

KNN to predict the Grade (no pass/pass)

Bogdan Tanasa

L2

THE SECTIONS in the RMARKDOWN DOCUMENT :

1. INTRODUCTION

2. DATA EXPLORATION

3. DATA FILTERING

4. DATA TRANSFORMATION

5. TRAINING AND TEST SETS

6. PRE-PROCESSING THE DATA

7. PERFORMING THE TRAINING

8. MAKING THE PREDICTIONS

9. THE CONFUSION MATRIX

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 $1 \leq n < 3$, 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(tibble)

library(class)
library(gmodels)
library(caret)
library(e1071)

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

FILE1="student.mat.txt"
# FILE2="student.por.txt"
# FILE3="student.mat.and.por.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"
```

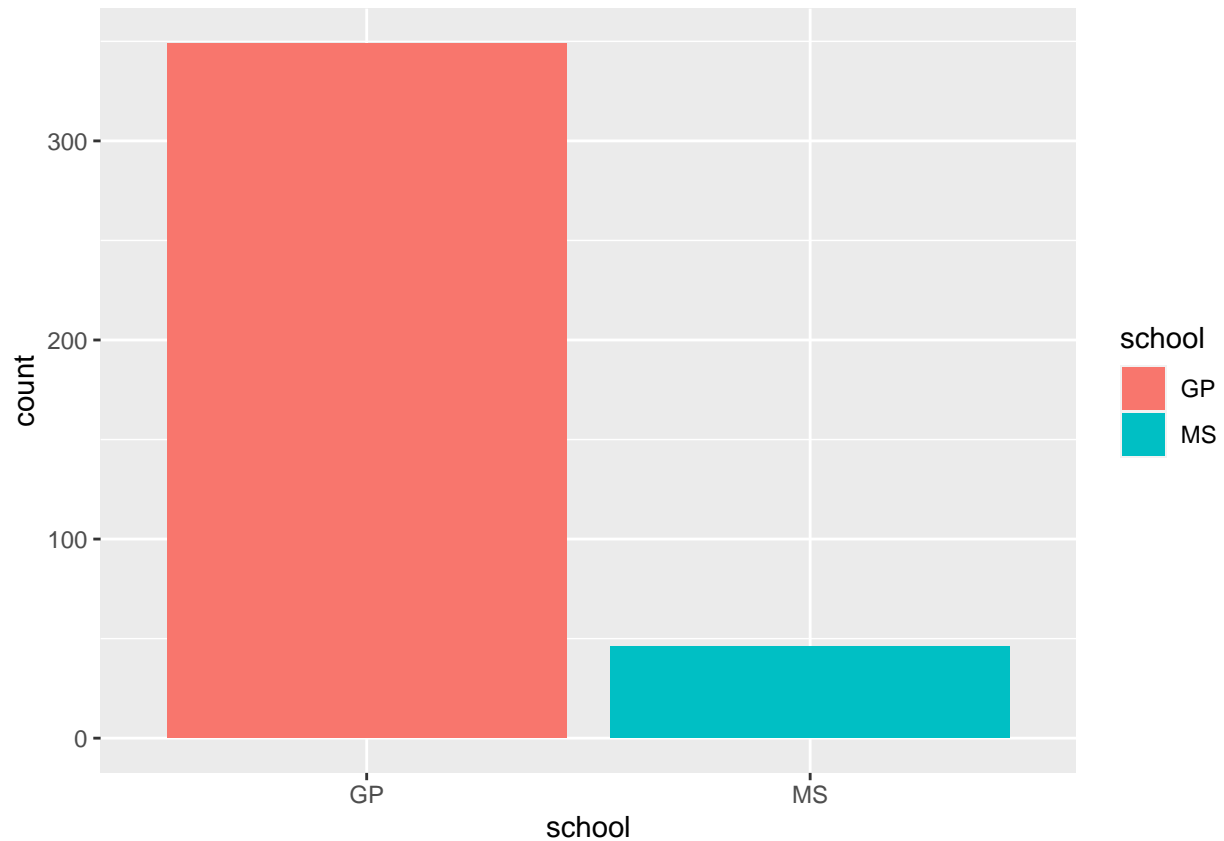
Here we are starting to display the data for visual exploration.

```
#####
#####
# 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")
```

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

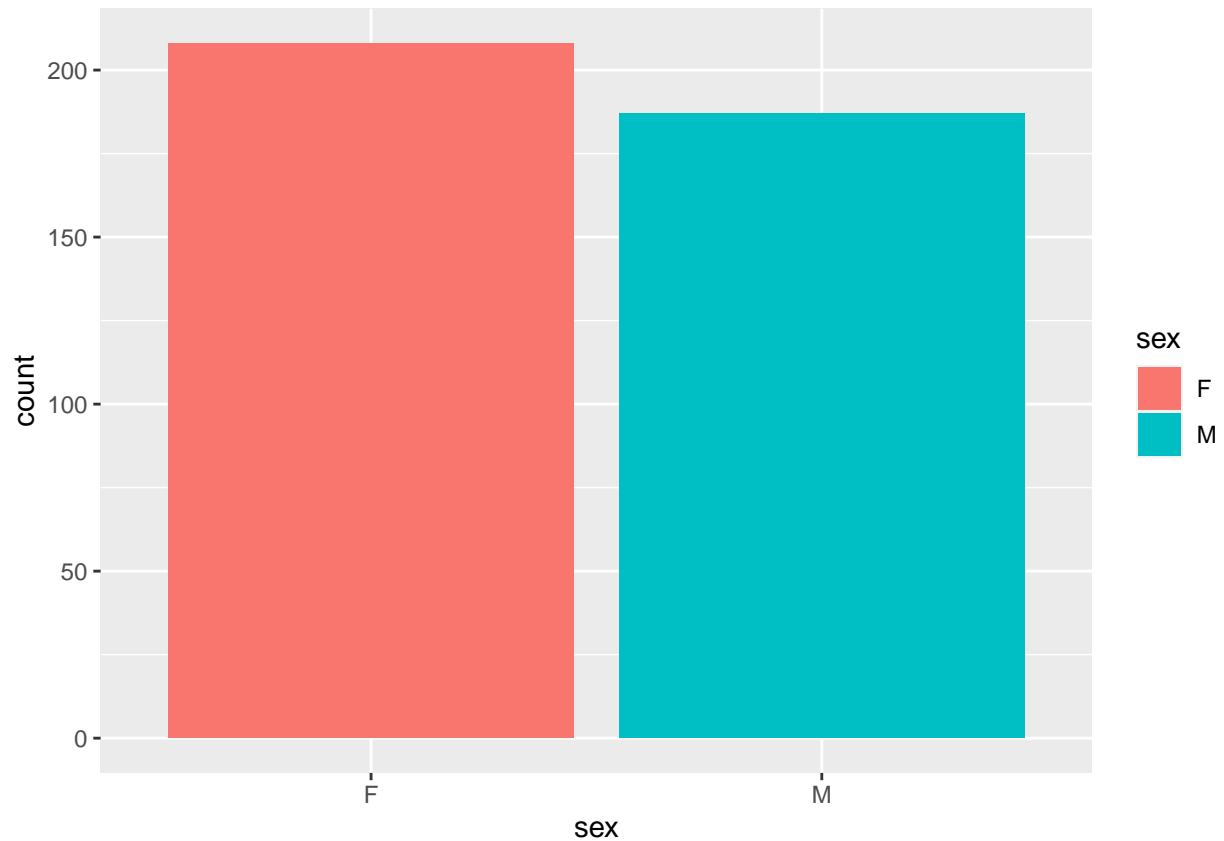
```
student$school = as.factor(student$school)
```

```
#####  
#####  
# 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")
```

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

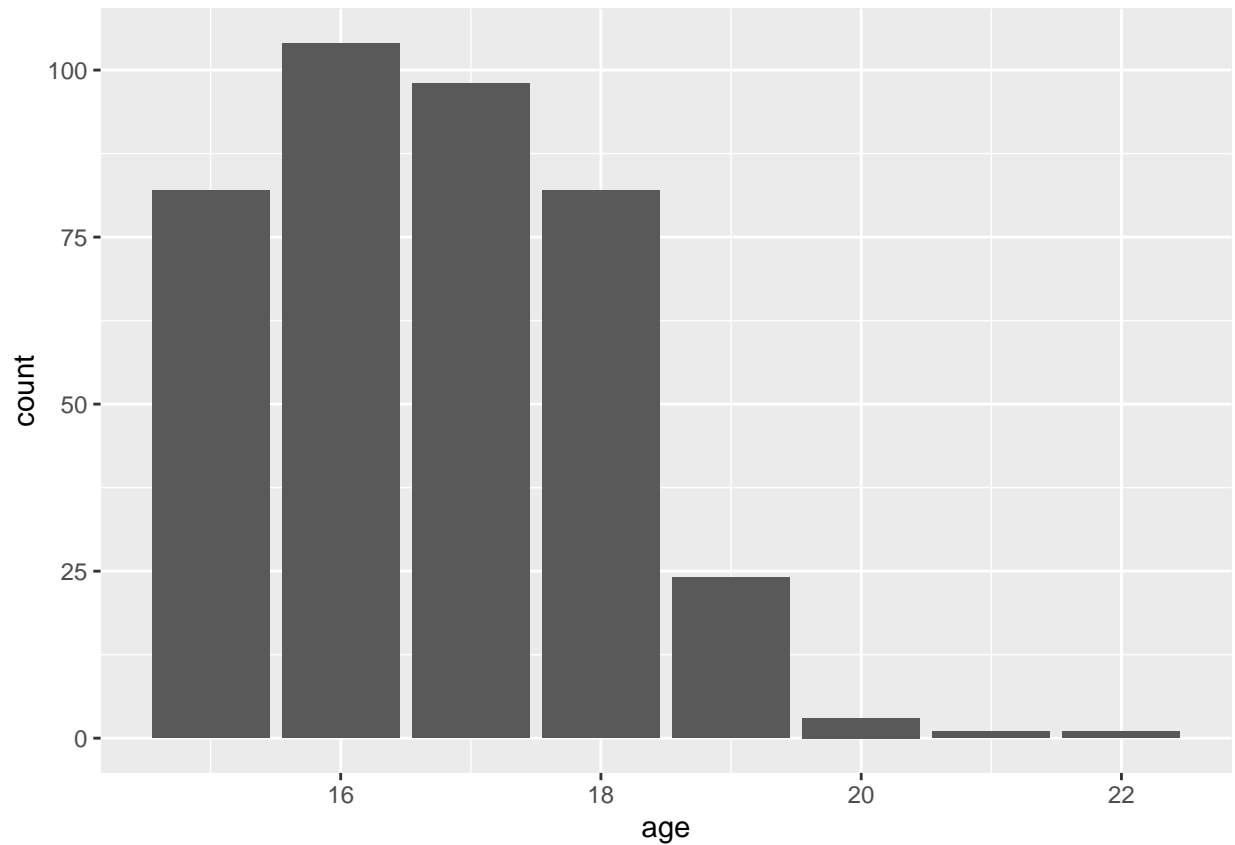
```
student$sex = as.factor(student$sex)
```

```
#####  
#####  
# 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))
```



```
ggsave("display.3.age.png")
```

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

