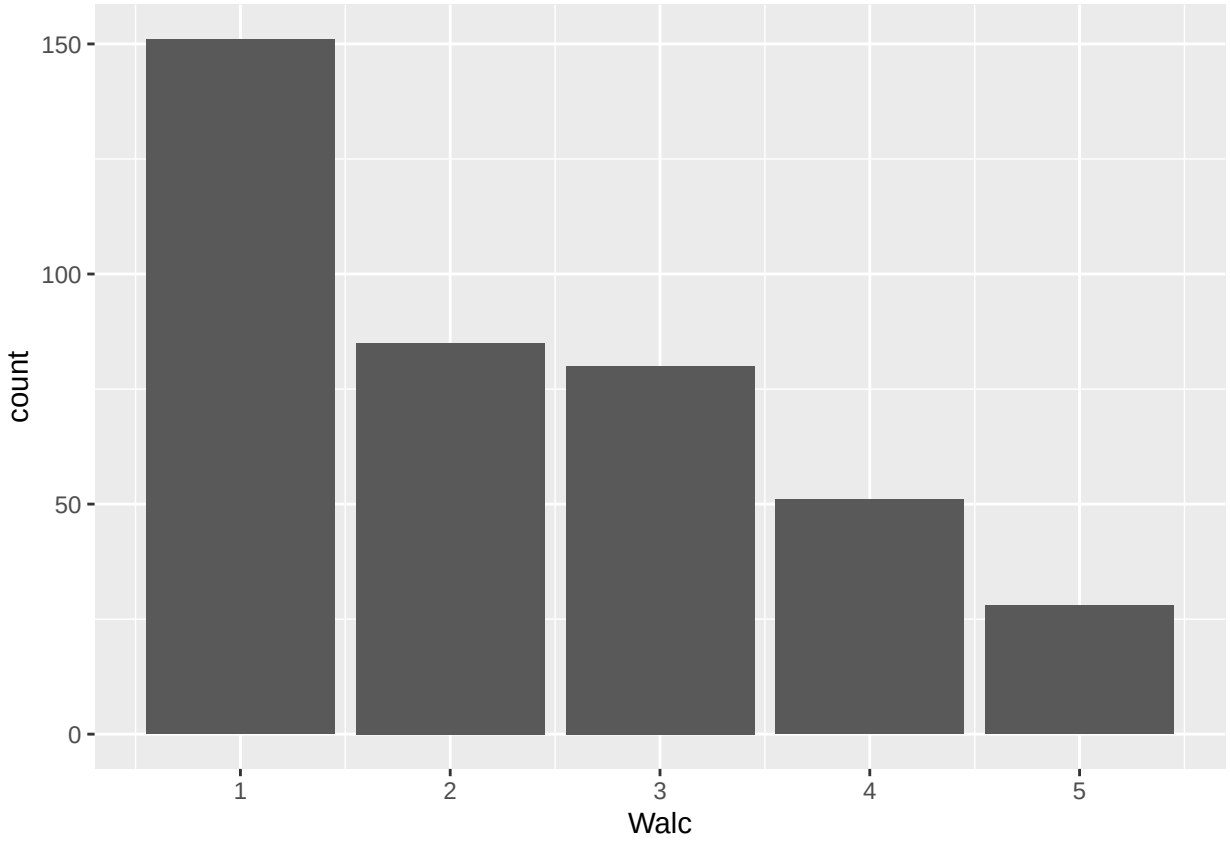
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Dalc, fill=Dalc))
```




```
# ggsave("display.27.Dalc.png")
# i believe that we can keep these as numerical :
student$Dalc = as.integer(student$Dalc)

##########################################################################################################
##########################################################################################################
# 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)

unique(student$Walc)
```

```
## [1] 1 3 2 4 5
```

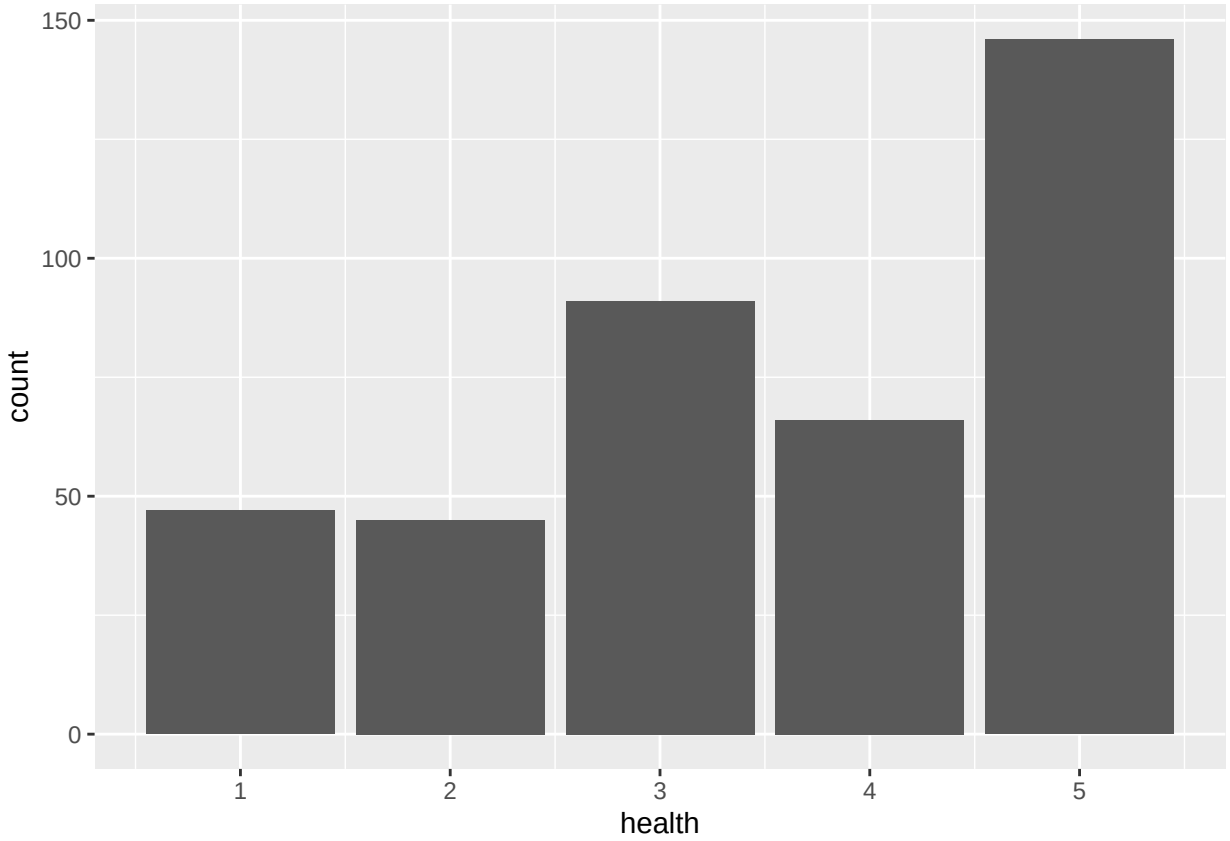
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=Walc, fill=Walc))
```




```
# ggsave("display.28.Walc.png")
# i believe that we can keep these as numerical :
student$Walc = as.integer(student$Walc)

##############################################################################################################
##############################################################################################################
# 29 health - current health status (numeric: from 1 - very bad to 5 - very good)

unique(student$health)
```

```
## [1] 3 5 1 2 4
```

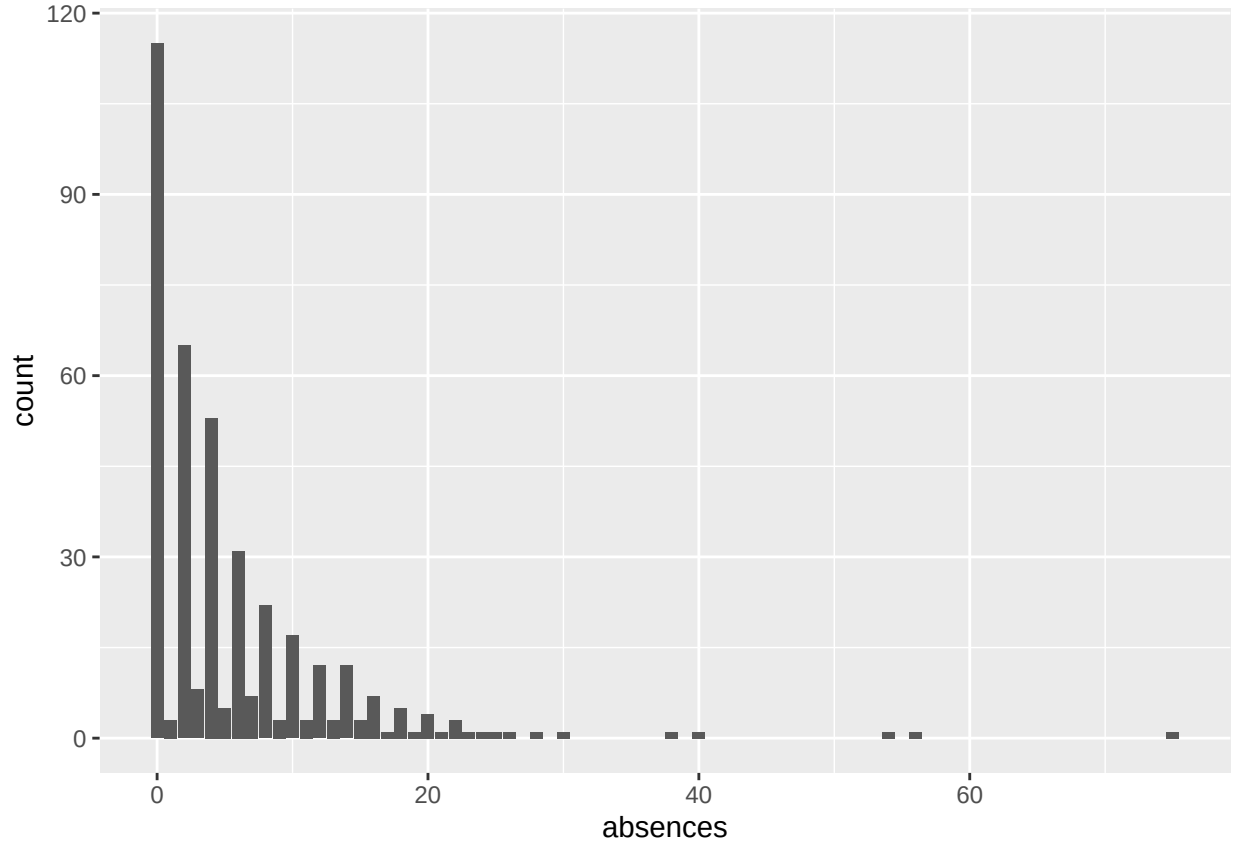
```
ggplot(data = student) +
       geom_bar(mapping = aes(x=health, fill=health))
```




```
# ggsave("display.29.health.png")
# i believe that we can keep these as numerical :
student$health = as.integer(student$health)

##############################################################################################################
##############################################################################################################
# 30 absences - number of school absences (numeric: from 0 to 93)

unique(student$absences)
```

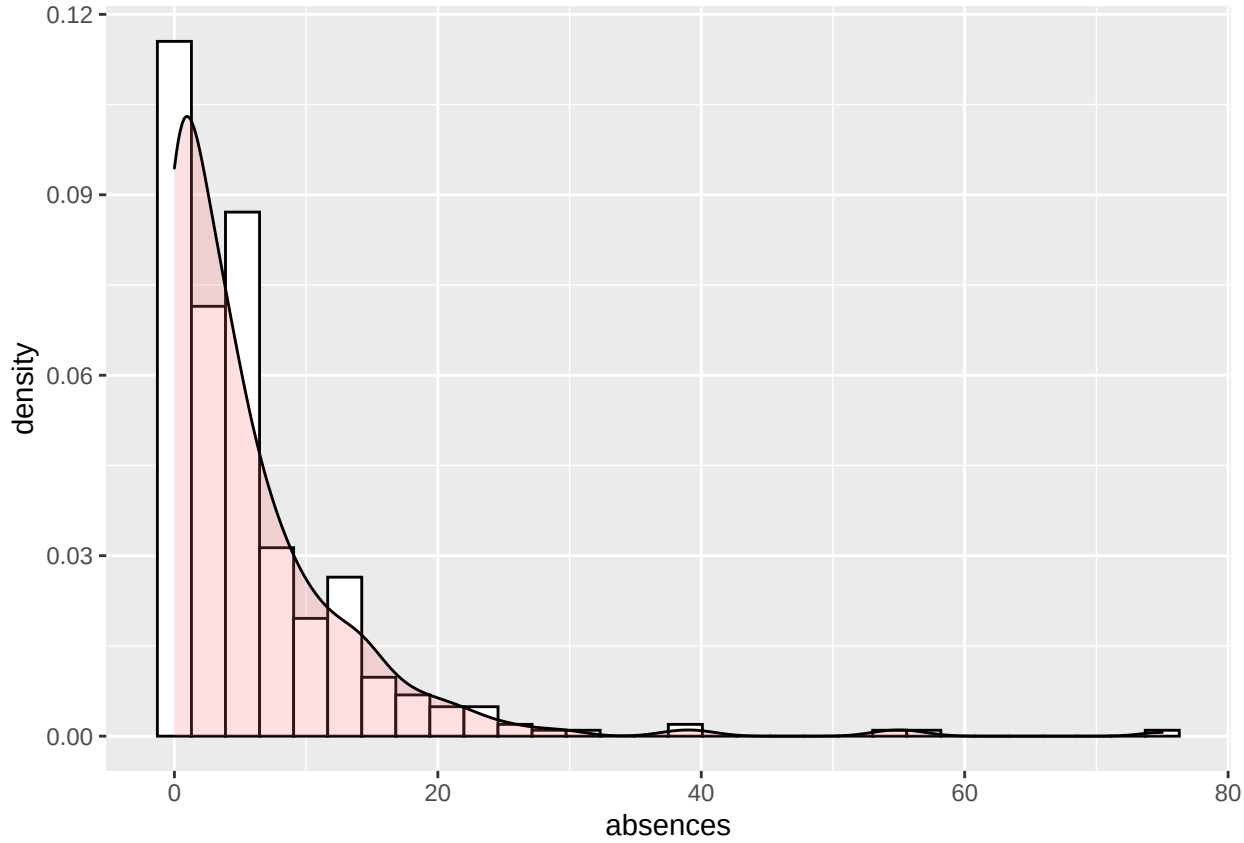
```
##  [1]  6  4 10  2  0 16 14  7  8 25 12 54 18 26 20 56 24 28  5 13 15 22  3 21  1
## [26] 75 30 19  9 11 38 40 23 17
```

```
ggplot(data = student) +
       geom_bar(mapping = aes(x=absences, fill=absences))
```




```
ggplot(data=student, aes(x=absences)) +
      geom_histogram(aes(y=..density..), colour="black", fill="white")+
      geom_density(alpha=.2, fill="#FF6666")
```