```
# AGE is already on the numerical scale !!
```

```
student$age = as.integer(student$Age)
```

```
#####
```

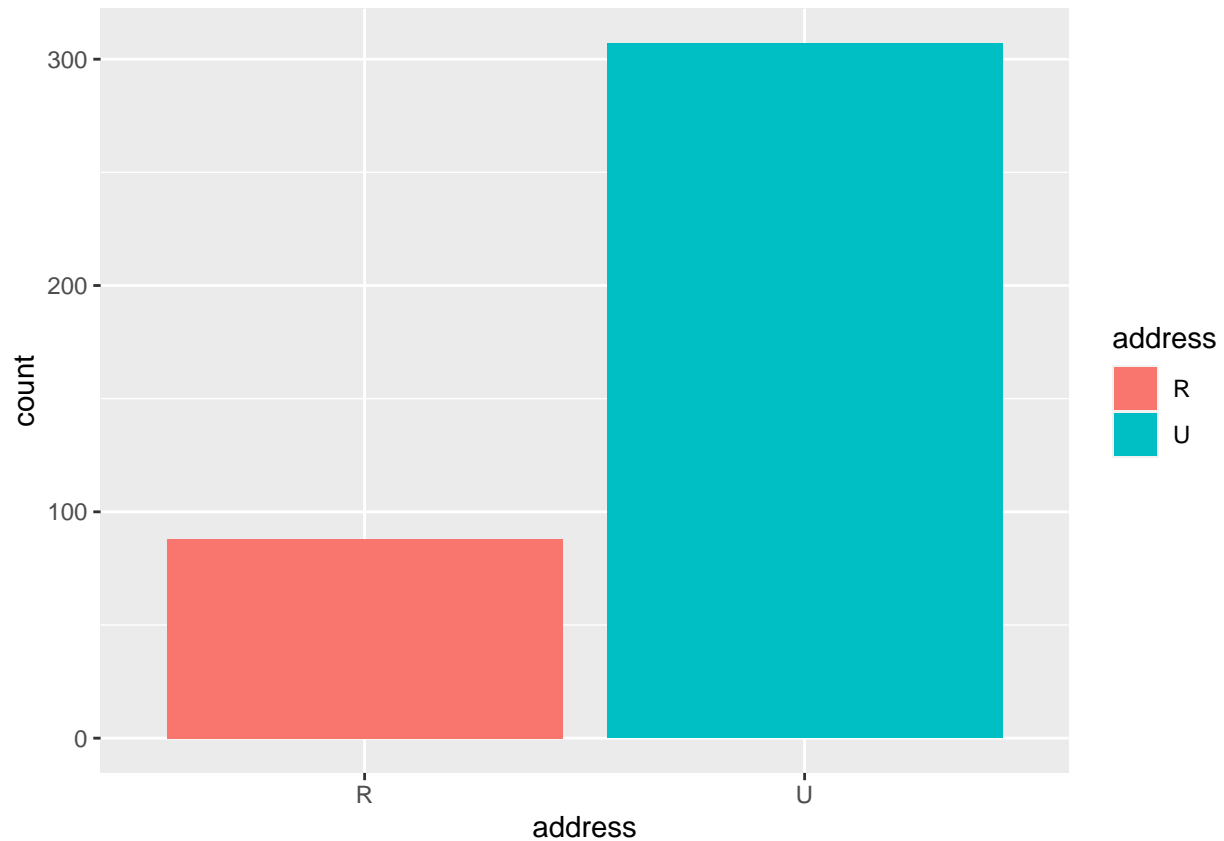
```
#####
```

```
# 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")
```

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

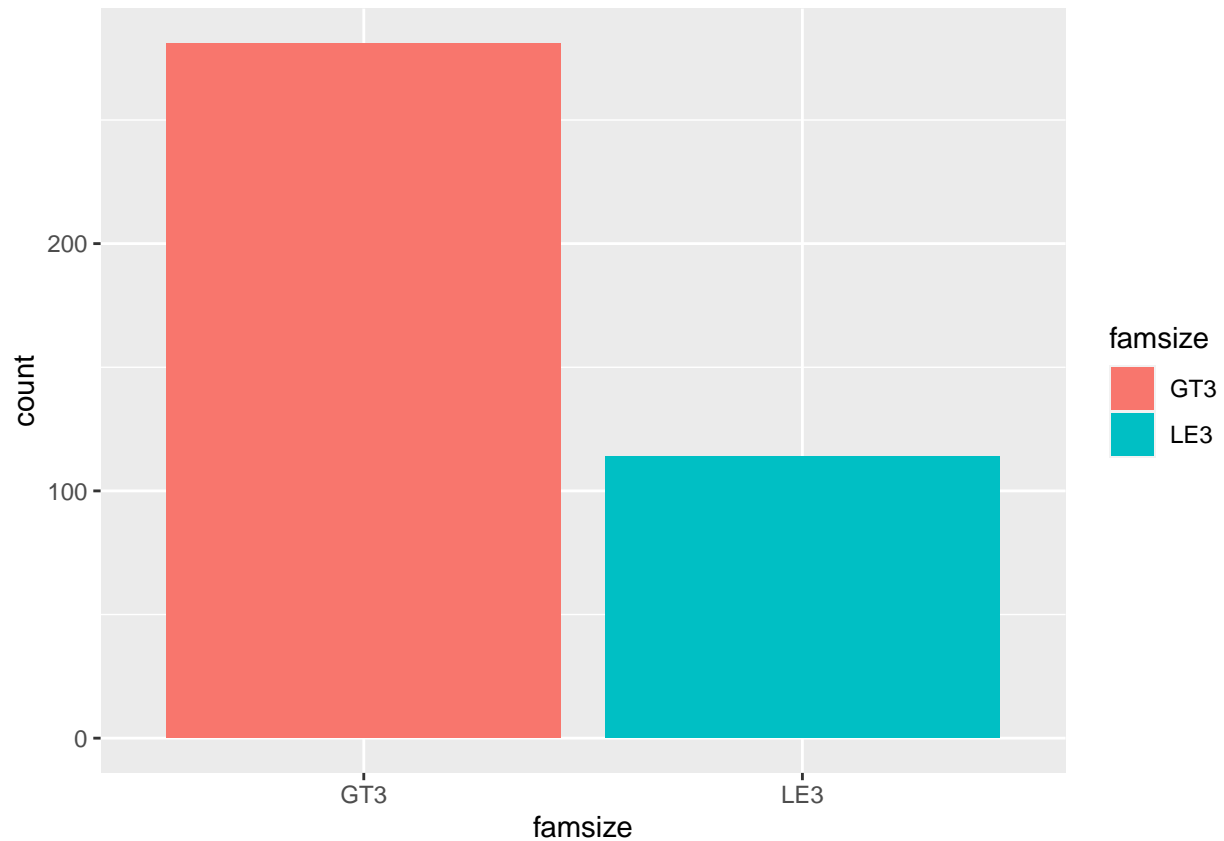
```
student$address = as.factor(student$address)
```

```
#####  
#####  
# 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")
```

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

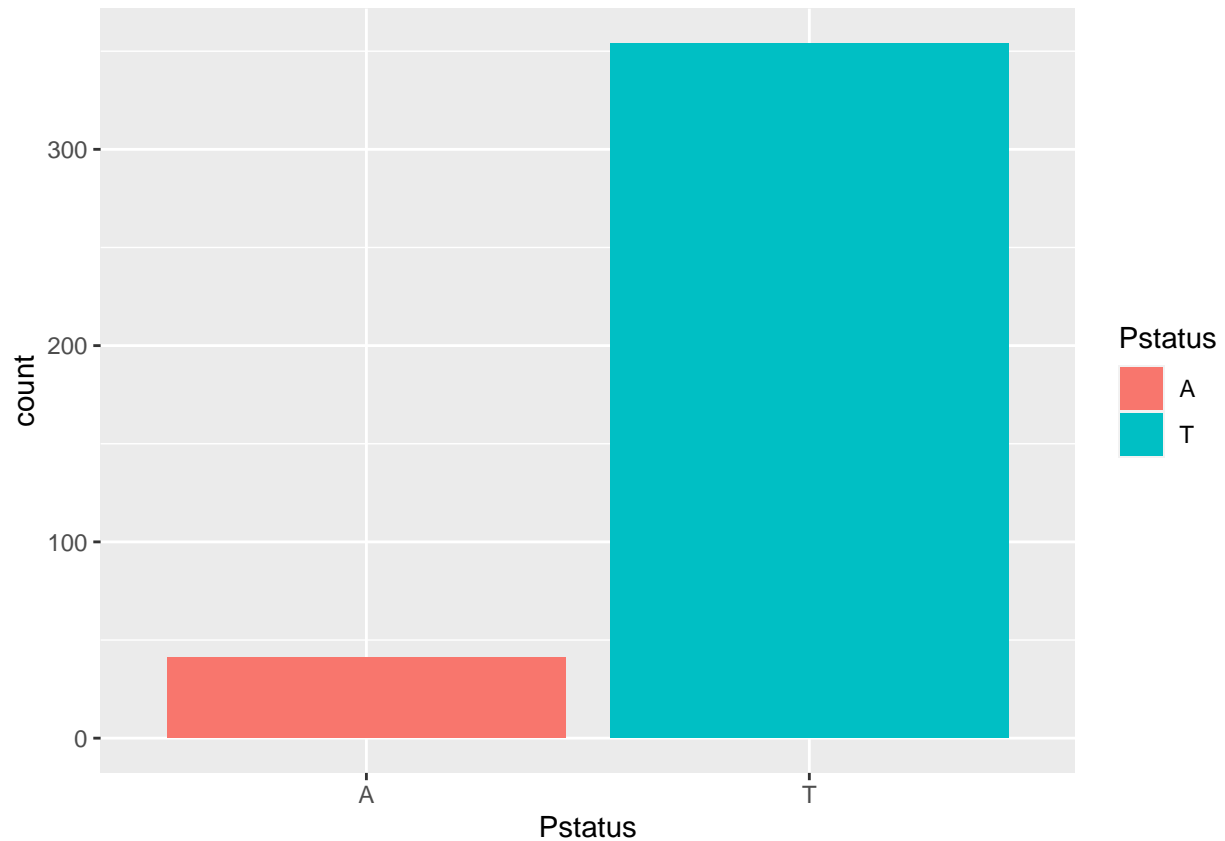
```
student$famsize = as.factor(student$famsize)
```

```
#####
#####
# 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")
```

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

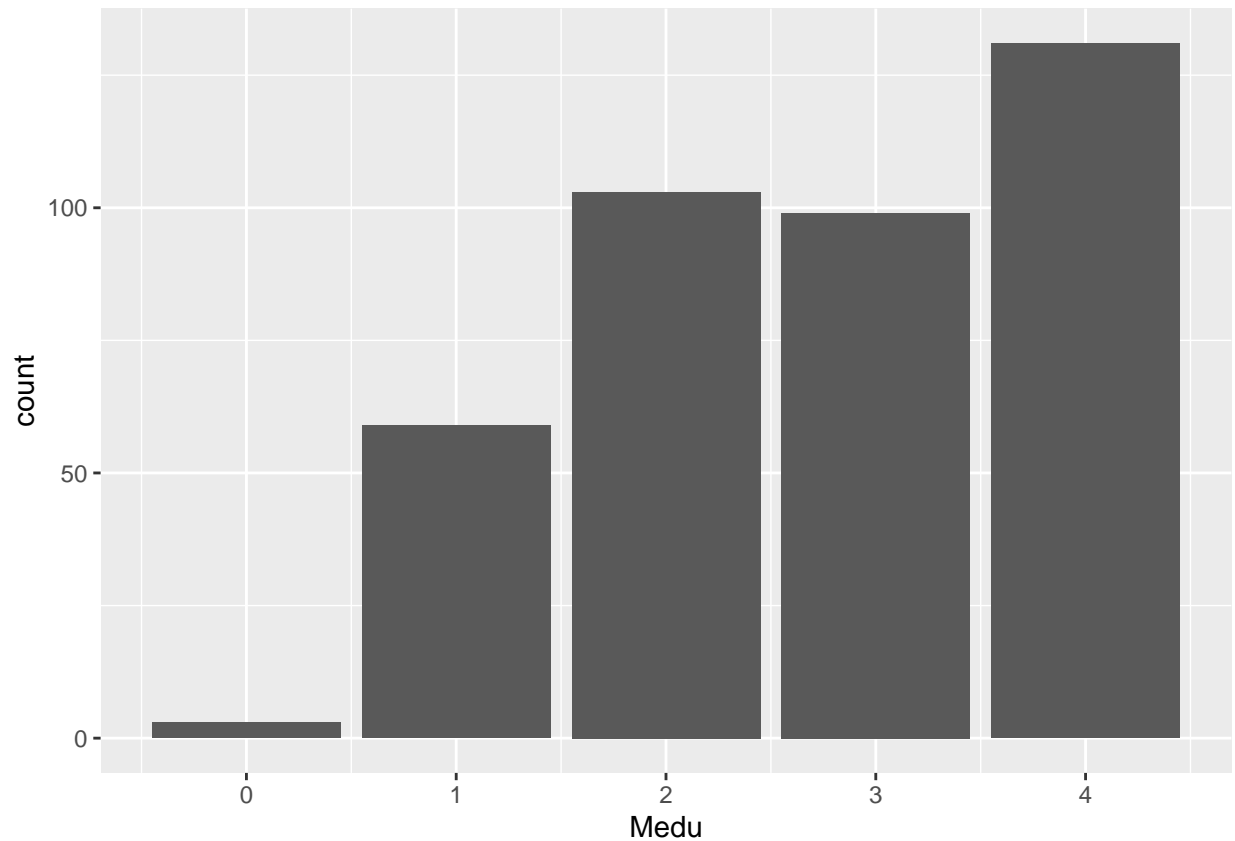
```
student$Pstatus = as.factor(student$Pstatus)
```

```
#####  
#####  
# 7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 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")
```

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

```
# we may wanna use the numerical values in various regression models
```

```
student$Medu = as.integer(student$Medu)
```

```
#####
```

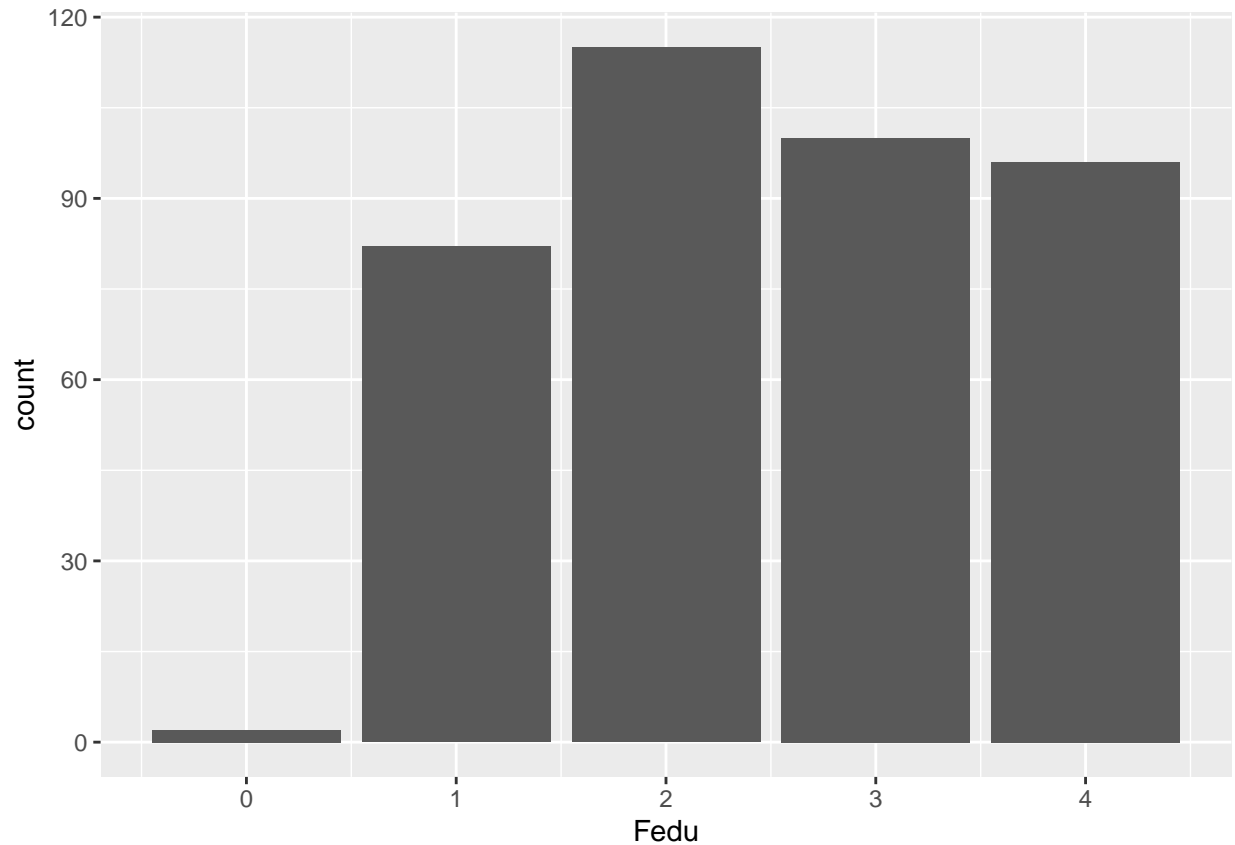
```
#####
```

```
# 8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 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")
```

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

```
# we may wanna use the numerical values in various regression models
```

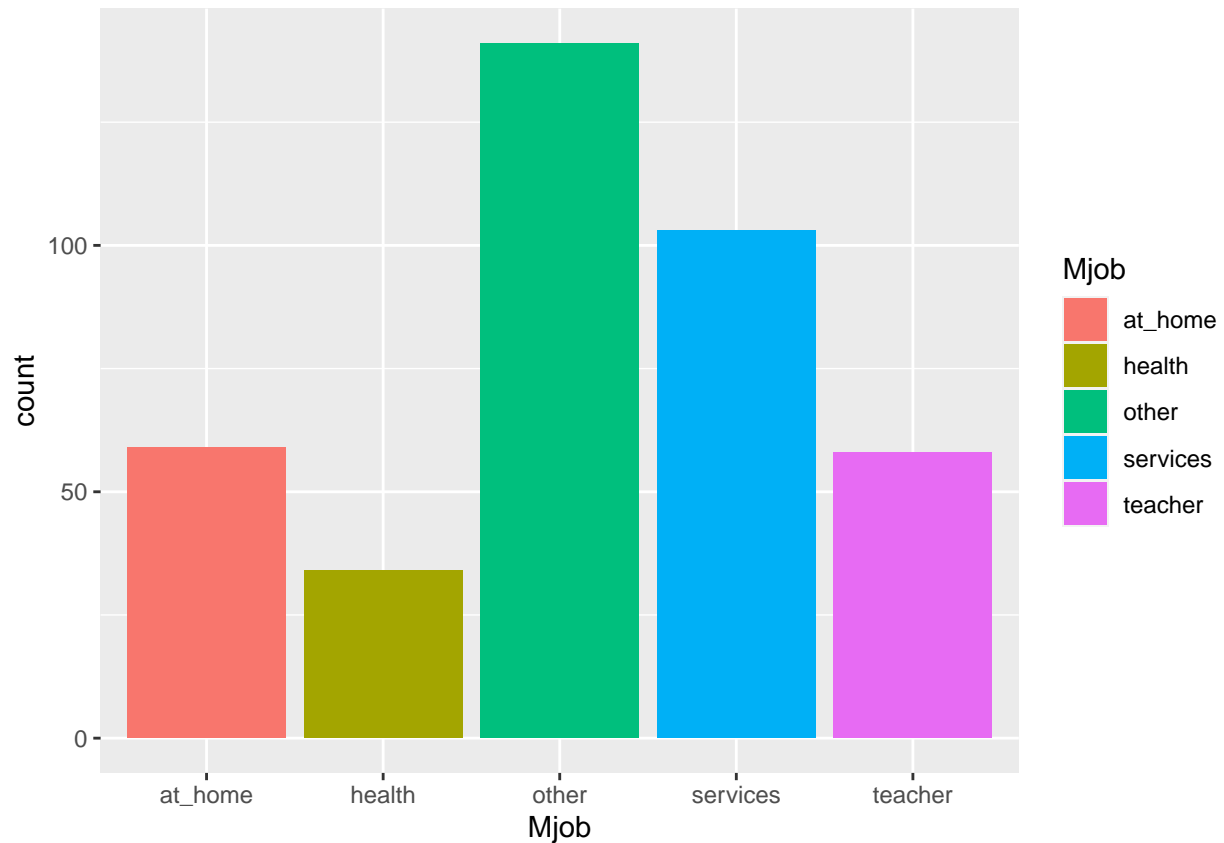
```
student$Fedu = as.integer(student$Fedu)
```

```
#####  
#####  
# 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")
```

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

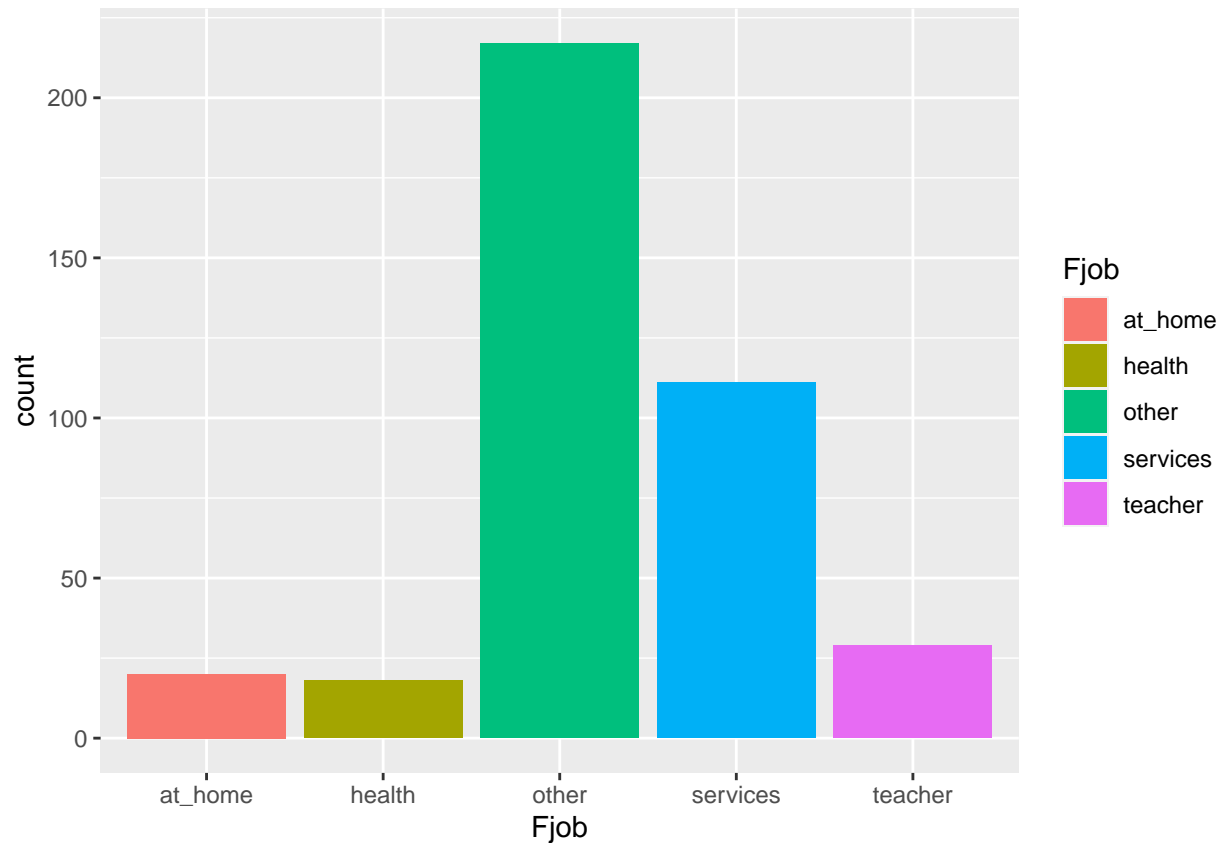
```
student$Mjob = as.factor(student$Mjob)
```

```
#####  
#####  
# 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")
```

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

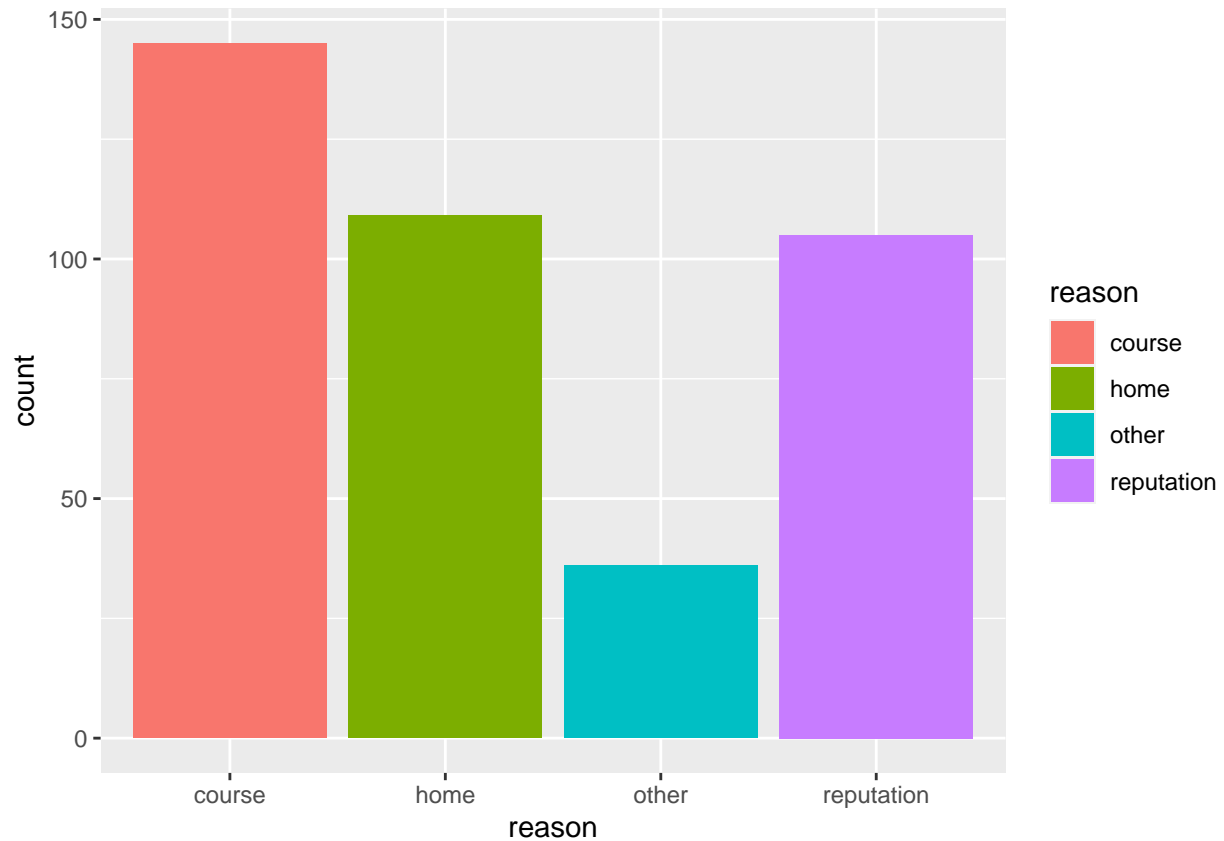
```
student$Fjob = as.factor(student$Fjob)
```