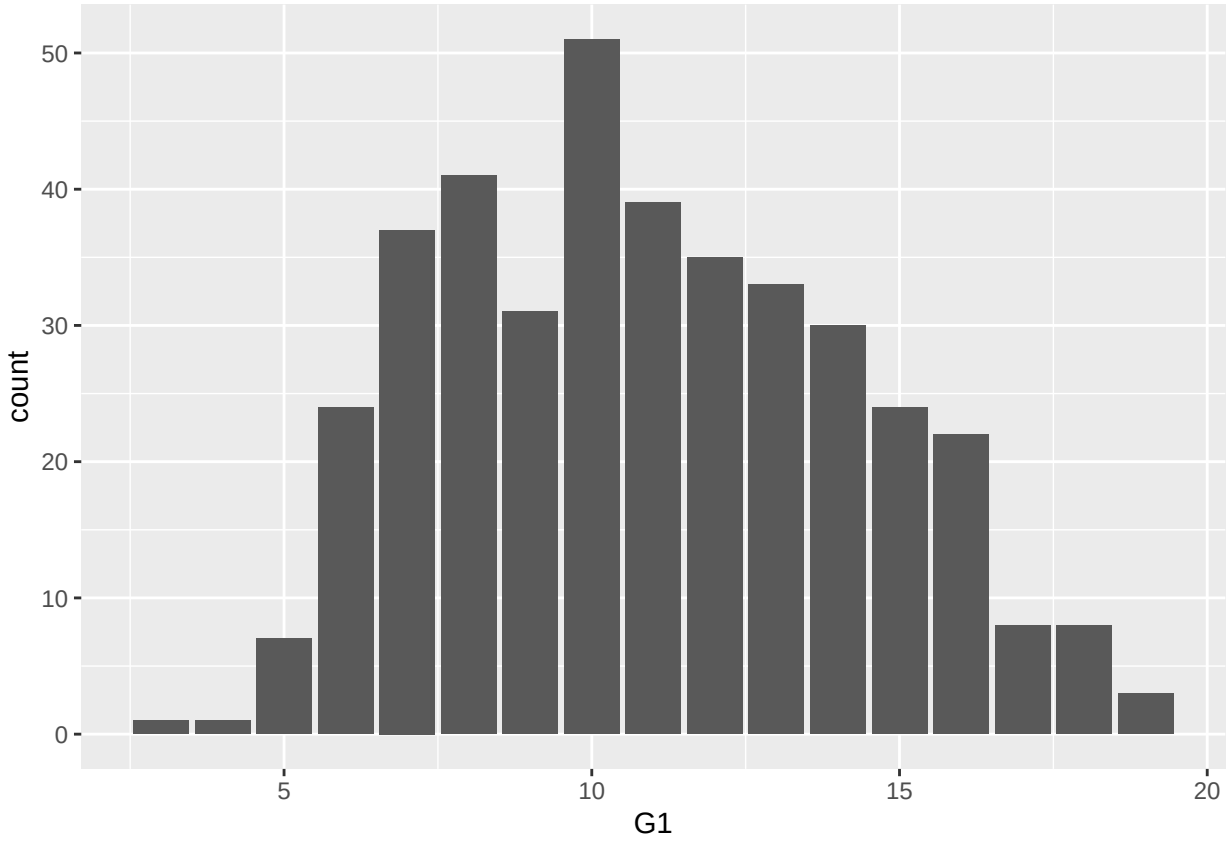
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggsave("display.30.absences.png")
# i believe that we can keep these as numerical :
student$absences = as.integer(student$absences)

##############################################################################################################
##############################################################################################################
# $ G1         : int  5 5 7 15 6 15 12 6 16 14 ...

unique(student$G1)
```

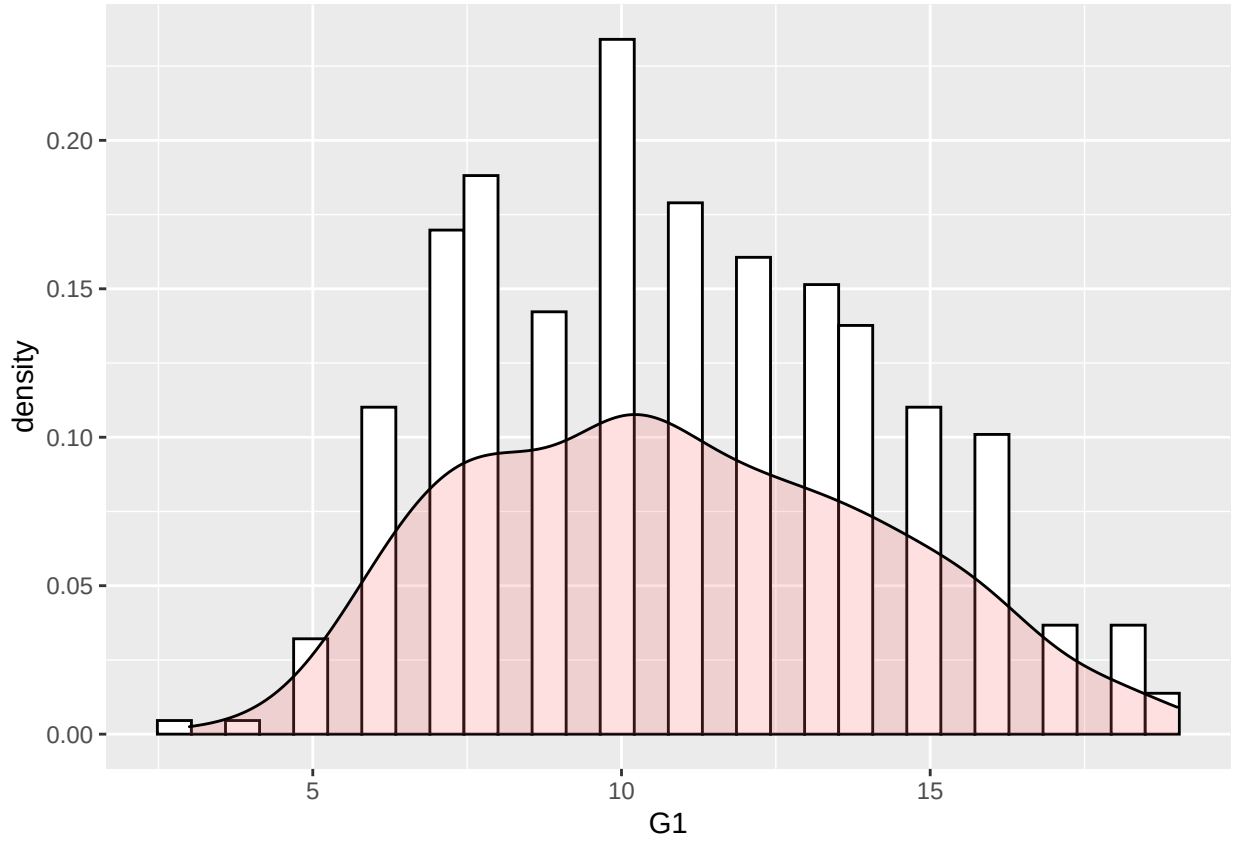
```
##  [1]  5  7 15  6 12 16 14 10 13  8 11  9 17 19 18  4  3
```

```
ggplot(data = student) +
       geom_bar(mapping = aes(x=G1, fill=G1))
```




```
ggplot(data=student, aes(x=G1)) +
      geom_histogram(aes(y=..density..), colour="black", fill="white")+
      geom_density(alpha=.2, fill="#FF6666")
```

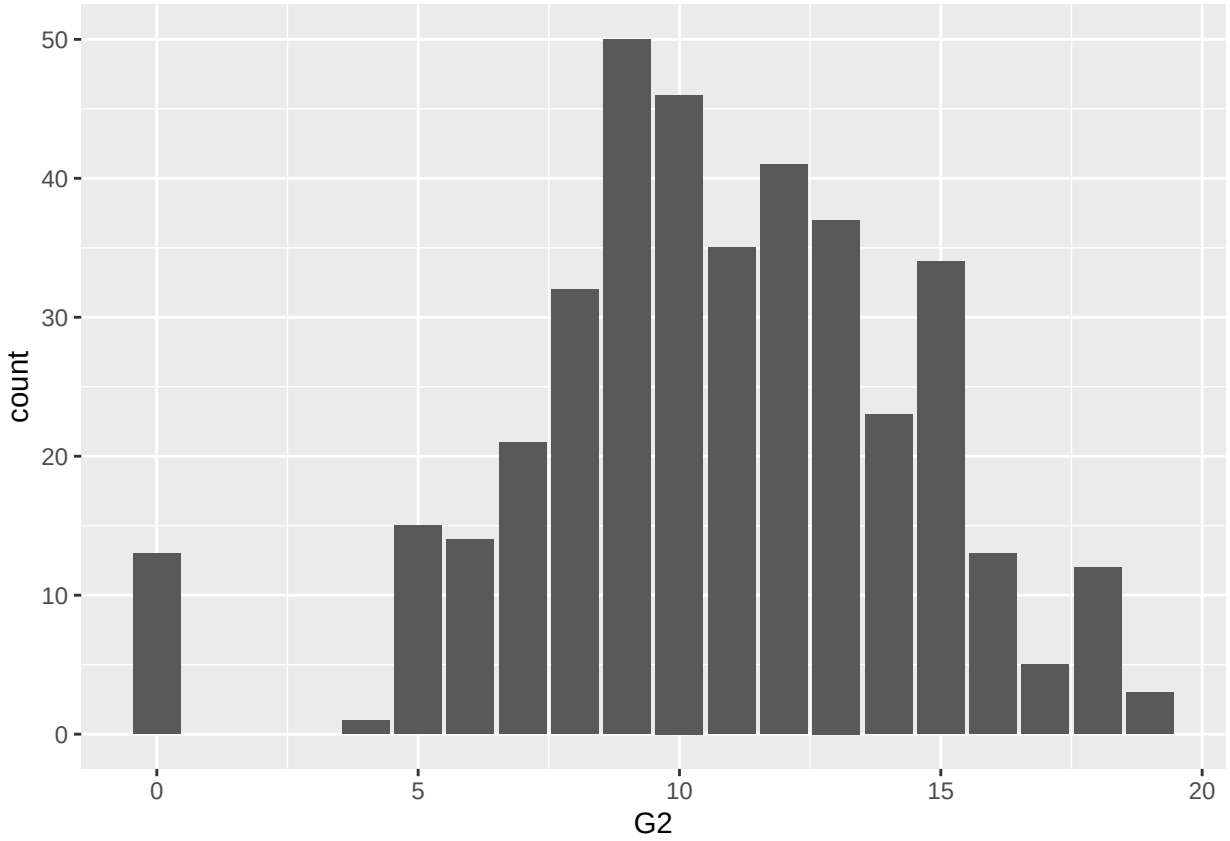
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggsave("display.0.G1.png")
# i believe that we can keep these as numerical, although we may not need it :
student$G1 = as.factor(student$G1)

##############################################################################################################
##############################################################################################################
# $ G2         : int  6 5 8 14 10 15 12 5 18 15 ...

unique(student$G2)
```

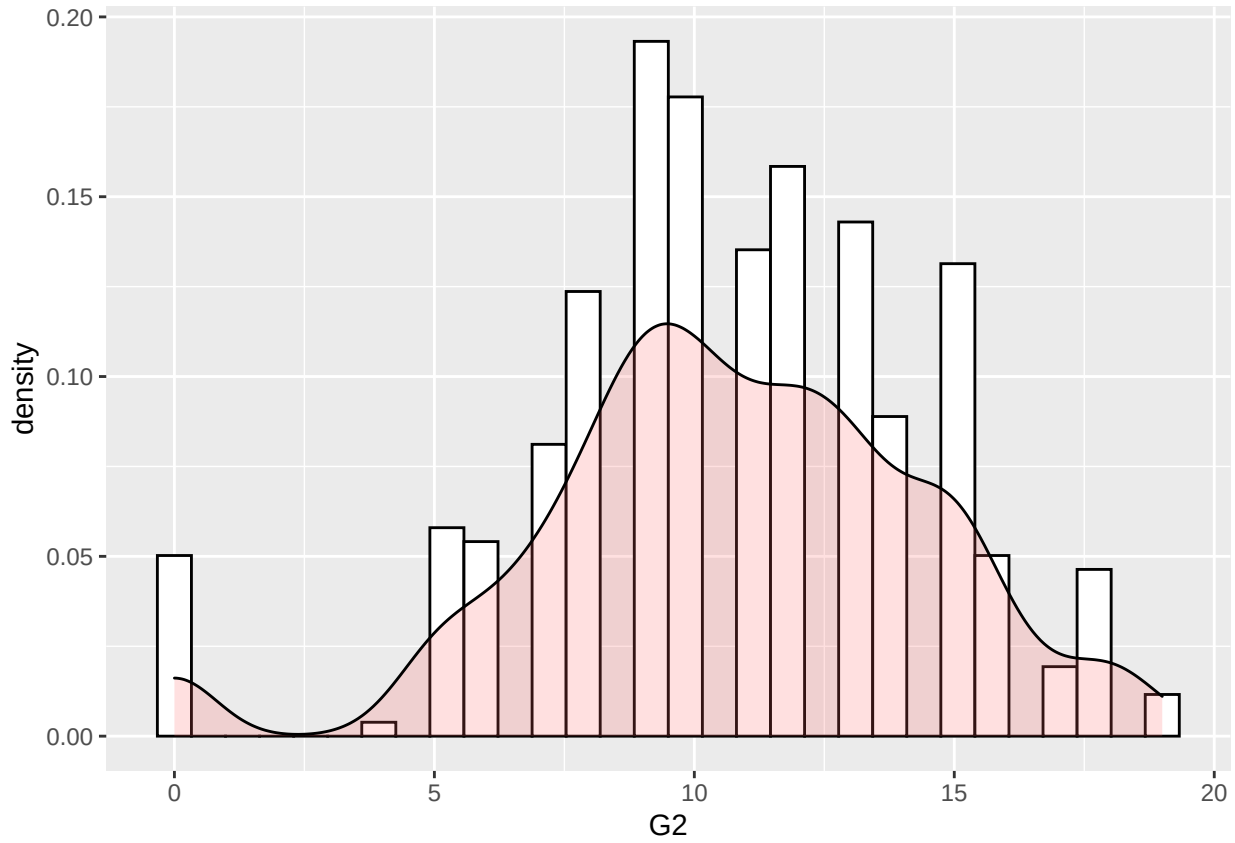
```
##  [1]  6  5  8 14 10 15 12 18 16 13  9 11  7 19 17  4  0
```

```
ggplot(data = student) +
       geom_bar(mapping = aes(x=G2, fill=G2))
```




```
ggplot(data=student, aes(x=G2)) +
      geom_histogram(aes(y=..density..), colour="black", fill="white")+
      geom_density(alpha=.2, fill="#FF6666")
```