```
#####
#####
# 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")
```

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

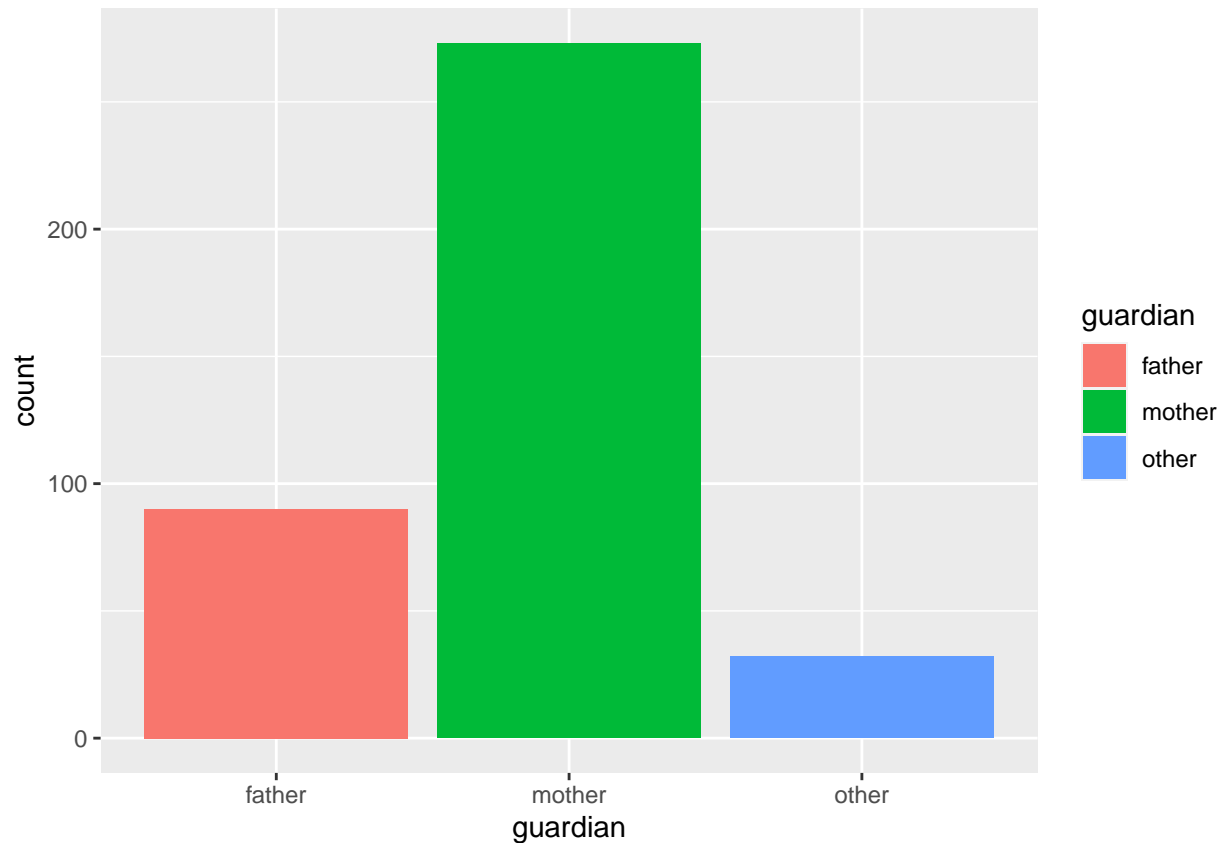
```
student$reason = as.factor(student$reason)
```

```
#####
#####
# 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")
```

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

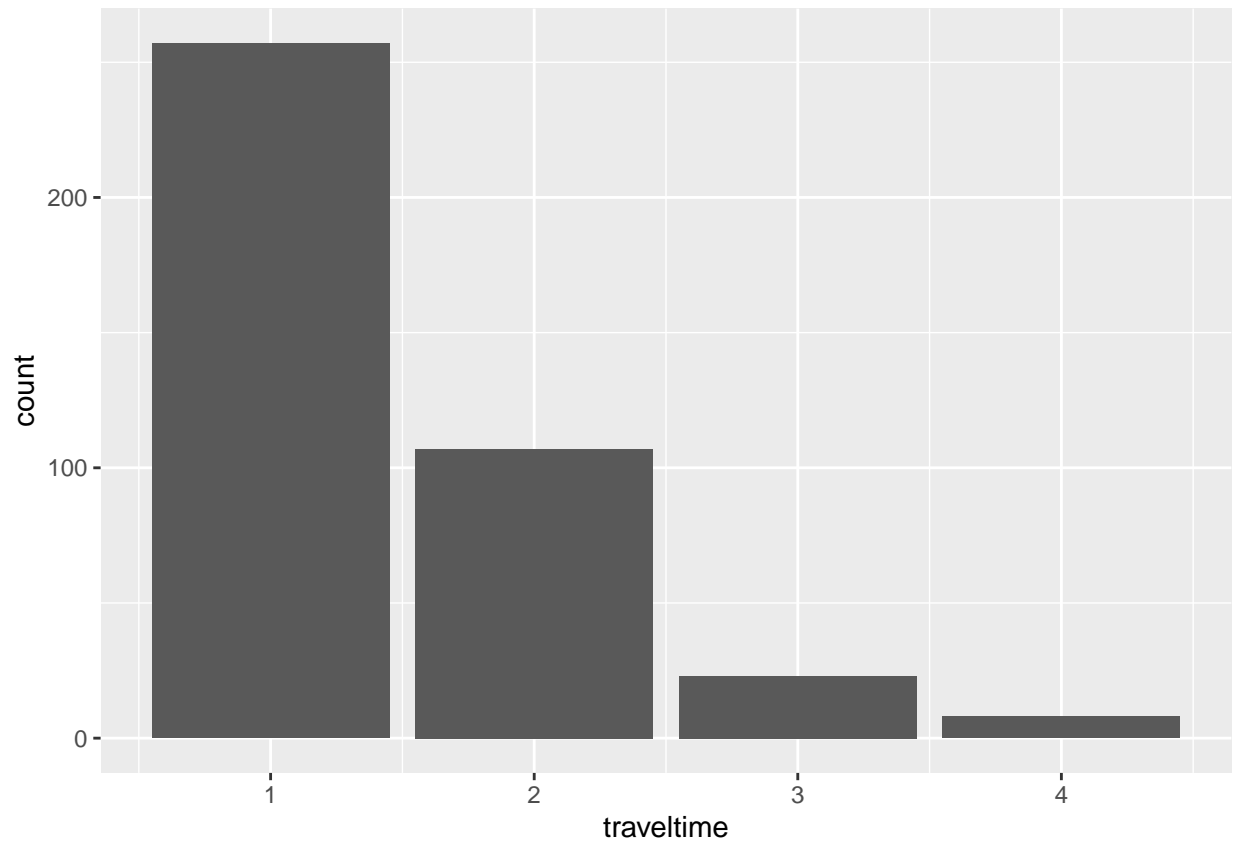
```
student$guardian = as.factor(student$guardian)
```

```
#####
#####
# 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")
```

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

```
# we may wanna use the NUMERICAL VALUES :
```

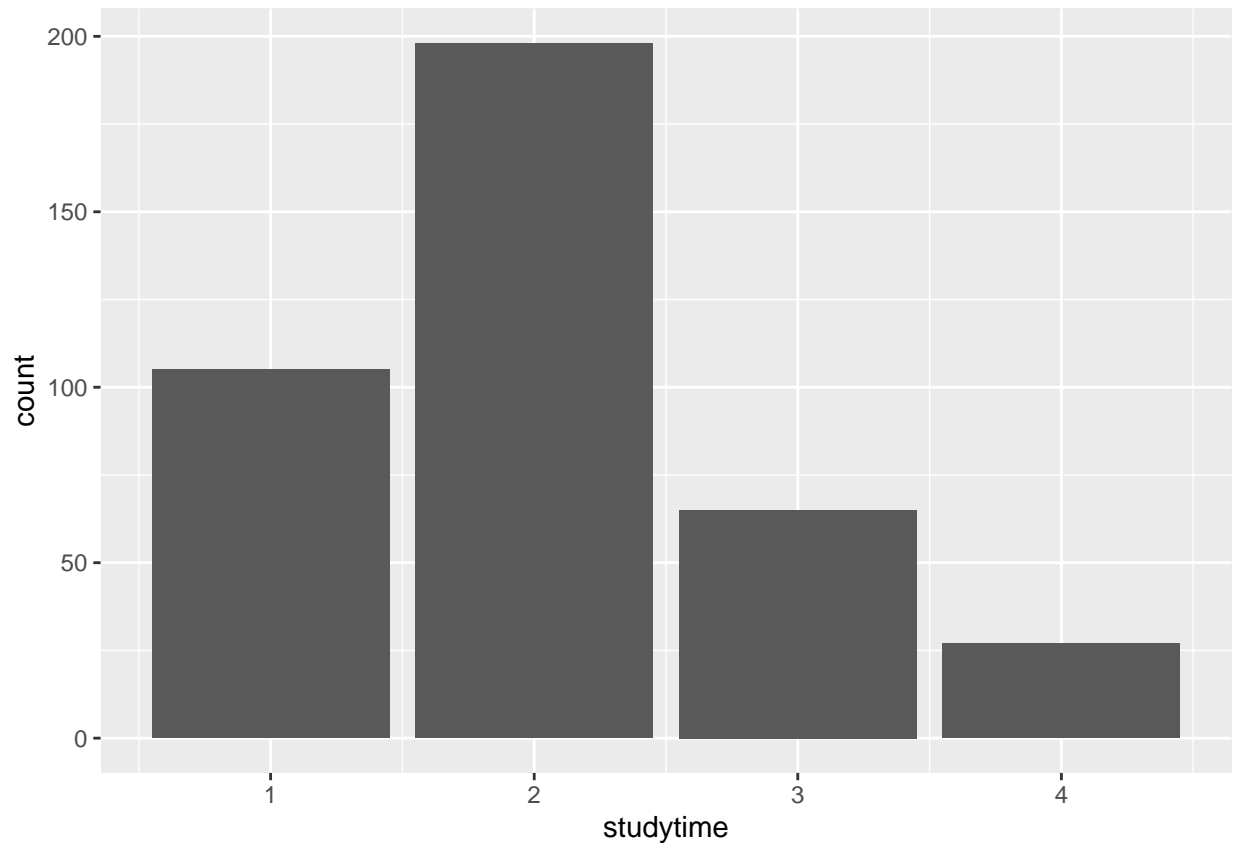
```
student$traveltime = as.integer(student$traveltime)
```

```
#####  
#####  
# 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")
```

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

```
# we may wanna use the NUMERICAL VALUES :
```

```
student$studytime = as.integer(student$studytime)
```

```
#####
```

```
#####
```

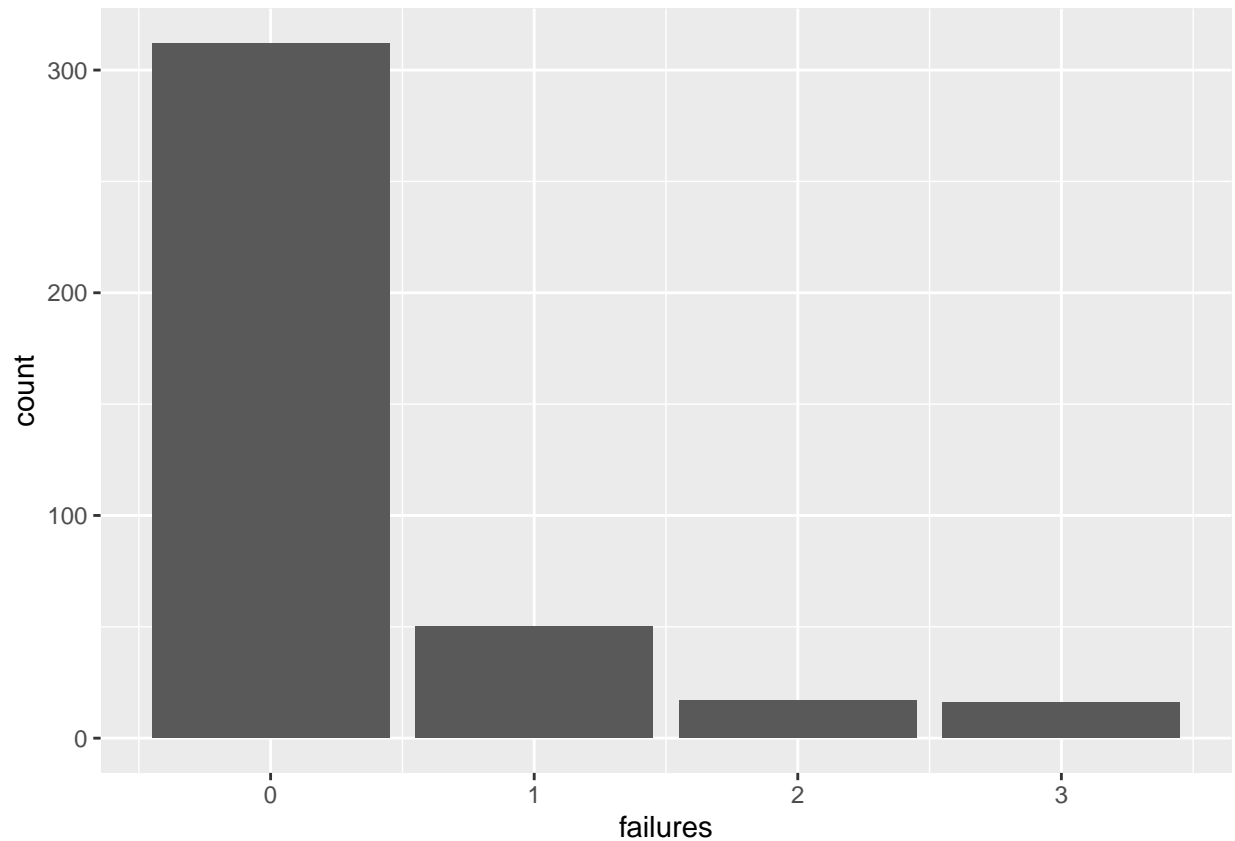
```
# 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)
```

```
unique(student$failures)
```

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

```
ggplot(data = student) +
```

```
  geom_bar(mapping = aes(x=failures, fill=failures))
```



```
ggsave("display.15.failures.png")
```

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

```
# we may wanna use the NUMERICAL VALUES :
```

```
student$failures = as.integer(student$failures)
```

```
#####
```

```
#####
```

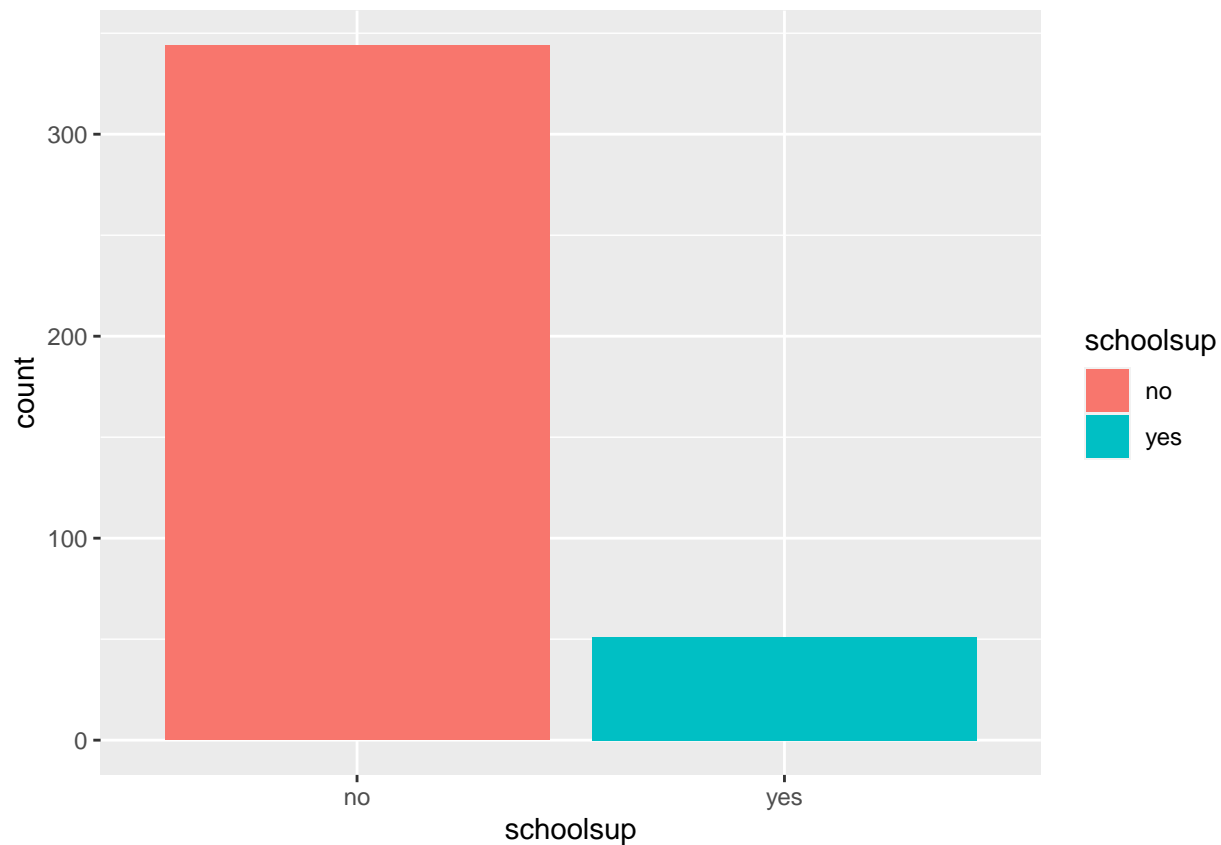
```
# 16 schoolsup - extra educational support (binary: yes or no)
```

```
unique(student$schoolsup)
```

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

```
ggplot(data = student) +
```

```
  geom_bar(mapping = aes(x=schoolsup, fill=schoolsup))
```



```
ggsave("display.16.schoolsup.png")
```

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

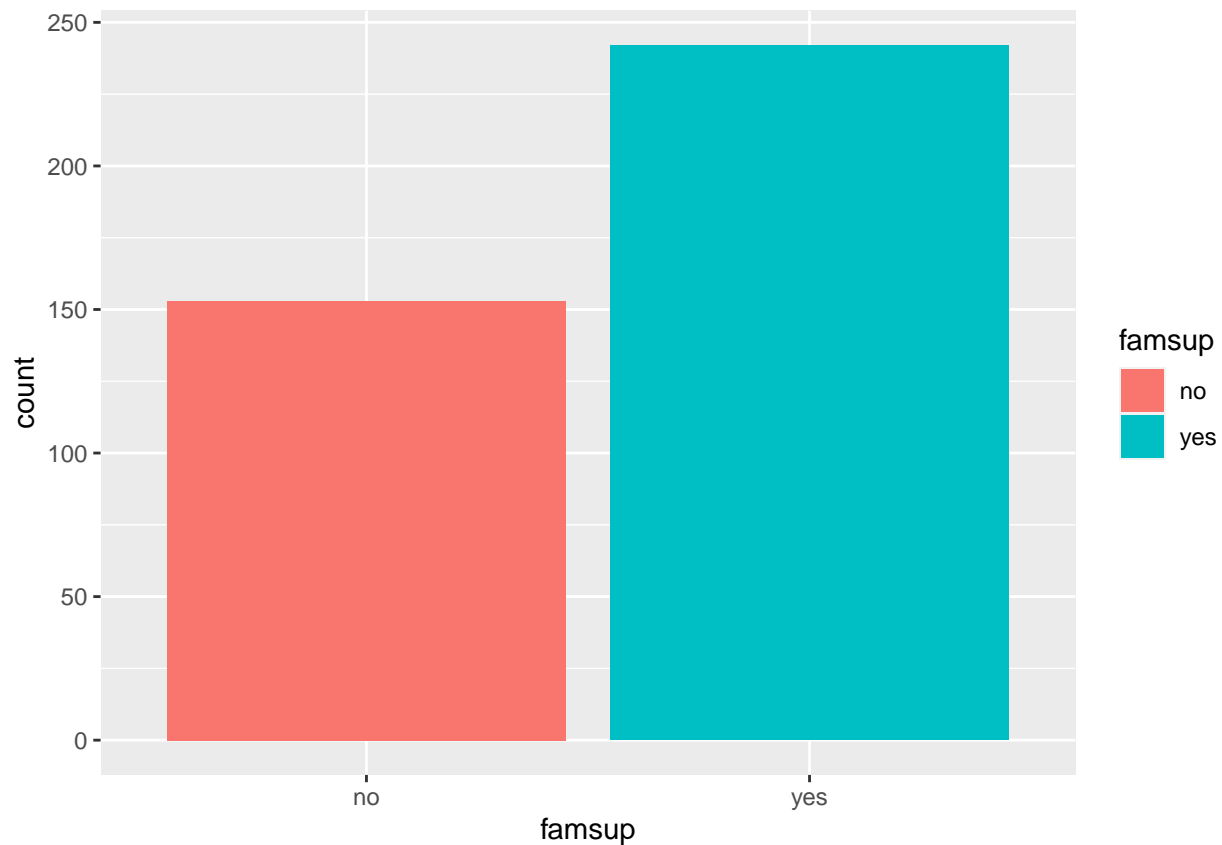
```
student$schoolsup = as.factor(student$schoolsup)
```

```
#####  
#####  
# 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")
```

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

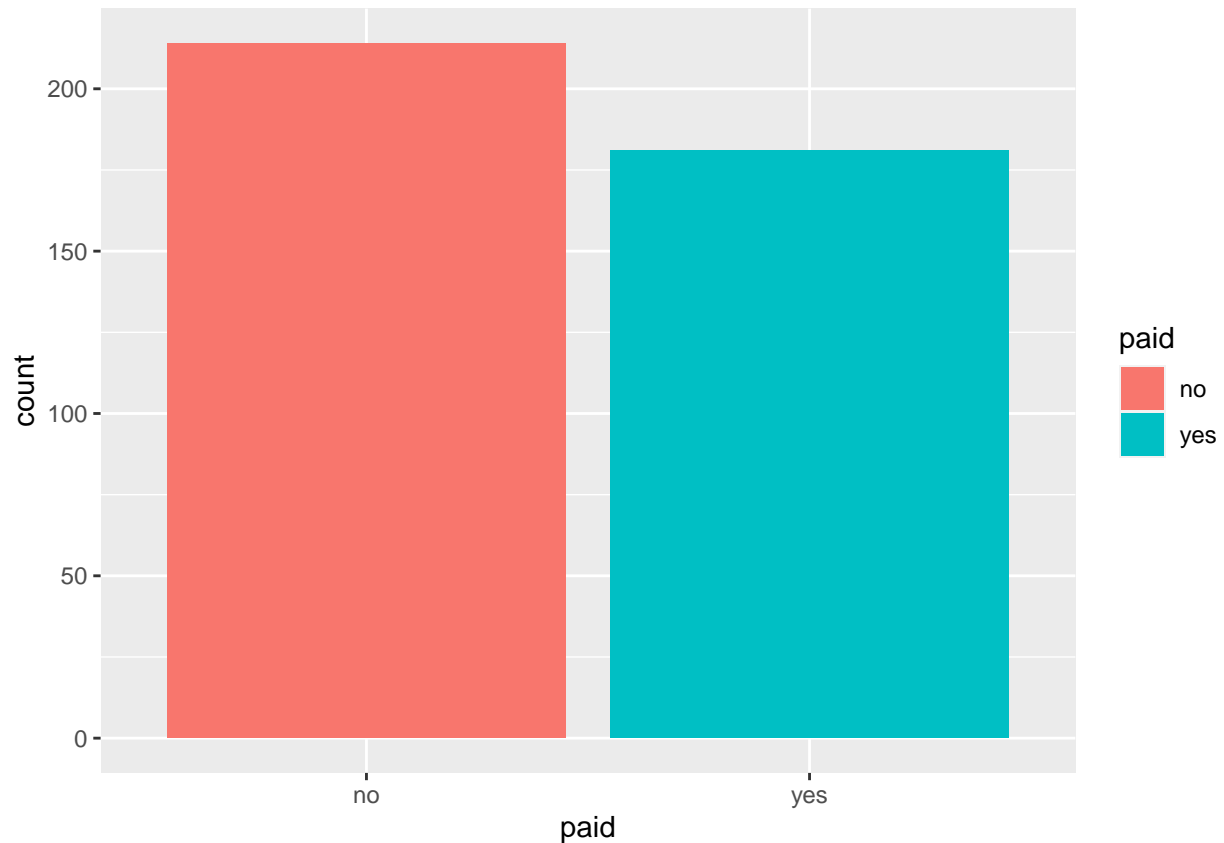
```
student$famsup = as.factor(student$famsup)
```