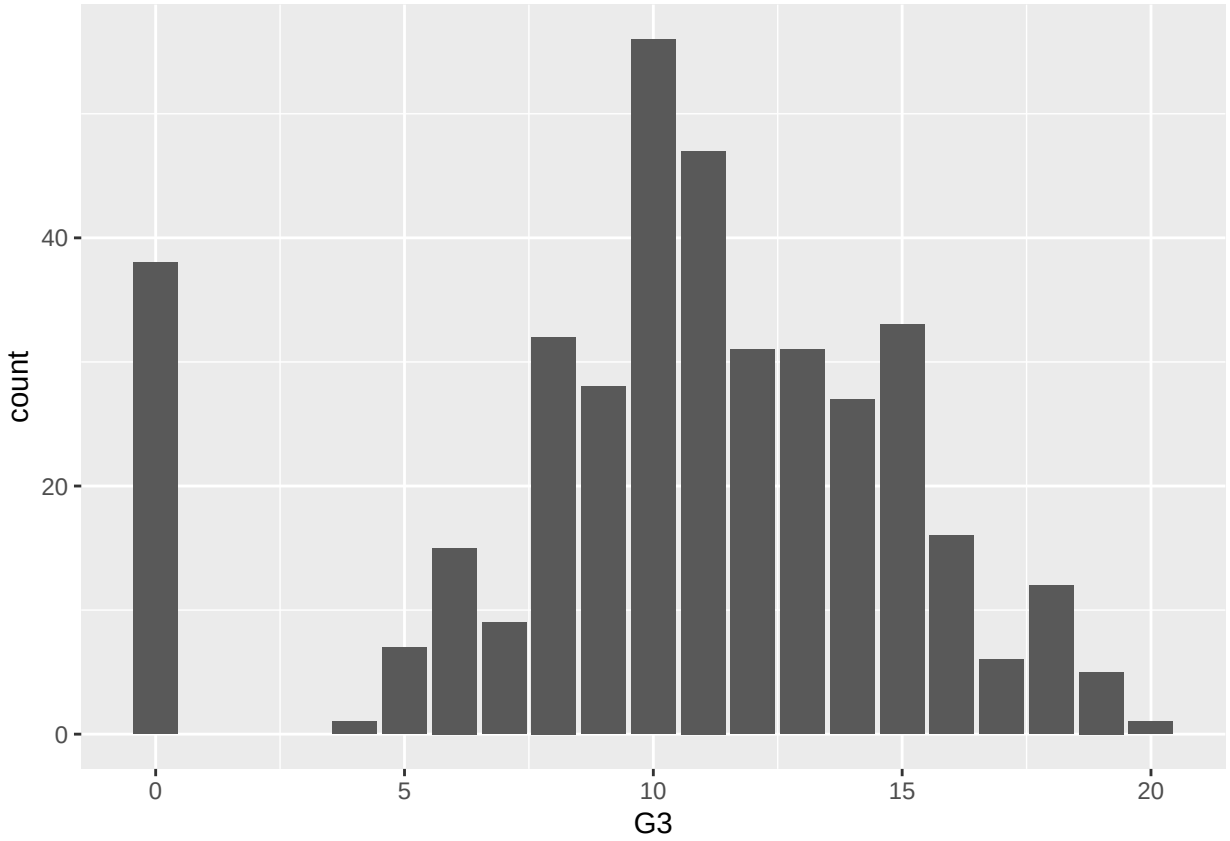
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggsave("display.0.G2.png")
# i believe that we can keep these as numerical, although we may not need it :
student$G2 = as.factor(student$G2)

##############################################################################################################
##############################################################################################################
# $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...

unique(student$G3)
```

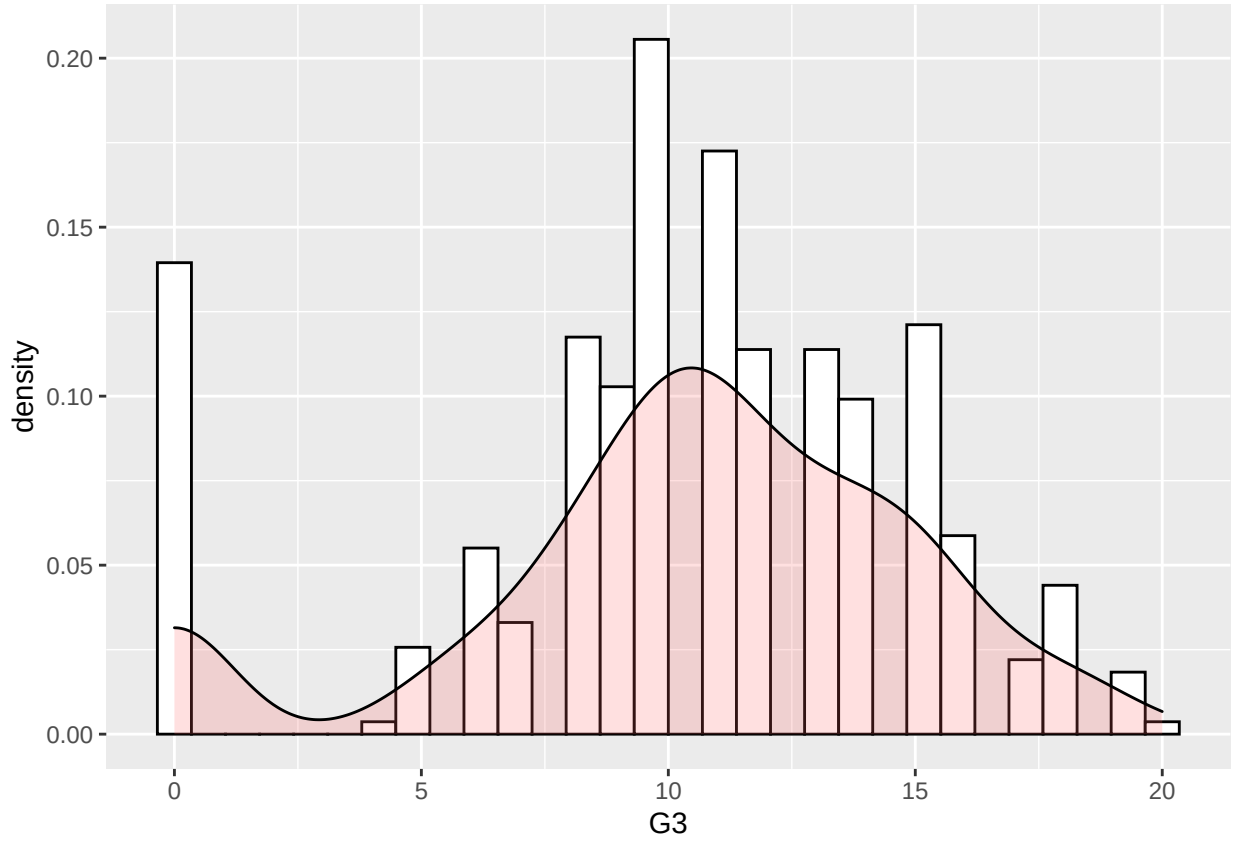
```
##  [1]  6 10 15 11 19  9 12 14 16  5  8 17 18 13 20  7  0  4
```

```
ggplot(data = student) +
       geom_bar(mapping = aes(x=G3, fill=G3))
```




```
ggplot(data=student, aes(x=G3)) +
      geom_histogram(aes(y=..density..), colour="black", fill="white")+
      geom_density(alpha=.2, fill="#FF6666")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggsave("display.0.G3.png")
# i believe that we can covert it into RANGES of VALUES :
student$G3 = as.factor(student$G3)

##################################################################################################################
##################################################################################################################
##################################################################################################################
##################################################################################################################
##################################################################################################################
##################################################################################################################

summary(student)
```

```
##  school   sex          age       address famsize   Pstatus      Medu
##  GP:349   F:208   Min.   :15.0   R: 88   GT3:281   A: 41   Min.   :0.000
##  MS: 46   M:187   1st Qu.:16.0   U:307   LE3:114   T:354   1st Qu.:2.000
##                   Median :17.0                           Median :3.000
##                   Mean   :16.7                           Mean   :2.749
##                   3rd Qu.:18.0                           3rd Qu.:4.000
##                   Max.   :22.0                           Max.   :4.000
##
##       Fedu             Mjob           Fjob            reason      guardian
##  Min.   :0.000   at_home : 59   at_home : 20   course    :145   father: 90
##  1st Qu.:2.000   health  : 34   health  : 18   home      :109   mother:273
##  Median :2.000   other   :141   other   :217   other     : 36   other : 32
##  Mean   :2.522   services:103   services:111   reputation:105
```

```
##  3rd Qu.:3.000   teacher : 58   teacher : 29
##  Max.   :4.000
##
##    traveltime      studytime        failures      schoolsup famsup     paid
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   no :344   no :153   no :214
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   yes: 51   yes:242   yes:181
##  Median :1.000   Median :2.000   Median :0.0000
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##
##  activities nursery   higher    internet  romantic      famrel
##  no :194    no : 81   no : 20   no : 66   no :263   Min.   :1.000
##  yes:201    yes:314   yes:375   yes:329   yes:132   1st Qu.:4.000
##                                                     Median :4.000
##                                                     Mean   :3.944
##                                                     3rd Qu.:5.000
##                                                     Max.   :5.000
##
##     freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      health         absences            G1            G2            G3
##  Min.   :1.000   Min.   : 0.000   10     : 51   9      : 50   10     : 56
##  1st Qu.:3.000   1st Qu.: 0.000   8      : 41   10     : 46   11     : 47
##  Median :4.000   Median : 4.000   11     : 39   12     : 41   0      : 38
##  Mean   :3.554   Mean   : 5.709   7      : 37   13     : 37   15     : 33
##  3rd Qu.:5.000   3rd Qu.: 8.000   12     : 35   11     : 35   8      : 32
##  Max.   :5.000   Max.   :75.000   13     : 33   15     : 34   12     : 31
##                                   (Other):159   (Other):152   (Other):158
```

```
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
```

```
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : Factor w/ 17 levels "3","4","5","6",..: 3 3 5 13 4 13 10 4 14 12 ...
##  $ G2        : Factor w/ 17 levels "0","4","5","6",..: 4 3 6 12 8 13 10 3 16 13 ...
##  $ G3        : Factor w/ 18 levels "0","4","5","6",..: 4 4 8 13 8 13 9 4 17 13 ...
```

```r
class(student)
```

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

3. DATA SELECTION

```
## the OUTPUT VARIABLES is G3
## we may remove G1 and G2
## and some of the other features that are not numerical

student1 <- subset(student, select = -c(G1, G2))

student2 <- subset(student1,
                   select = -c(school, sex, address, famsize, Pstatus,
                   Mjob, Fjob, reason, guardian, schoolsup, famsup, paid, activities, nursery,
                   higher, internet, romantic))

str(student2)
```

```
## 'data.frame':    395 obs. of  14 variables:
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G3        : Factor w/ 18 levels "0","4","5","6",..: 4 4 8 13 8 13 9 4 17 13 ...
```

```
student2$G3 = as.factor(student2$G3)

table(student2$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

```
### for simplicity, to work with a copy of STUDENT2, let's call it STUDENT3

student3 = subset(student2,
                  select= c(age, traveltime, studytime, failures, absences, G3))

table(student3$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

4. DATA FILTERING, SUMMARIES and VISUALIZATION

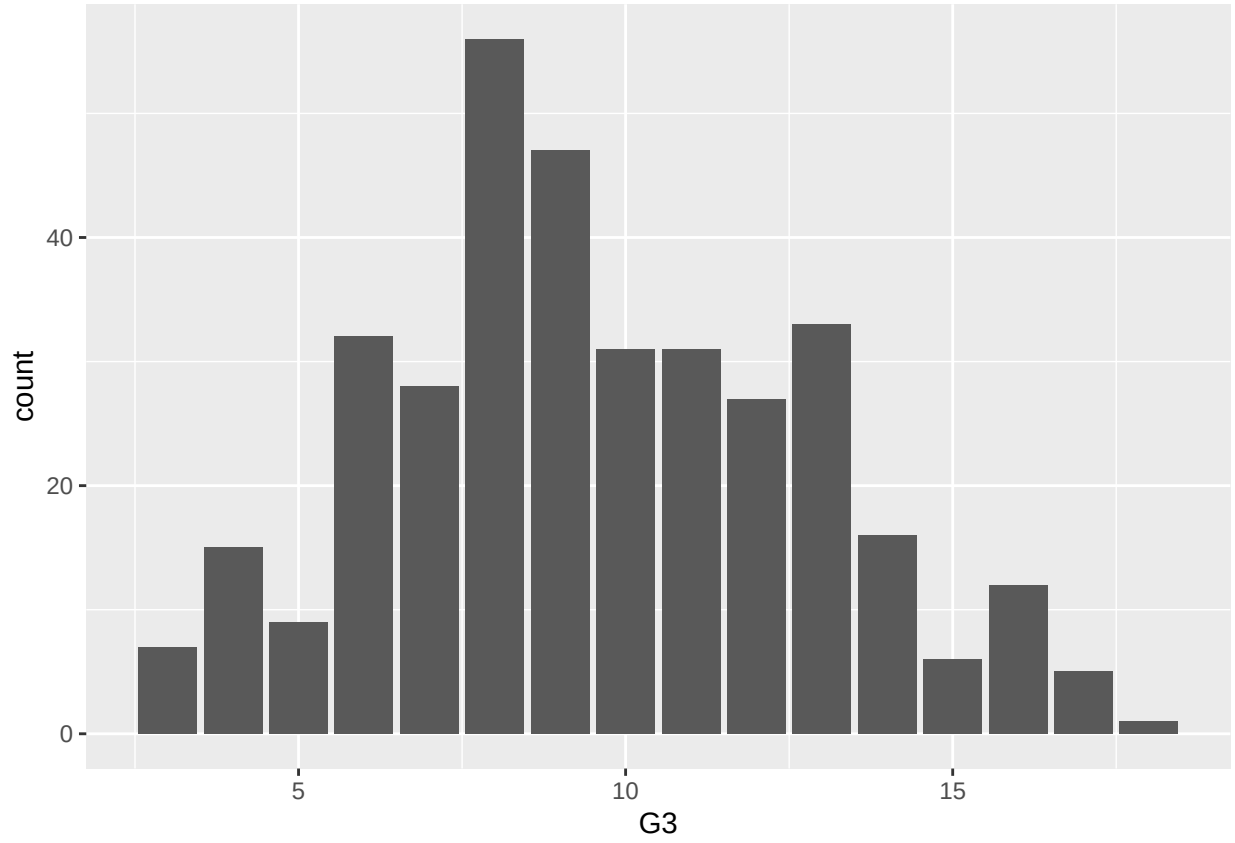
```
## in order to REMOVE the RECORDS where the GRADE 3 is > 2 :