```
#####  
#####  
# 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")
```

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

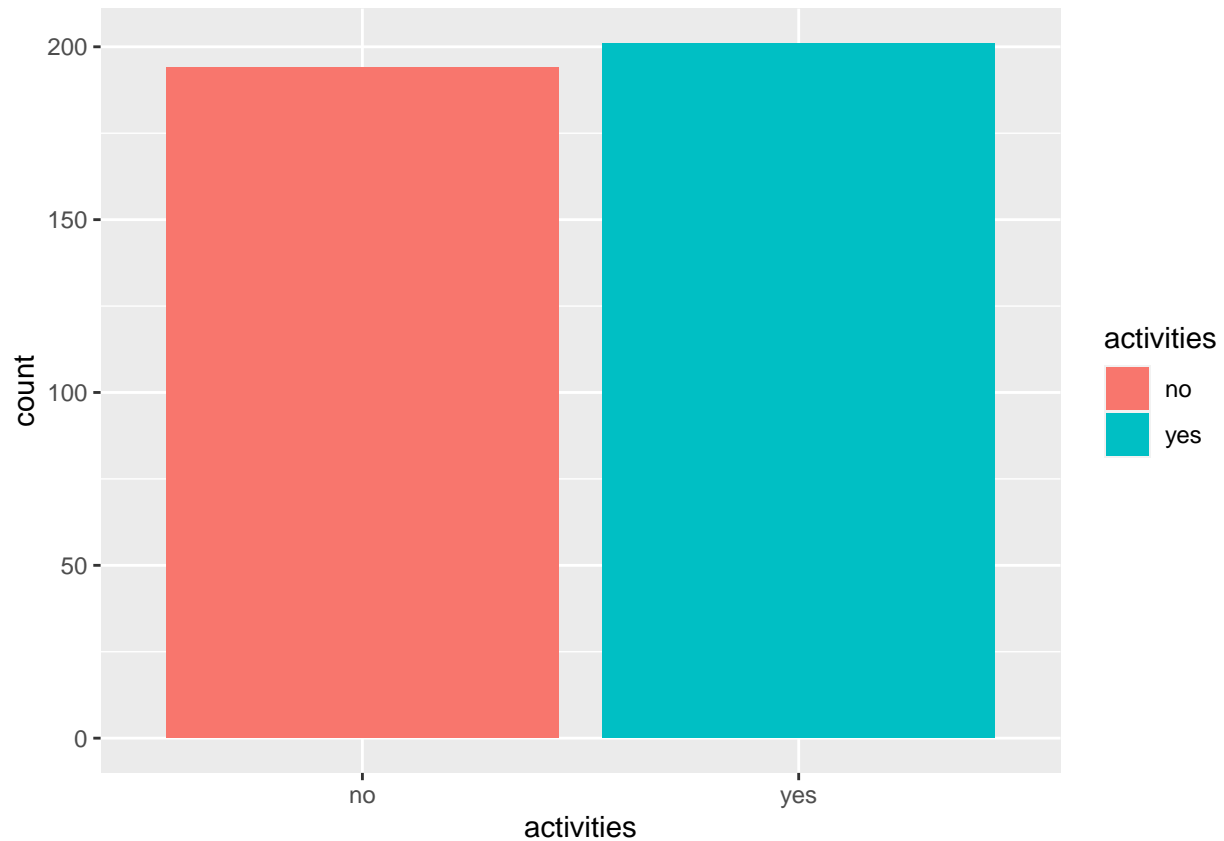
```
student$paid = as.factor(student$paid)
```

```
#####
#####
# 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")
```

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

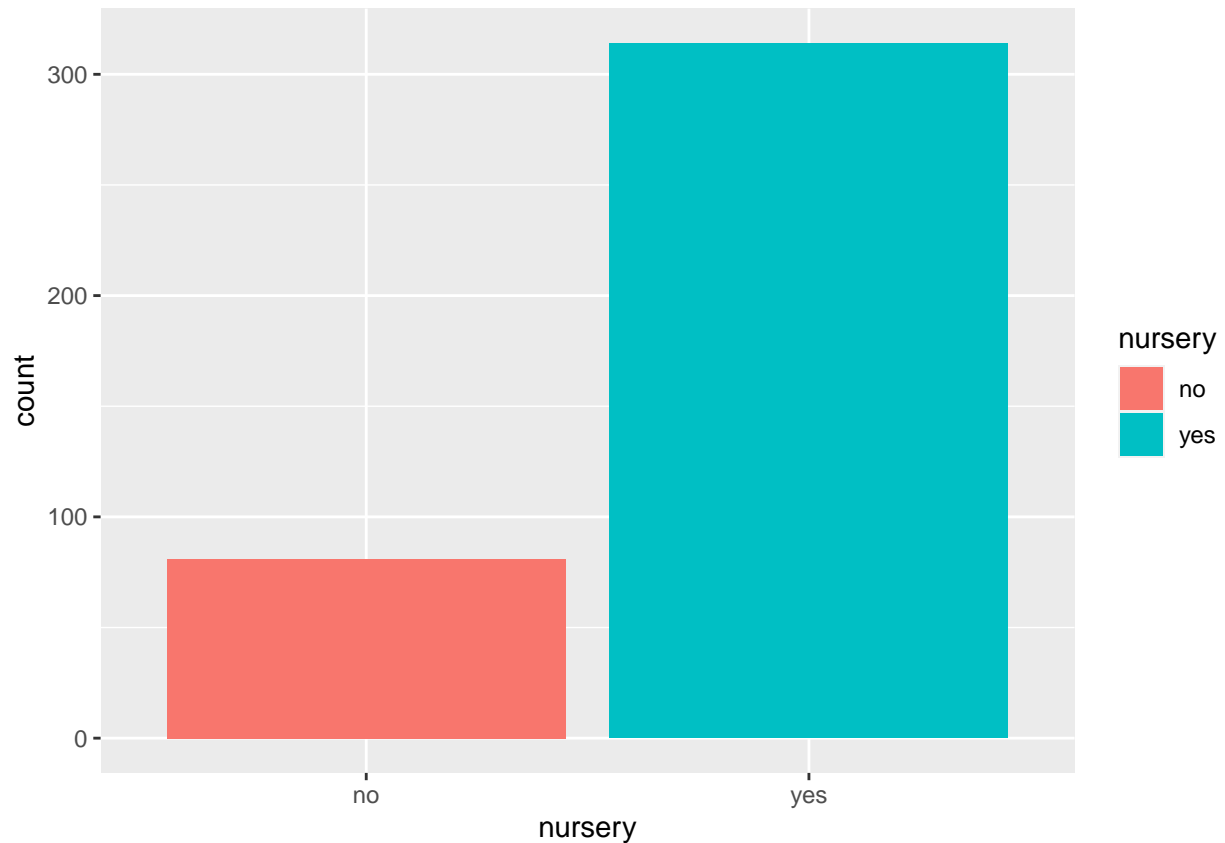
```
student$activities = as.factor(student$activities)
```

```
#####
#####
# 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")
```

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

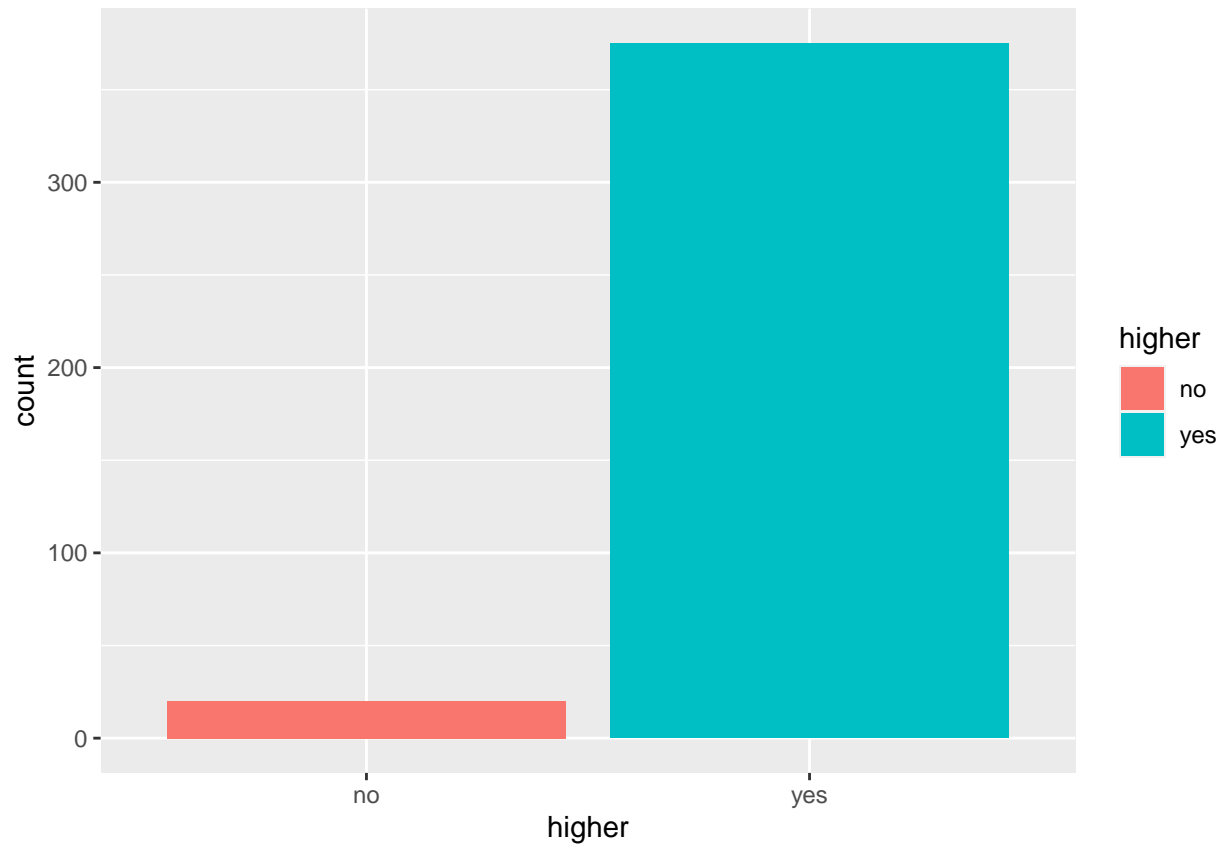
```
student$nursery = as.factor(student$nursery)
```

```
#####
#####
# 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")
```

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

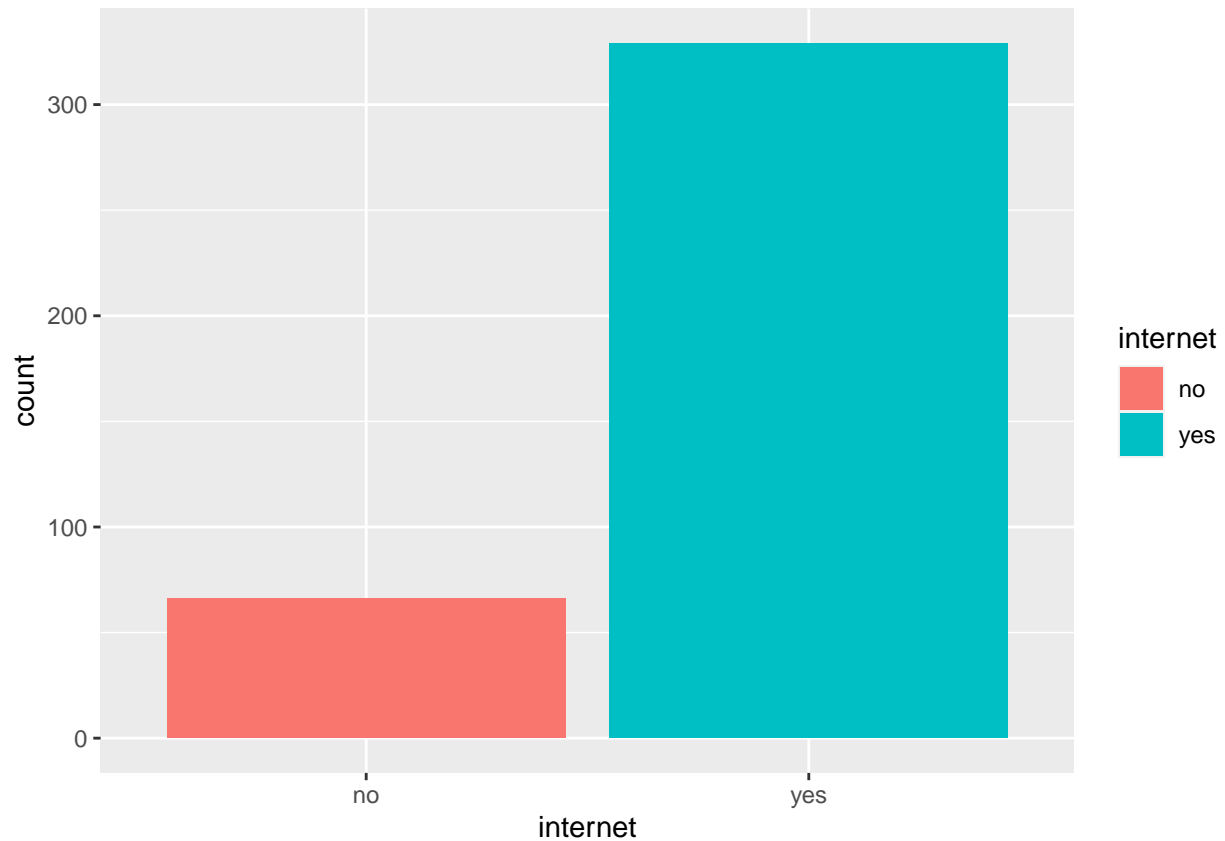
```
student$higher = as.factor(student$higher)
```