dim(student3)
```

```
## [1] 395   6
```

```
student3$G3 = as.integer(student3$G3)

student4 = student3[student3$G3 > 2, ]

dim(student4)
```

```
## [1] 356   6
```

```
ggplot(data = student4) +
       geom_bar(mapping = aes(x=G3, fill=G3))
```

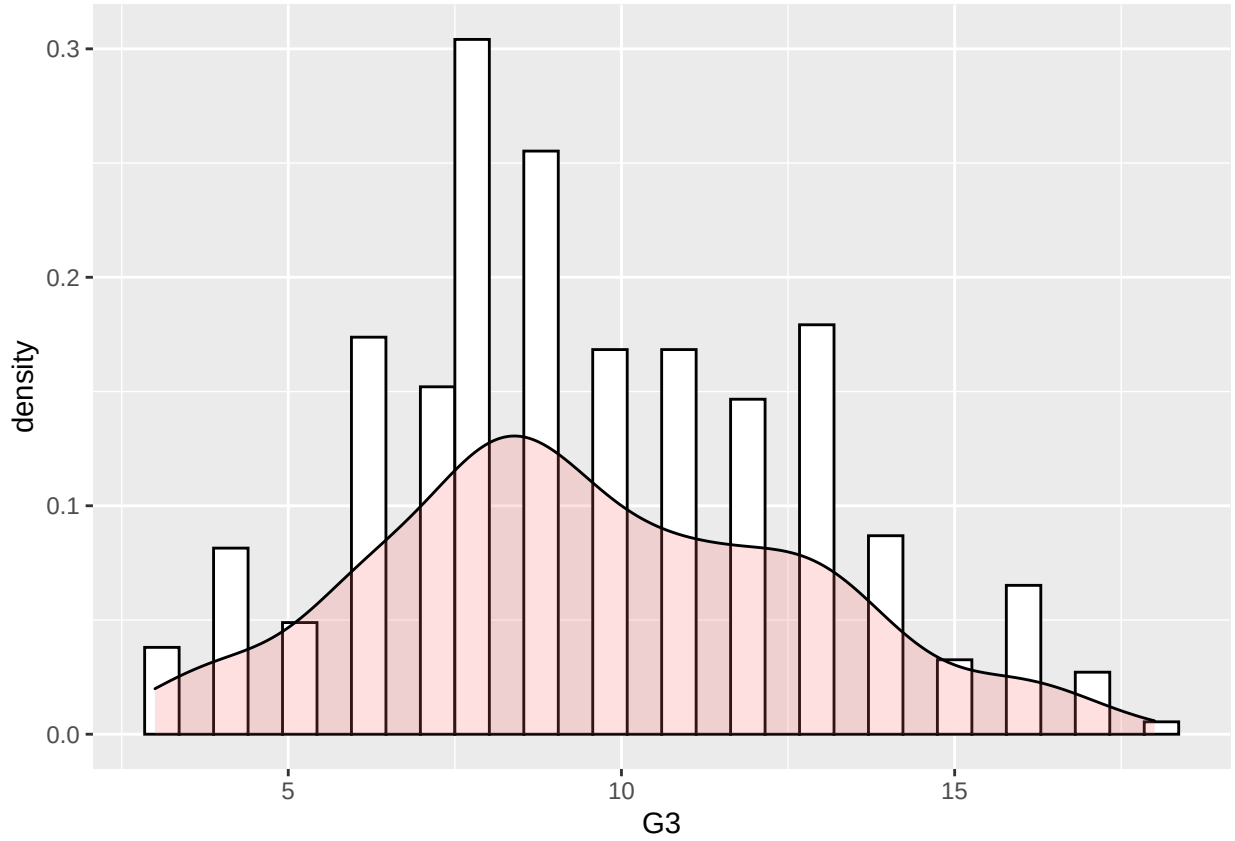



```
ggsave("display.0.G3.after.filtering.grade3.frequency.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(data=student4, aes(x=G3)) +
 geom_histogram(aes(y=..density..), colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("display.0.G3.after.filtering.grade3.density.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
student3 = student4

## TRANSFORMING G3 into RANGES of PASS and NO-PASS :

student3$G3 = as.integer(student3$G3)

student3$RESULT[student3$G3 <= 10] = "NO_PASS"
student3$RESULT[student3$G3 >=10 ] = "PASS"

student3 <- subset(student3, select = -c(G3))

student3$RESULT = as.factor(student3$RESULT)
```

4. DATA SUMMARY and VISUALIZATION

In the section 4, we aim to address the following Q1 from the course.

STEP 1 Data Descriptive Statistics

Q1. Amongst the variables of interest identify one that is categorical and one that is quantitative and then provide the following descriptive deliverables:

Summaries (Do this for at least one categorical and one quantitative variable).

a) For the categorical variable create a frequency distribution.

b) For the categorical variable create a bar diagram.

c) For the quantitative variable create numerical summaries grouped by a categorical variable.

d) For the quantitative variable create a histogram and a boxplot grouped by categorical variable.

4. DATA SUMMARY and VISUALIZATION

We display the GRADE G3 (NO PASS/PASS), function of AGE

```
## after we REMOVE the RECORDS where the GRADE G3 is > 2 ;
## we add a new piece of R code where we display the GRADE G3, function of AGE

student3 %>%
  group_by(RESULT, age) %>%
  summarise (n = n()) %>%
  mutate(freq = n / sum(n))
```

```
## `summarise()` has grouped output by 'RESULT'. You can override using the `.groups` argument.
```

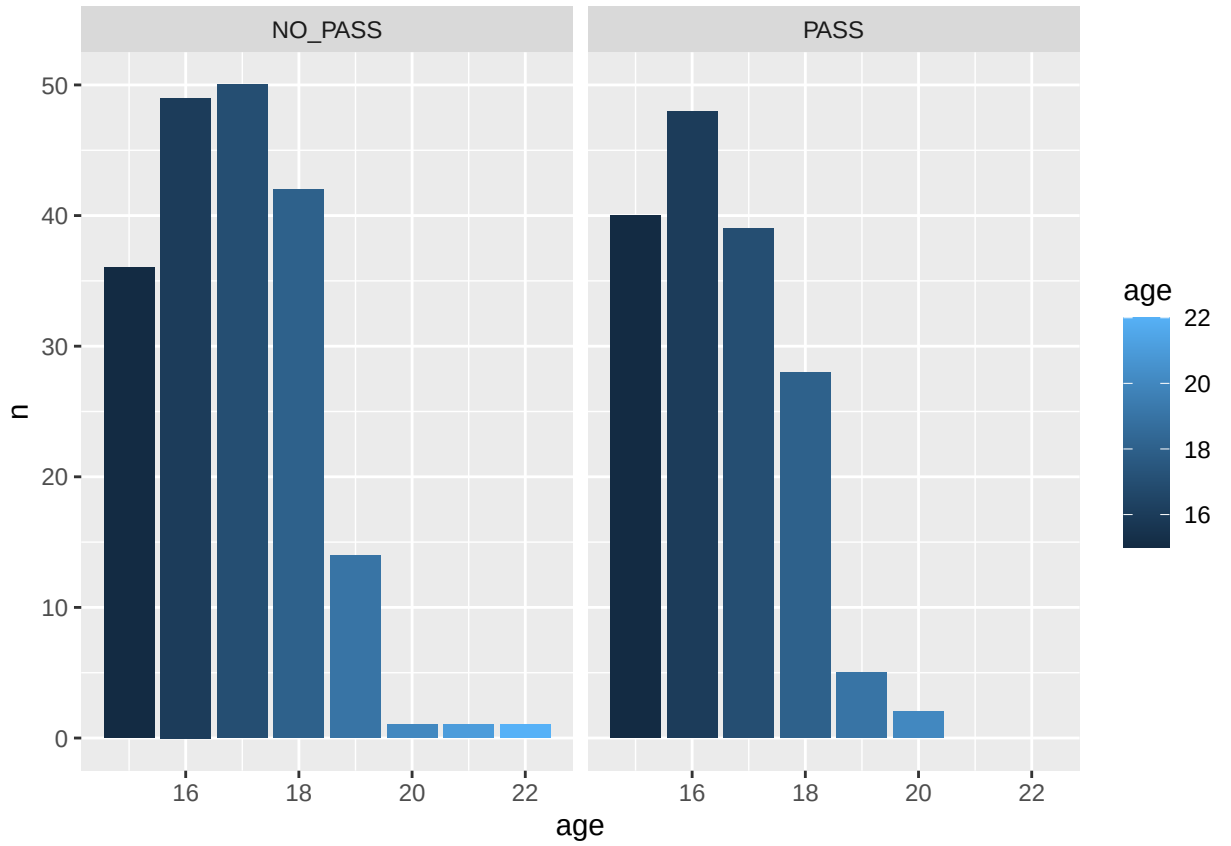
```
## # A tibble: 14 x 4
## # Groups:   RESULT [2]
##    RESULT    age     n    freq
##    <fct>   <int> <int>   <dbl>
##  1 NO_PASS    15    36 0.186
##  2 NO_PASS    16    49 0.253
##  3 NO_PASS    17    50 0.258
##  4 NO_PASS    18    42 0.216
##  5 NO_PASS    19    14 0.0722
##  6 NO_PASS    20     1 0.00515
##  7 NO_PASS    21     1 0.00515
##  8 NO_PASS    22     1 0.00515
##  9 PASS       15    40 0.247
## 10 PASS       16    48 0.296
## 11 PASS       17    39 0.241
## 12 PASS       18    28 0.173
## 13 PASS       19     5 0.0309
## 14 PASS       20     2 0.0123
```

```
# %>% arrange(desc(freq))

student3 %>%
  group_by(RESULT, age) %>%
  tally() %>%
  arrange(desc(n))
```

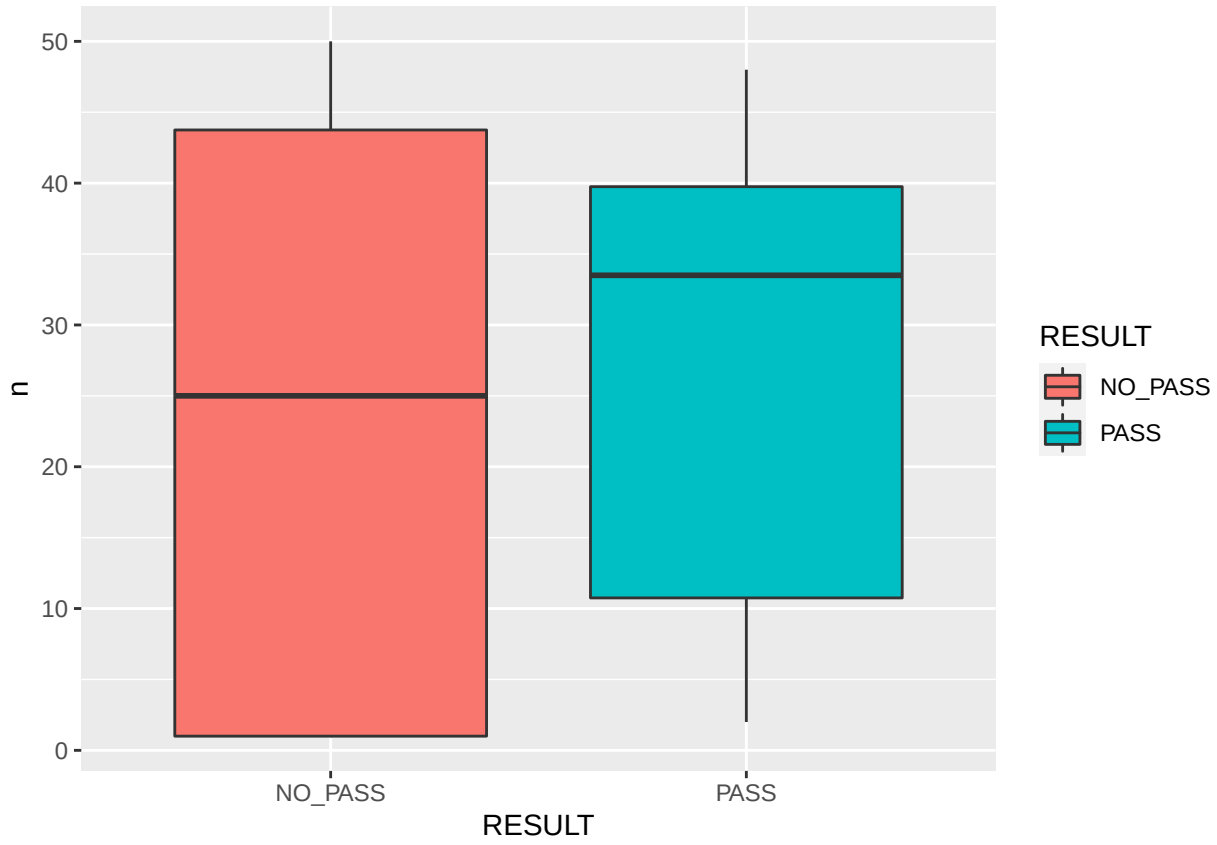
```
## # A tibble: 14 x 3
## # Groups:   RESULT [2]
##    RESULT    age     n
##    <fct>   <int> <int>
##  1 NO_PASS    17    50
##  2 NO_PASS    16    49
##  3 PASS       16    48
##  4 NO_PASS    18    42
##  5 PASS       15    40
##  6 PASS       17    39
##  7 NO_PASS    15    36
##  8 PASS       18    28
##  9 NO_PASS    19    14
```

```
## 10 PASS       19     5
## 11 PASS       20     2
## 12 NO_PASS    20     1
## 13 NO_PASS    21     1
## 14 NO_PASS    22     1
```

```
student3 %>%
  group_by(RESULT, age) %>%
  tally() %>%
  arrange(desc(n)) %>%
ggplot(aes(x = age, y=n)) +
       geom_bar(stat="identity", aes(fill=age)) +
       facet_wrap(~RESULT)
```




```
student3 %>%
  group_by(RESULT, age) %>%
  tally() %>%
  # arrange(desc(n)) %>%
ggplot(aes(x = RESULT, y=n)) +
       geom_boxplot(aes(fill=RESULT))
```


50
40
30
20
10
0
n
NO_PASS
PASS
RESULT
RESULT
NO_PASS
PASS

4. DATA SUMMARY and VISUALIZATION

We display the GRADE G3 (NO PASS/PASS), function of ABSENCES

```
## after we REMOVE the RECORDS where the GRADE G3 is > 2 ;
## we add a new piece of R code where we display the GRADE G3, function of ABSENCES

student3 %>%
  group_by(RESULT, absences) %>%
  summarise (n = n()) %>%
  mutate(freq = n / sum(n))
```

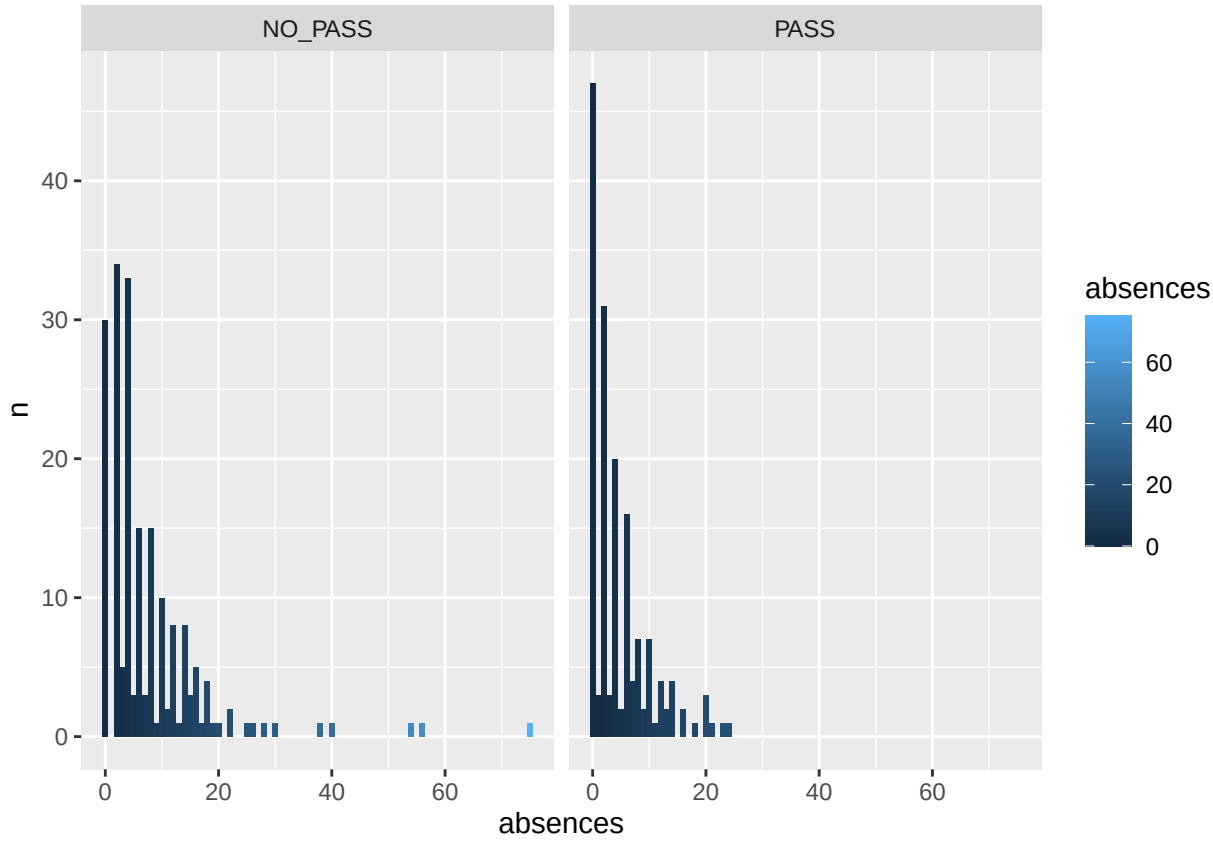
```
## `summarise()` has grouped output by 'RESULT'. You can override using the `.groups` argument.
```

```
## # A tibble: 51 x 4
## # Groups:   RESULT [2]
##    RESULT  absences     n    freq
##    <fct>      <int> <int>   <dbl>
##  1 NO_PASS        0    30 0.155
##  2 NO_PASS        2    34 0.175
##  3 NO_PASS        3     5 0.0258
##  4 NO_PASS        4    33 0.170
##  5 NO_PASS        5     3 0.0155
##  6 NO_PASS        6    15 0.0773
##  7 NO_PASS        7     3 0.0155
##  8 NO_PASS        8    15 0.0773
##  9 NO_PASS        9     1 0.00515
## 10 NO_PASS       10    10 0.0515
## # ... with 41 more rows
```

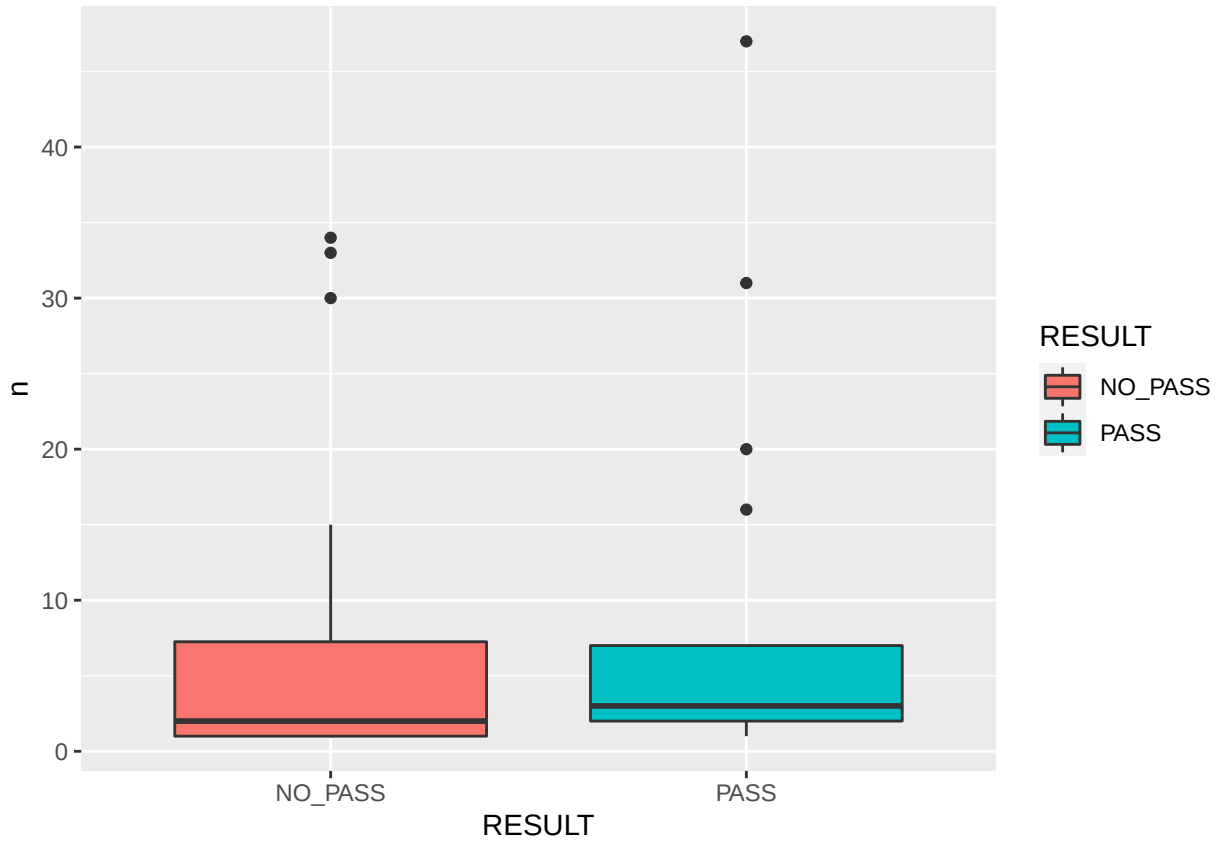
```
# %>% arrange(desc(freq))

student3 %>%
  group_by(RESULT, absences) %>%
  tally() %>%
  arrange(desc(n))
```

```
## # A tibble: 51 x 3
## # Groups:   RESULT [2]
##    RESULT  absences     n
##    <fct>      <int> <int>
##  1 PASS           0    47
##  2 NO_PASS        2    34
##  3 NO_PASS        4    33
##  4 PASS           2    31
##  5 NO_PASS        0    30
##  6 PASS           4    20
##  7 PASS           6    16
##  8 NO_PASS        6    15
##  9 NO_PASS        8    15
## 10 NO_PASS       10    10
## # ... with 41 more rows
```

```
student3 %>%
  group_by(RESULT, absences) %>%
  tally() %>%
  arrange(desc(n)) %>%
ggplot(aes(x = absences, y=n)) +
       geom_bar(stat="identity", aes(fill=absences)) +
      facet_wrap(~RESULT)
```




```
student3 %>%
  group_by(RESULT, absences) %>%
  tally() %>%
  # arrange(desc(n)) %>%
ggplot(aes(x = RESULT, y=n)) +
       geom_boxplot(aes(fill=RESULT))
```


40
30
20
10
0
n
NO_PASS
PASS
RESULT
RESULT
NO_PASS
PASS

4. DATA SUMMARY and VISUALIZATION : the CORRELATION PLOTS

We aim to address the following Q2 from the course

STEP 2 Correlation and Regression Analysis

although the data that we have chosen and the numerical features does not allow us a classical regression analysis. We will do it, just to set up the code in R (for other datasets).

Q2. Among the quantitative variables generate Relationships and Associations.

Correlation and Regression:

a) Identify two or more quantitative variables that might be correlated.

b) Find the correlation coefficient.

c) Create the scatter diagram under graphs.

d) Provide your rationale and justify your findings regarding the correlation between two quantitative variables of interest.

Here, we aim to answer also the question Q3 :

Q3.Prepare data by using the following preprocessing transformation and plots:

a) Please standardize the data.

b) Check for null values

c) Check for outliers

d) Check for Regression assumptions generate regression diagnostic plots.

4. DATA SUMMARY and VISUALIZATION : the CORRELATION PLOTS

We display the SCATTER PLOTS between the numerical features that we have the dataset i.e. AGE and ABSENCES (although the SCATTER PLOTS looks atypical for the data that we have chosen).

```
# library(Hmisc)
suppressMessages(library(Hmisc))

# computing the CORRELATION COEFFICIENT between AGE and ABSENCES ;
# we find a SMALL CORRELATION COEFFICIENT (< 0.3)

cor(student3$age, student3$absences)
```

```
## [1] 0.2152499
```

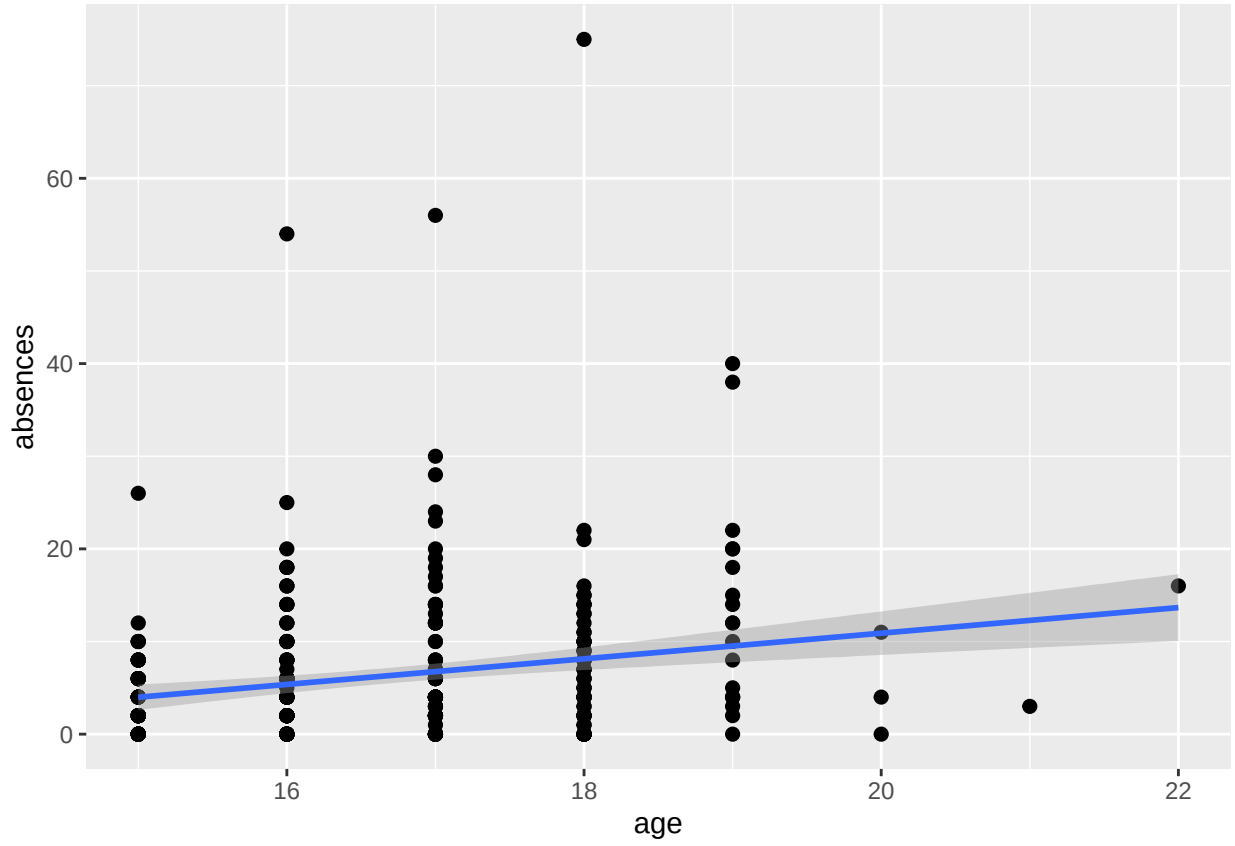
As the CORRELATION COEFFICIENT is small (<0.3), we can keep both AGE and ABSENCES in the model, as **INDEPENDENT FEATURES** (we know that some ML approaches are sensitive to features that are highly correlated).