```
#####  
#####  
# 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")
```

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

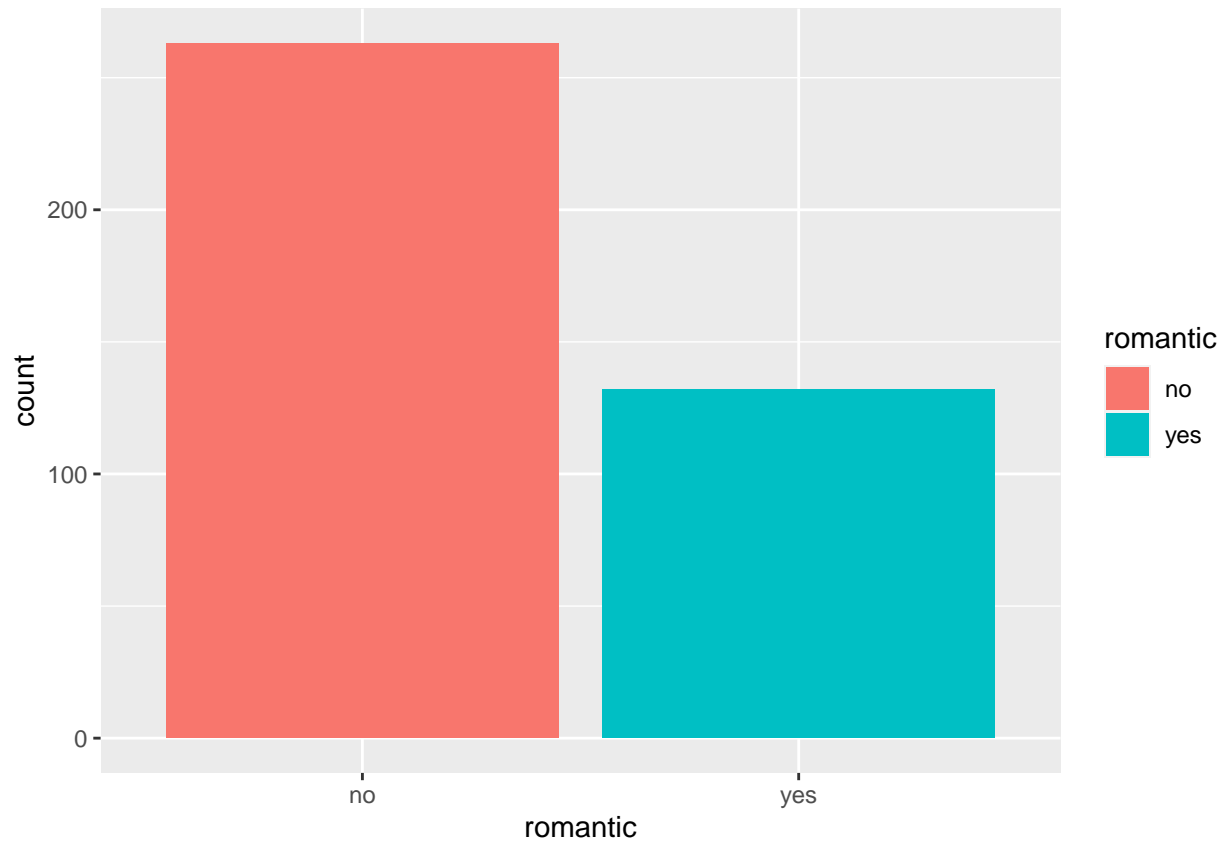
```
student$internet = as.factor(student$internet)
```

```
#####  
#####  
# 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")
```

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

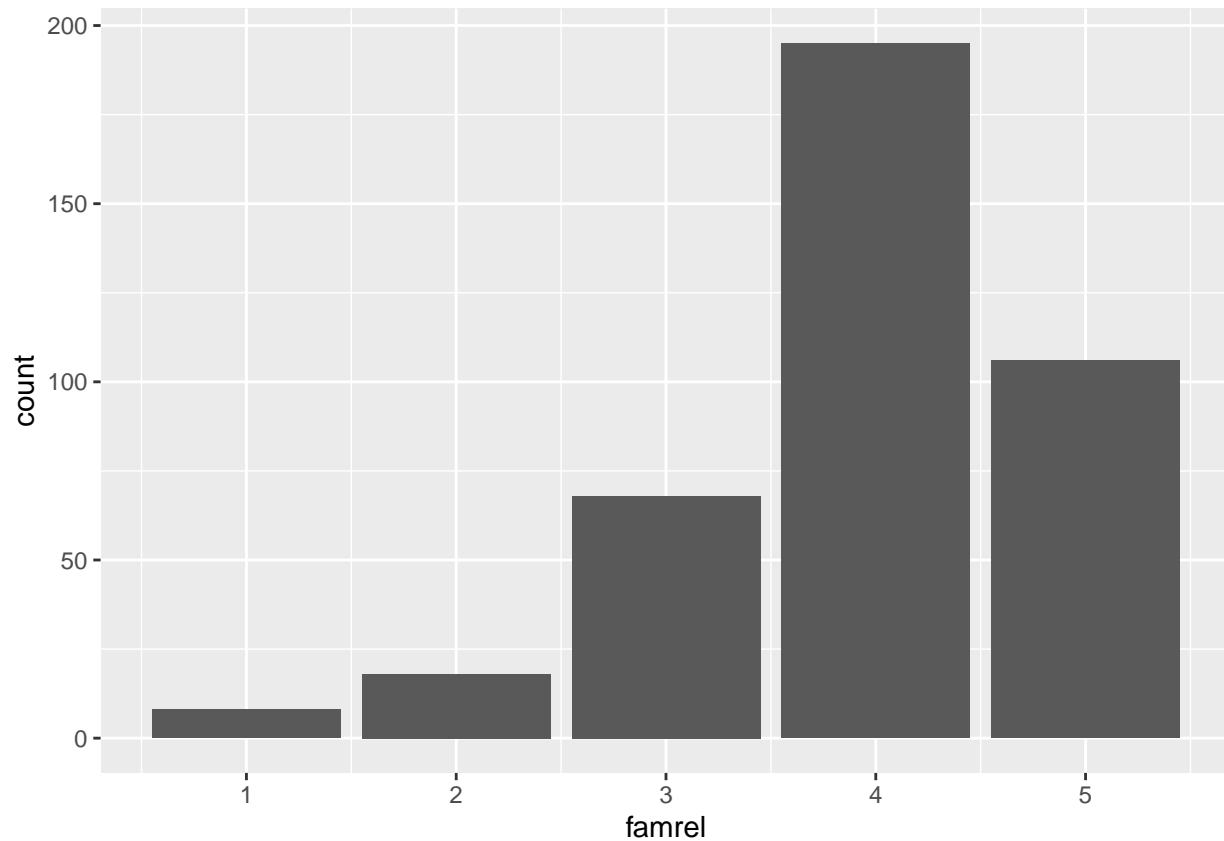
```
student$romantic = as.factor(student$romantic)
```

```
#####  
#####  
# 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")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$famrel = as.integer(student$famrel)
```

```
#####
```

```
#####
```

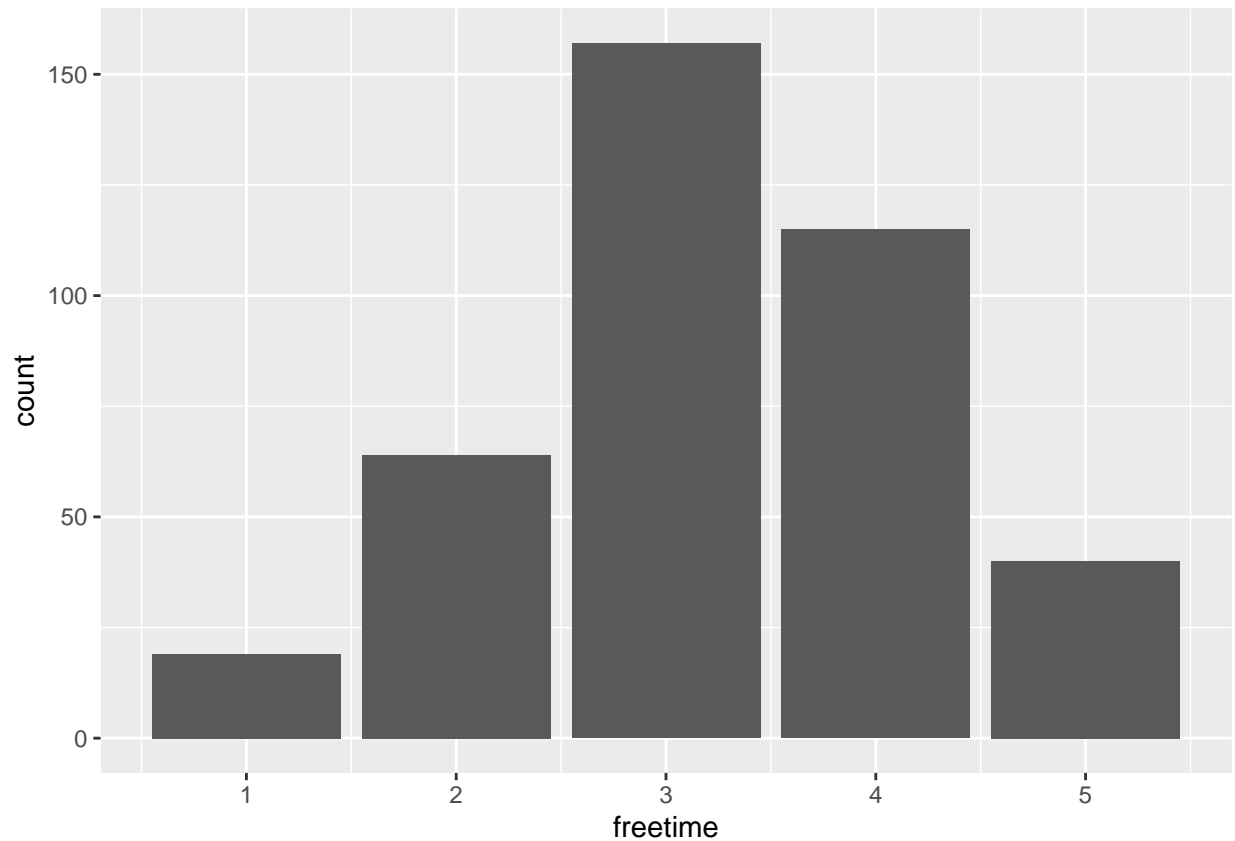
```
# 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
```

```
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")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$freetime = as.integer(student$freetime)
```