```
cov(student3$age, student3$absences)
```

```
## [1] 2.229617
```

```
# a SCATTER PLOT using geom_smooth()

# ggplot(student3, aes(x=age, y=absences)) +
#       geom_point(size=2) +
#       geom_smooth()

# a SCATTER PLOT using geom_smooth(method=lm)

ggplot(student3, aes(x=age, y=absences)) +
       geom_point(size=2) +
       geom_smooth(method=lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



As an exercise in this section, as we look at the correlation between **AGE** and **ABSENCES**, we also perform a more formal linear regression analysis and compute the **DIAGNOSTIC PLOTS**.

```
library(broom) ### in order to add : AUGMENT

## A LM approach :

reg_model <- lm(absences~age, data = student3)

reg_model
```

```
##
## Call:
## lm(formula = absences ~ age, data = student3)
##
## Coefficients:
## (Intercept)          age
##     -16.753        1.383
```

```
## Listing R.squared in the LM approach :

summary(reg_model)$r.squared
```

```
## [1] 0.04633253
```

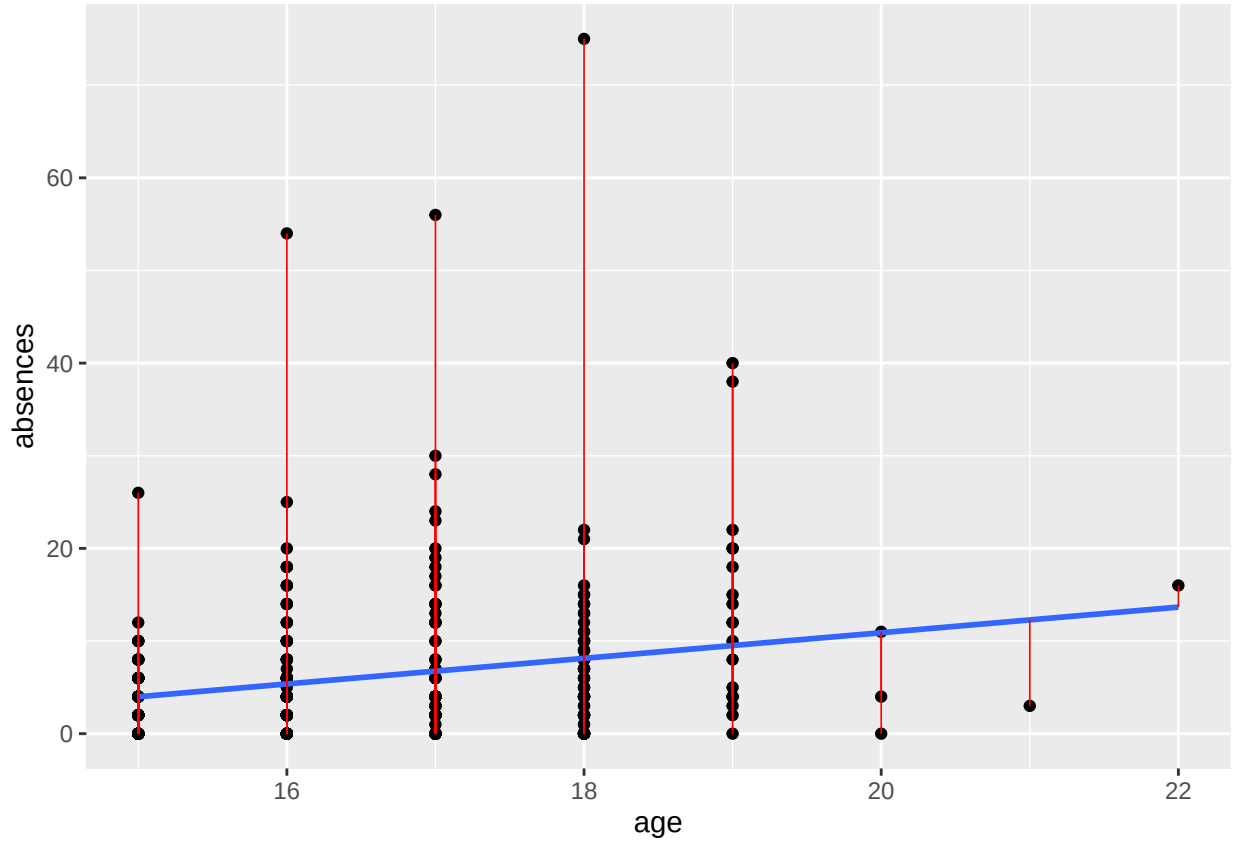
```
## Making the Diagnostic Plots:

reg_model.diagnostics <- augment(reg_model)
```

```
head(reg_model.diagnostics)
```

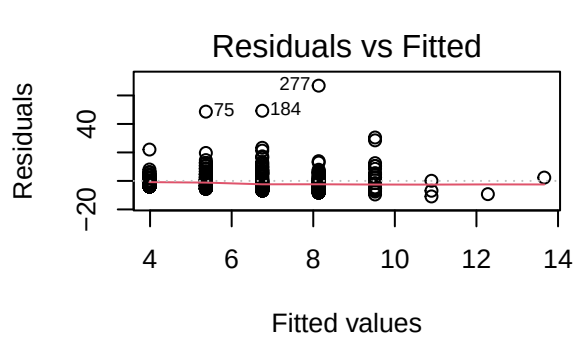
```
## # A tibble: 6 x 9
##   .rownames absences   age .fitted .resid    .hat .sigma   .cooksd .std.resid
##   <chr>        <int> <int>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>      <dbl>
## 1 1                6    18    8.13  -2.13 0.00597   7.99 0.000216      -0.268
## 2 2                4    17    6.75  -2.75 0.00302   7.99 0.000180      -0.345
## 3 3               10    15    3.99   6.01 0.00759   7.98 0.00219        0.757
## 4 4                2    15    3.99  -1.99 0.00759   7.99 0.000239      -0.250
## 5 5                4    16    5.37  -1.37 0.00356   7.99 0.0000527     -0.172
## 6 6               10    16    5.37   4.63 0.00356   7.98 0.000604       0.582
```

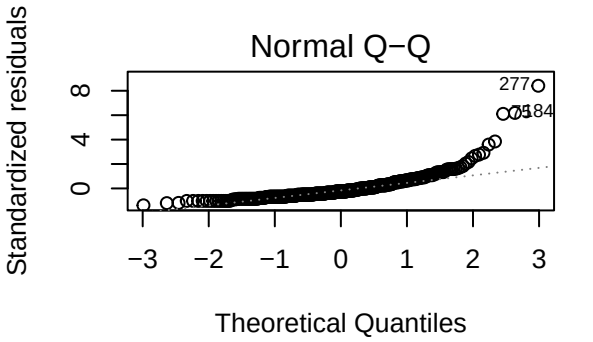
```
## Another view at the data ;
## potentially to identify the outlier values :

ggplot(reg_model.diagnostics, aes(age, absences)) +
geom_point() +
stat_smooth(method = lm, se = FALSE) +
geom_segment(aes(xend = age, yend = .fitted), color = "red", size = 0.3)
```

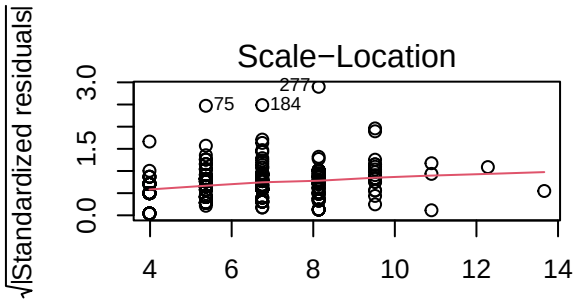
```
## `geom_smooth()` using formula 'y ~ x'
```

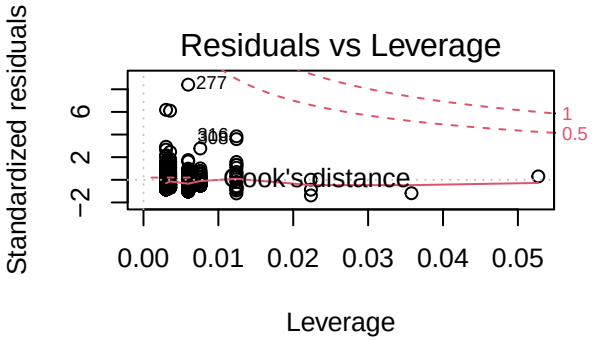



```
## A view at the LINEAR REGRESSION RESULTS :

par(mfrow = c(2, 2))
plot(reg_model)
```

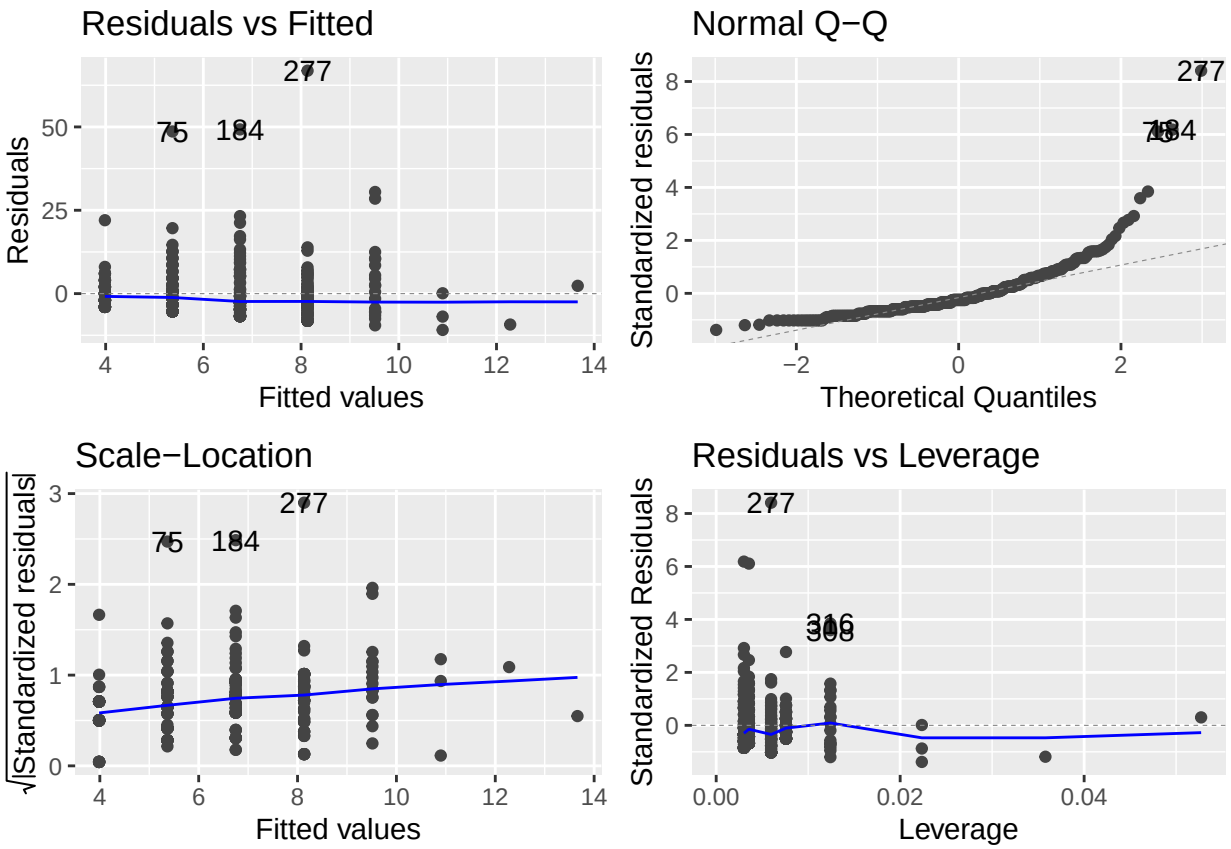









```
## Another view at the LINEAR REGRESSION RESULTS :

library(ggfortify)
autoplot(reg_model)
```



As we can see in the **Q-Q PLOT**, the distribution of the data is not GAUSSIAN, and there are 3 OUTLIER POINTS that have the INDEXES 75, 184, and 277 (below).

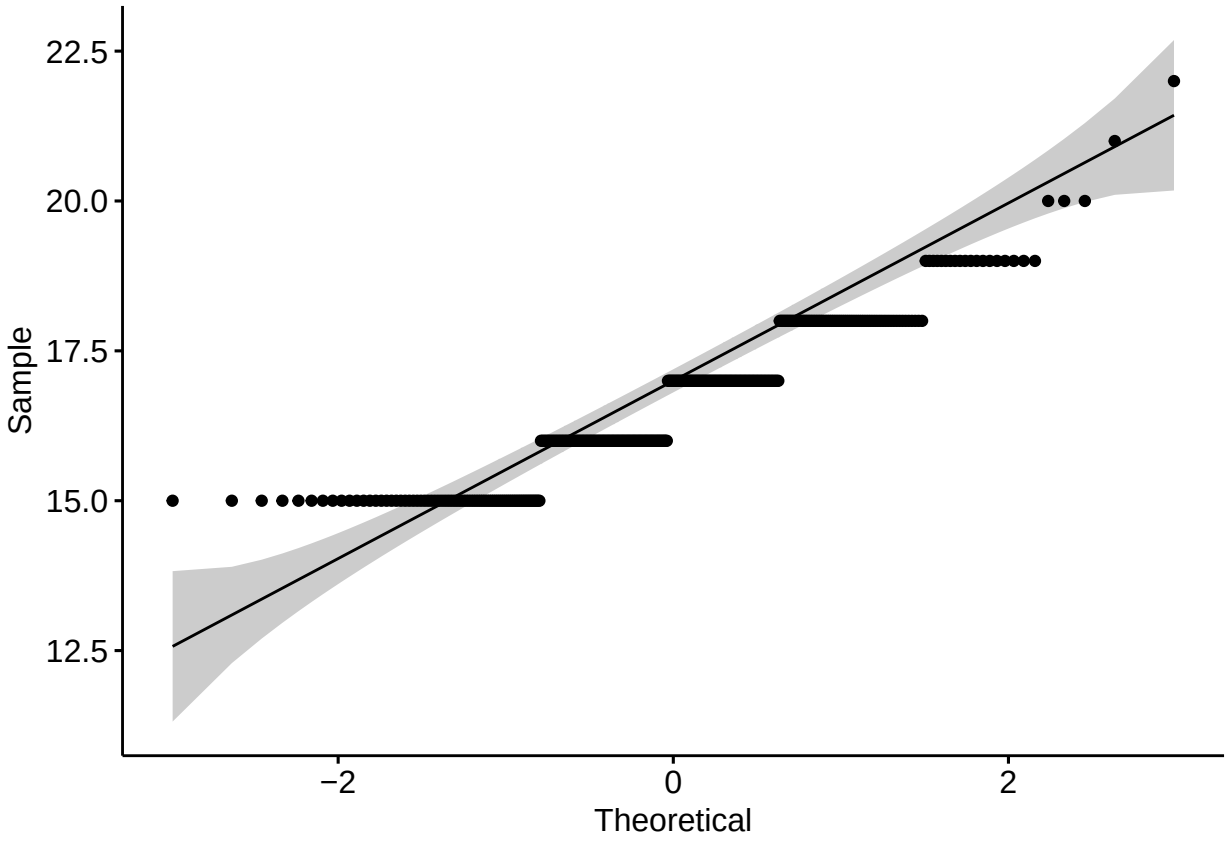
```
## Indeed, the DATA POINTS on ABSENCES that have the INDEXES 75, 184, 277,
## are the TOP OULIERS, and we may wanna remove these OUTLIERS from the data.

student3[75,]
```

```
##    age traveltime studytime failures absences  RESULT
## 75  16          1         2        0       54 NO_PASS
```

```
# student3[184,]
# student3[277,]
```

We also use several methods for normality testing such as **Kolmogorov-Smirnov (K-S) normality test** and **Shapiro-Wilk's test**. As we see below, the hypothesis of "normality" is rejected for both features "age" and "absences".

```
library(ggpubr)
ggqqplot(student3$age)
```




```
shapiro.test(student3$age)
```

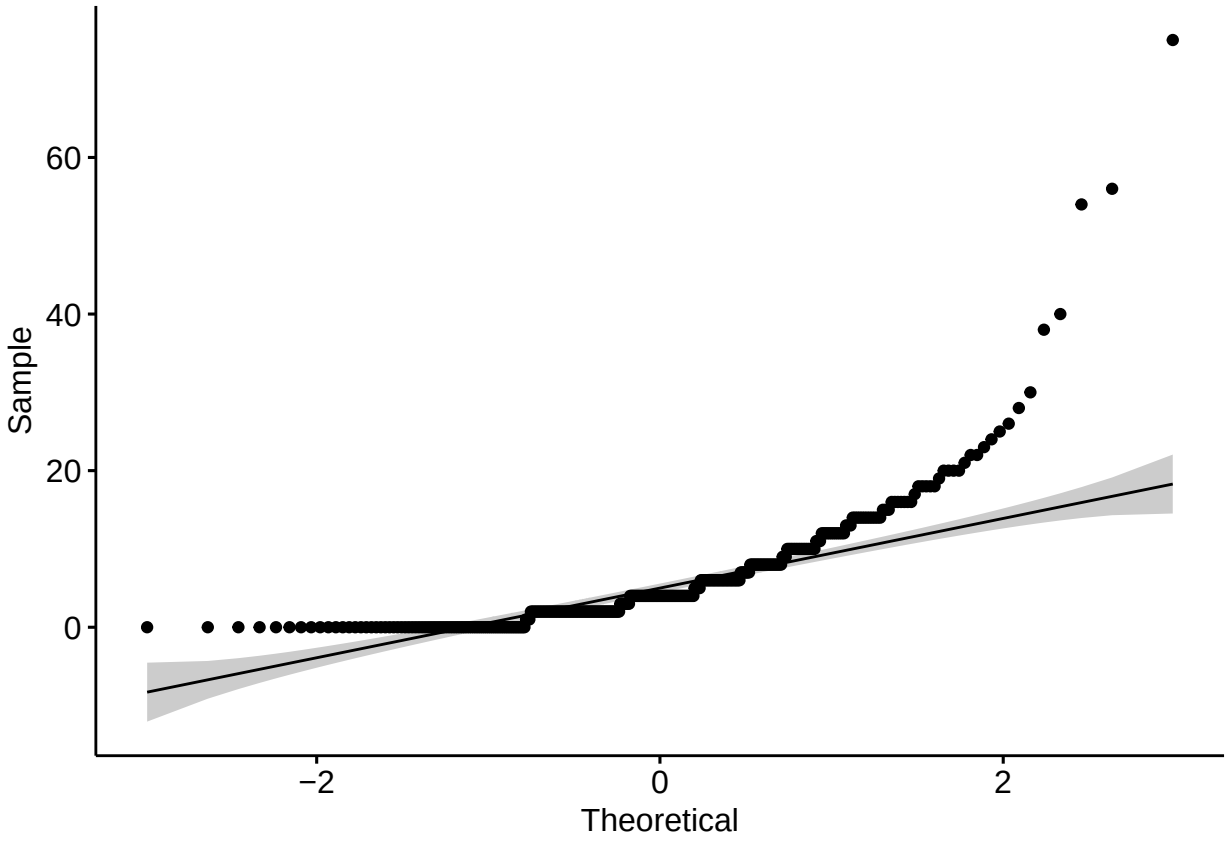
```
##
##  Shapiro-Wilk normality test
##
## data:  student3$age
## W = 0.90721, p-value = 5.763e-14
```

```
ks.test(student3$age, "pnorm")
```

```
## Warning in ks.test(student3$age, "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  student3$age
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
library(ggpubr)
ggqqplot(student3$absences)
```




```
shapiro.test(student3$absences)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  student3$absences
## W = 0.67768, p-value < 2.2e-16
```

```
ks.test(student3$absences, "pnorm")
```

```
## Warning in ks.test(student3$absences, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  student3$absences
## D = 0.75253, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

5. TRAINING AND TEST SETS

```
## CHOOSING the TRAINING and the TESTING SETS

indxTrain <- createDataPartition(student3$RESULT,
                                 p = .70,
                                 list = FALSE)

training <- student3[indxTrain,]
# training

testing <- student3[-indxTrain,]
# testing

dim(student3)
```

```
## [1] 356   6
```

```
dim(training)
```

```
## [1] 250   6
```

```
dim(testing)
```

```
## [1] 106   6
```

6. PRE-PROCESSING THE DATA

We can pre-process the data in a manner that is shown below, by using the COMMAND "preProcess" and "method = c("center","scale")", although it is likely easier to do the pre-processing by using the option "preProcess = c("center","scale")" in *train()*.

```
## PRE-PROCESSING the DATA

trainX        <- training[, names(training) != "RESULT"]

preProcValues <- preProcess(x = trainX, method = c("center", "scale"))

# preProcValues

names(trainX)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
```

```
dim(trainX)
```

```
## [1] 250   5
```

```
names(training)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
## [6] "RESULT"
```

```
names(testing)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
## [6] "RESULT"
```

```
scaledTrain <- predict(preProcValues, trainX)
```

7. PERFORMING THE TRAINING

We cover in the following sections the following :

Step 4 Implement Regression and Decision Trees

Conduct Regression and answer the following questions:

Q4 Implement Regression and Decision Tree.

a) Objective and rationale of using the specific algorithm to achieve the objective.

b) Steps of implementing the algorithm with regards to the context.

c) Interpretation of the results and prediction accuracy achieved.

d) Performance improvement techniques and improved accuracy achieved.

Use feature selection, variable importance, compare RMSE(Regression) across models and Information gain (Decision Trees), K-fold cross validation, grid search etc.

e) Implement the two algorithms and state the insights obtained from the implemented project.

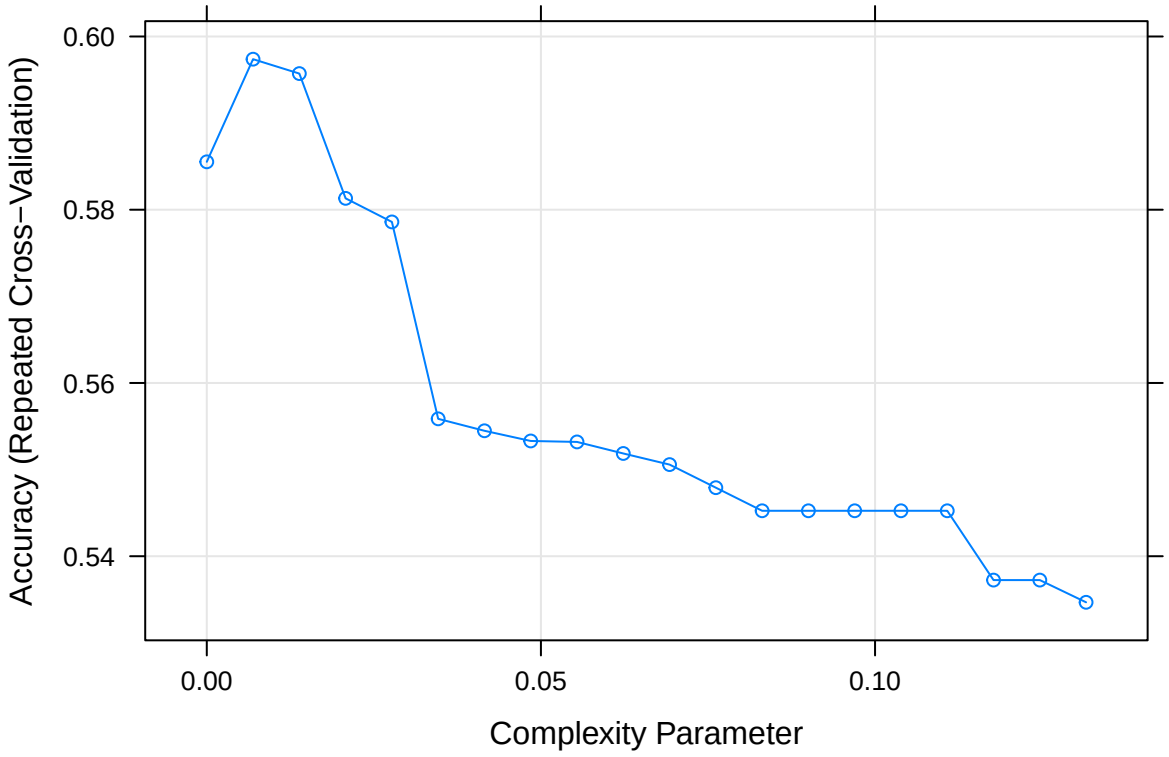
7. PERFORMING THE TRAINING

```
## PERFORMING the TRAINING

set.seed(400)
ctrl <- trainControl(method="repeatedcv", repeats = 3)

rpartFit <- train( RESULT~ .,
                   data = training,
                   method = "rpart",
                   trControl = ctrl,
                   preProcess = c("center","scale"), tuneLength = 20)

## The output of rpartFit

rpartFit
```

```
## CART
##
## 250 samples
##   5 predictor
##   2 classes: 'NO_PASS', 'PASS'
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 225, 226, 224, 225, 225, 224, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.000000000  0.5855299   0.165413546
##   0.006925208  0.5973803   0.191076588
##   0.013850416  0.5957179   0.183528032
##   0.020775623  0.5813120   0.145796673
##   0.027700831  0.5785897   0.135930269
##   0.034626039  0.5558590   0.095223364
##   0.041551247  0.5544786   0.081210972
##   0.048476454  0.5533034   0.070681772
##   0.055401662  0.5531923   0.065080454
##   0.062326870  0.5518590   0.052701444
##   0.069252078  0.5505769   0.045445650
##   0.076177285  0.5479103   0.039345433
##   0.083102493  0.5452436   0.031770953
##   0.090027701  0.5452436   0.031770953
##   0.096952909  0.5452436   0.031770953
##   0.103878116  0.5452436   0.031770953
##   0.110803324  0.5452436   0.031770953
##   0.117728532  0.5372436   0.013966295
##   0.124653740  0.5372436   0.013966295
##   0.131578947  0.5346795  -0.005510597
##
## Accuracy was used to select the optimal model using the largest value.
```

```
## The final value used for the model was cp = 0.006925208.
```

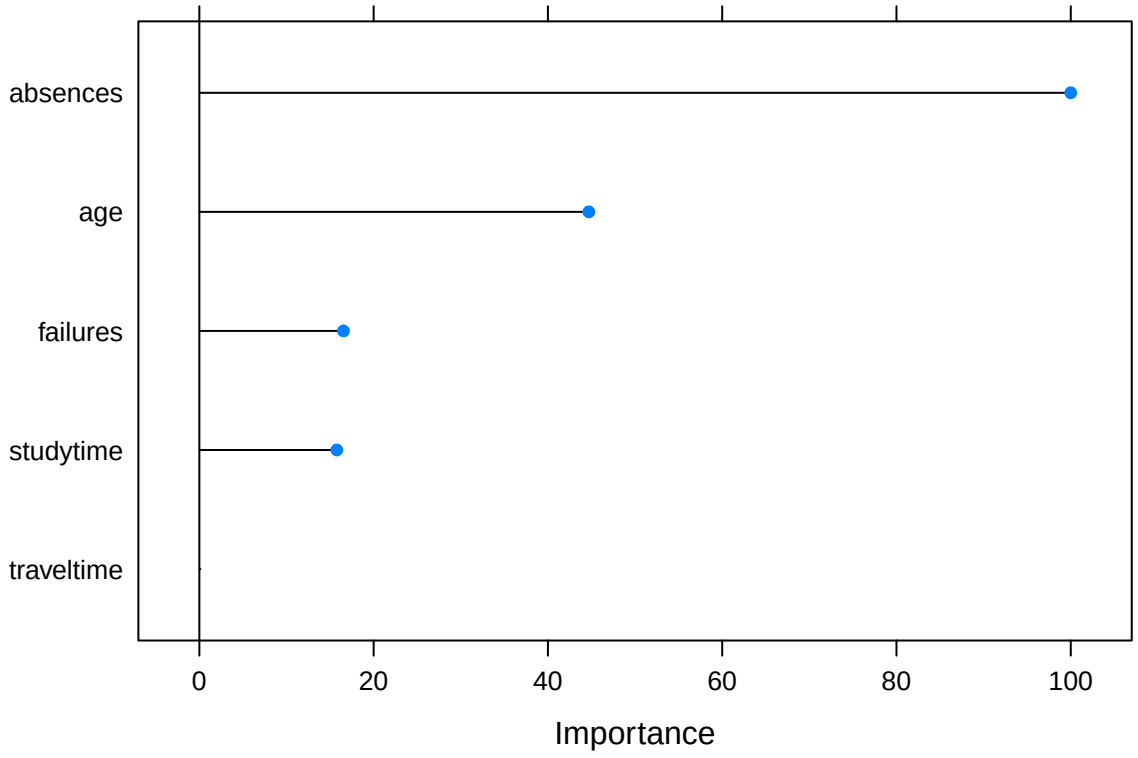
```
## summary(rpartFit$finalModel)
## it outputs a very long summary

## The plot of rpartFit

plot(rpartFit)
```




```
png("the.results.rpartFIT.png")
plot(rpartFit)
dev.off()
```

```
## pdf
##   2
```

```
## To look at the VARIABLE IMPORTANCE

X <- varImp(rpartFit)
plot(X)
```

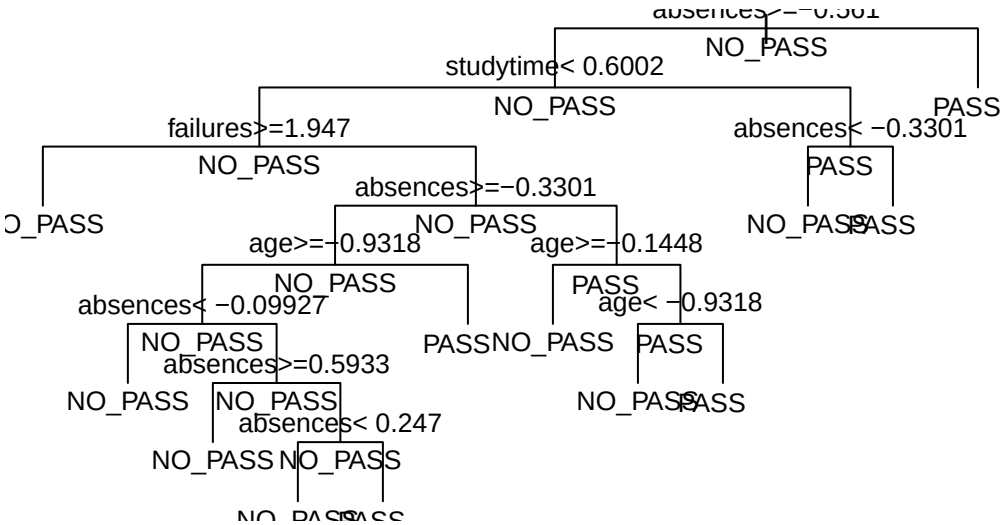


As we can note, in the current model, **more important FEATURES are AGE, ABSENCES, and STUDY TIME.**

```
### DISPLAYING THE TREE

plot(rpartFit$finalModel,
    uniform=TRUE,
    main="Classification Tree")
text(rpartFit$finalModel, use.n.=TRUE, all=TRUE, cex=.8)
```

```
## Warning in text.default(xy$x, xy$y + 0.5 * cxy[2L], rows[left.child], ...):
## "use.n." is not a graphical parameter
```

```
## Warning in text.default(xy$x[leaves], xy$y[leaves] - 0.5 * cxy[2L], stat, :
## "use.n." is not a graphical parameter
```



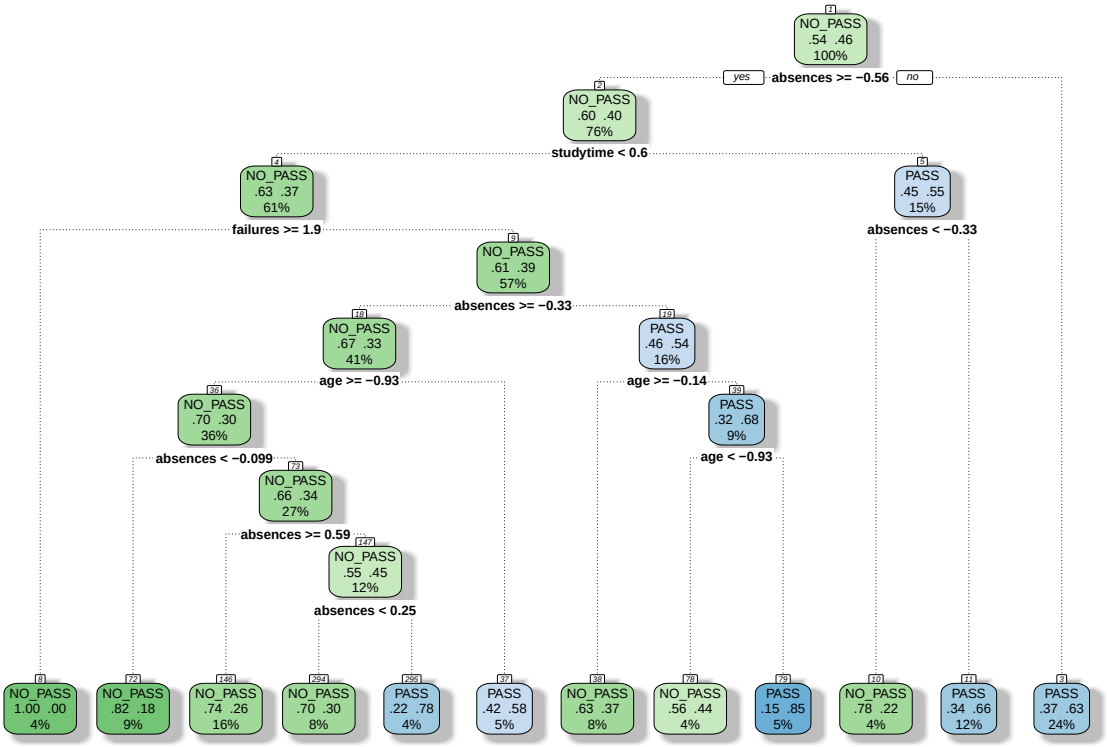

```
png("the.results.rpartFIT.finalModel.png")
plot(rpartFit$finalModel,
    uniform=TRUE,
    main="Classification Tree")
text(rpartFit$finalModel, use.n.=TRUE, all=TRUE, cex=.8)
```

```
## Warning in text.default(xy$x, xy$y + 0.5 * cxy[2L], rows[left.child], ...):
## "use.n." is not a graphical parameter

## Warning in text.default(xy$x, xy$y + 0.5 * cxy[2L], rows[left.child], ...):
## "use.n." is not a graphical parameter
```

```
dev.off()
```

```
## pdf
##   2
```

```
# library(rattle)
suppressMessages(library(rattle))

fancyRpartPlot(rpartFit$finalModel)
```




```
png("the.results.rpartFIT.fancyR.png")
fancyRpartPlot(rpartFit$finalModel)
dev.off()
```

```
## pdf
##   2
```

8. MAKING THE PREDICTIONS

```
## Making the PREDICTIONS :

rpartPredict <- predict(rpartFit, newdata = testing)

# rpartPredict
```

We may aim to optimize the model by FEATURE SELECTION or by including NEW FEATURES from the data that is available (we have excluded at the beginning many fetures).

9. THE CONFUSION MATRIX

```
## COMPUTING the CONFUSION MATRIX :

confusionMatrix(rpartPredict, testing$RESULT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction NO_PASS PASS
##    NO_PASS      35   18
##    PASS         23   30
##
##                Accuracy : 0.6132
##                  95% CI : (0.5137, 0.7062)
##     No Information Rate : 0.5472
##     P-Value [Acc > NIR] : 0.1019
##
##                   Kappa : 0.2264
##
##  Mcnemar's Test P-Value : 0.5322
##
##             Sensitivity : 0.6034
##             Specificity : 0.6250
##          Pos Pred Value : 0.6604
##          Neg Pred Value : 0.5660
##              Prevalence : 0.5472
##          Detection Rate : 0.3302
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.6142
##
##        'Positive' Class : NO_PASS
##
```

```
mean(rpartPredict == testing$RESULT)
```

```
## [1] 0.6132075
```

```
dim(student3)
```

```
## [1] 356   6
```

The ACCURACY of the MODEL based on DECISION TREES is :

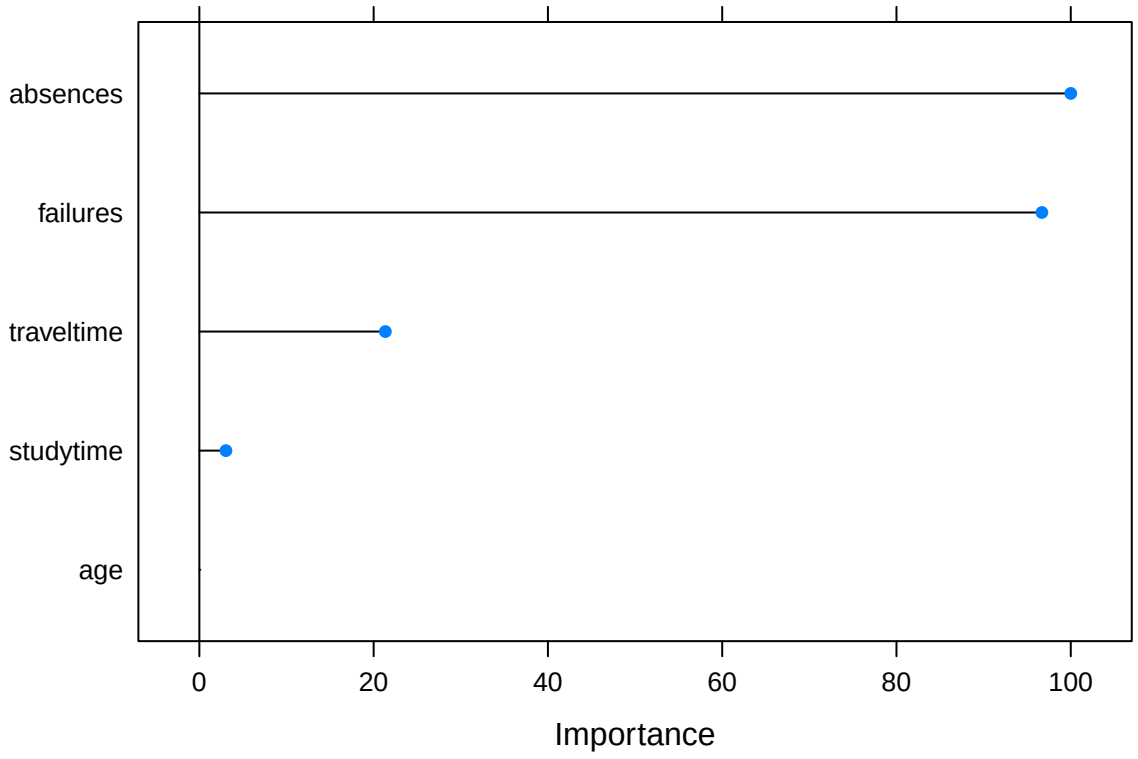
```
mean(rpartPredict == testing$RESULT)
```

```
## [1] 0.6132075
```

10. CONCLUSIONS AND OTHER MODELS

Because the LABEL is BINARY (PASS/NO PASS), we can compare the performance of the model above with the performance of a model that is based on **LOGISTIC REGRESSION**.

```
logisticFit = train( RESULT ~ .,
  data = training,
  trControl = ctrl,
  method = "glm",
  family = "binomial",
  preProcess = c("center","scale"), tuneLength = 20)

## To look at the VARIABLE IMPORTANCE

X <- varImp(logisticFit)
plot(X)
```




```
## To compute the PREDICTIONS
logisticPredict <- predict(logisticFit, newdata = testing)

## To display the CONFUSION MATRIX
confusionMatrix(logisticPredict, testing$RESULT)
```

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

```
##
##           Reference
## Prediction NO_PASS PASS
##    NO_PASS      37   18
##    PASS         21   30
##
##                Accuracy : 0.6321
##                  95% CI : (0.5329, 0.7237)
##     No Information Rate : 0.5472
##     P-Value [Acc > NIR] : 0.04781
##
##                   Kappa : 0.2615
##
##  Mcnemar's Test P-Value : 0.74877
##
##             Sensitivity : 0.6379
##             Specificity : 0.6250
##          Pos Pred Value : 0.6727
##          Neg Pred Value : 0.5882
##              Prevalence : 0.5472
##          Detection Rate : 0.3491
##    Detection Prevalence : 0.5189
##       Balanced Accuracy : 0.6315
##
##        'Positive' Class : NO_PASS
##
```

```
mean(logisticPredict == testing$RESULT)
```

```
## [1] 0.6320755
```

The ACCURACY of the MODEL based on LOGISTIC REGRESSION is :

```
mean(logisticPredict == testing$RESULT)
```

```
## [1] 0.6320755
```

As we can see, by comparing the **ACCURACY**, the ML model that is based on **DECISION TREES** performs better than the ML model that is based on **LOGISTIC REGRESSION**.

Also the **FEATURES** that are considered as important differ between these two models : the LOGISTIC REGRESSION model emphasizes more on “failures”, “absences”, and “traveltime”, and less on “studytime” and on “age”, in sharp contrast with the model based on DECISION TREES.

A note to add about the model based on DECISION TREES, we do not have to standardize the data..