```
#####
```

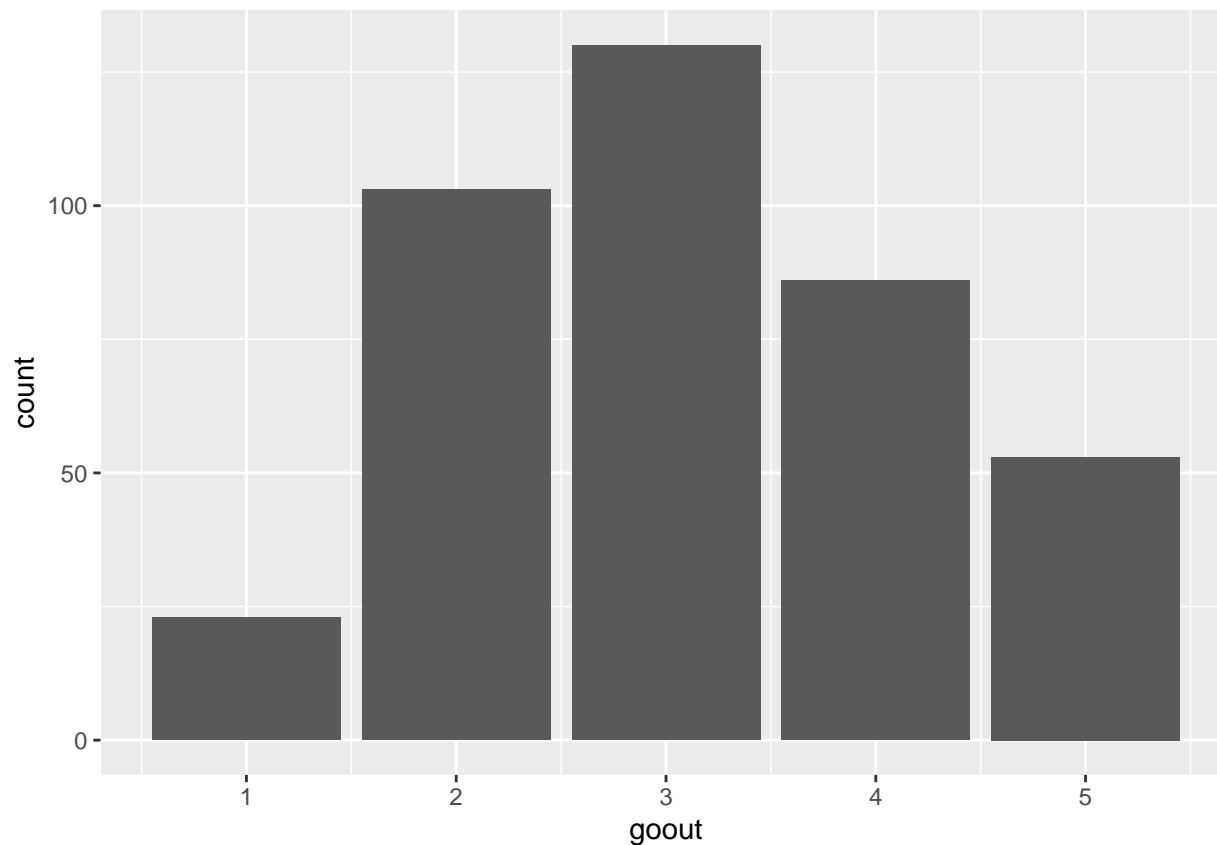
```
#####
```

```
# 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")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$goout = as.integer(student$goout)
```

```
#####
```

```
#####
```

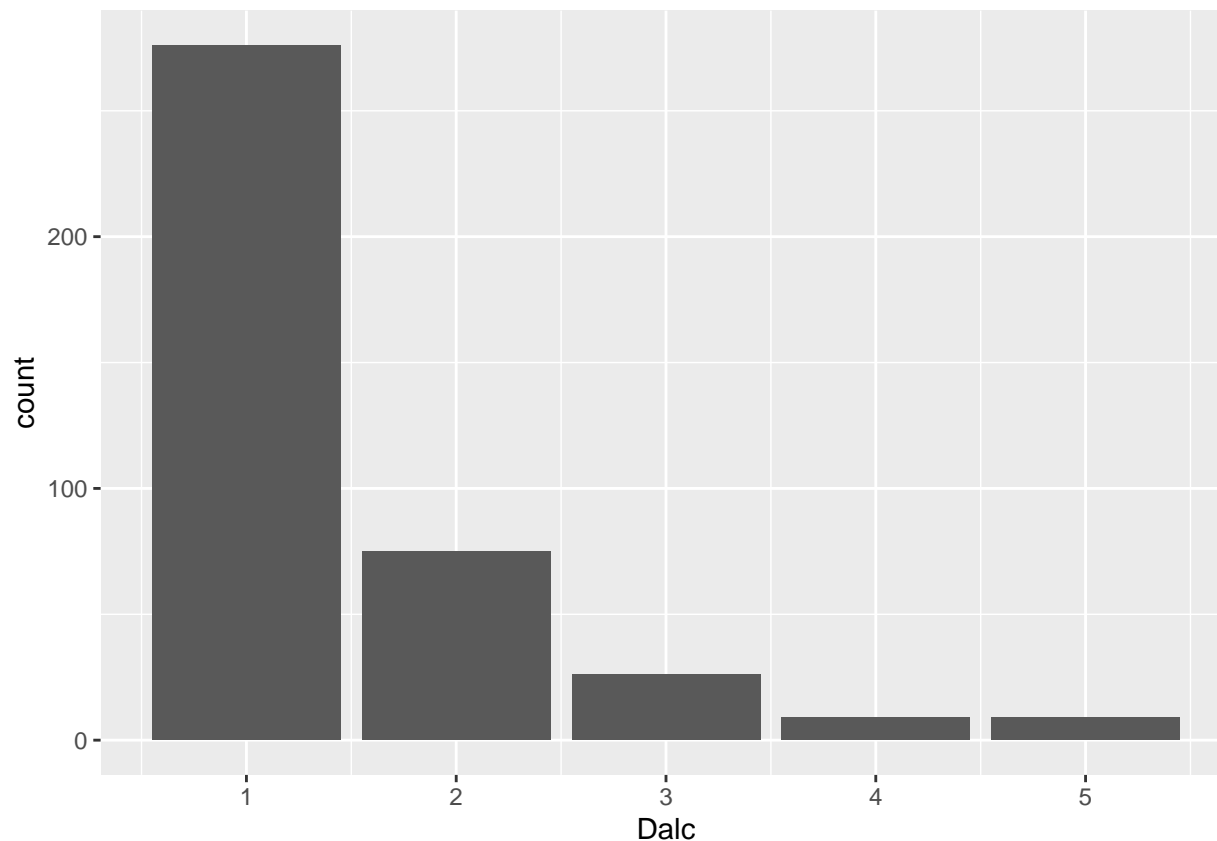
```
# 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
```

```
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")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$Dalc = as.integer(student$Dalc)
```

```
#####
```

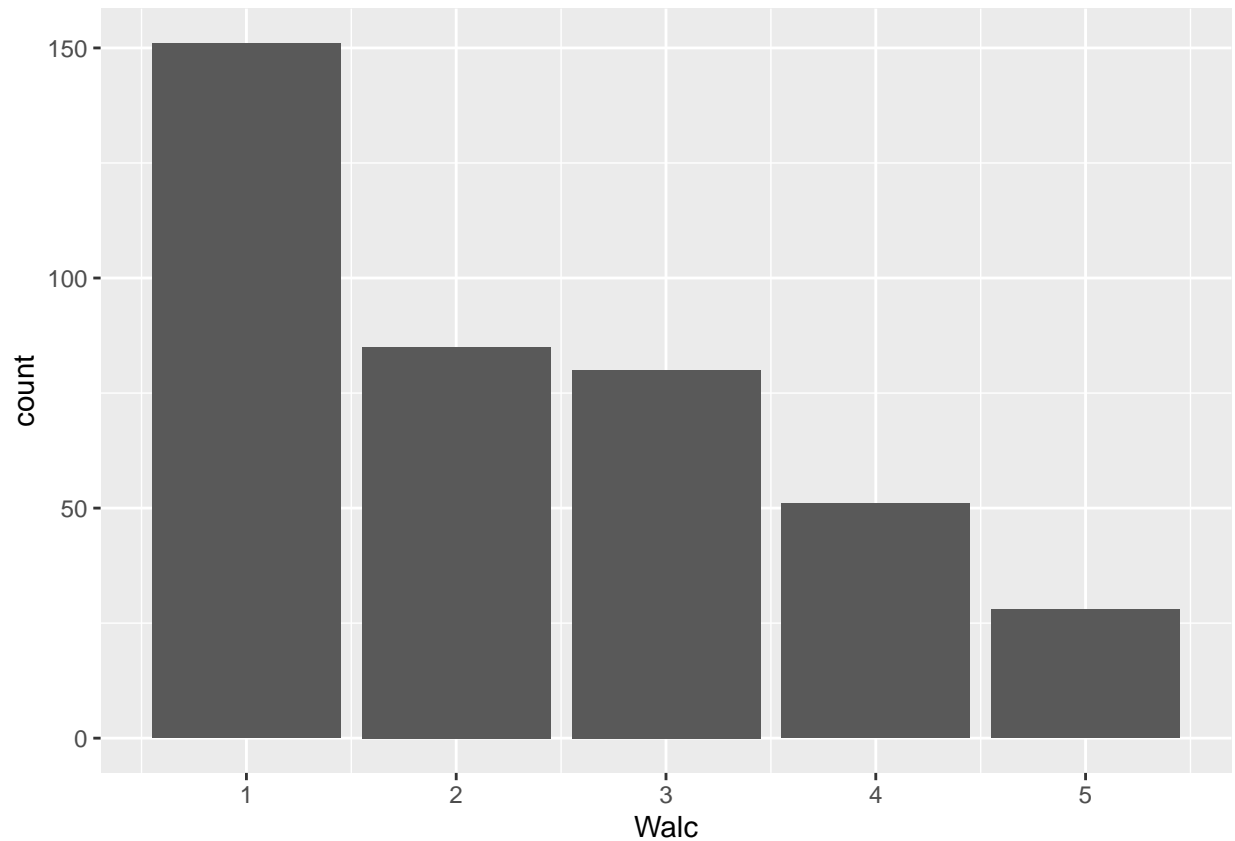
```
#####
```

```
# 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")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$Walc = as.integer(student$Walc)
```

```
#####
```

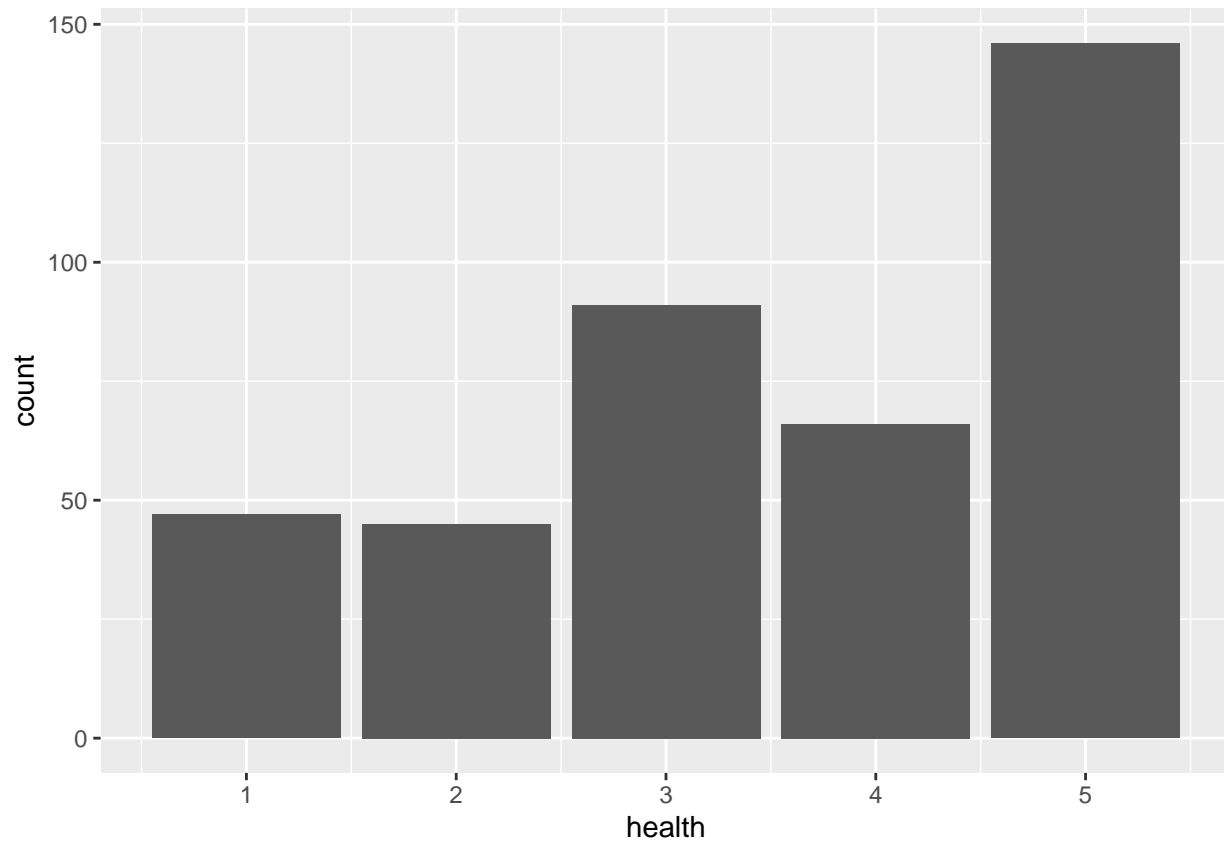
```
#####
```

```
# 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")
```

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

```
# 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)
```

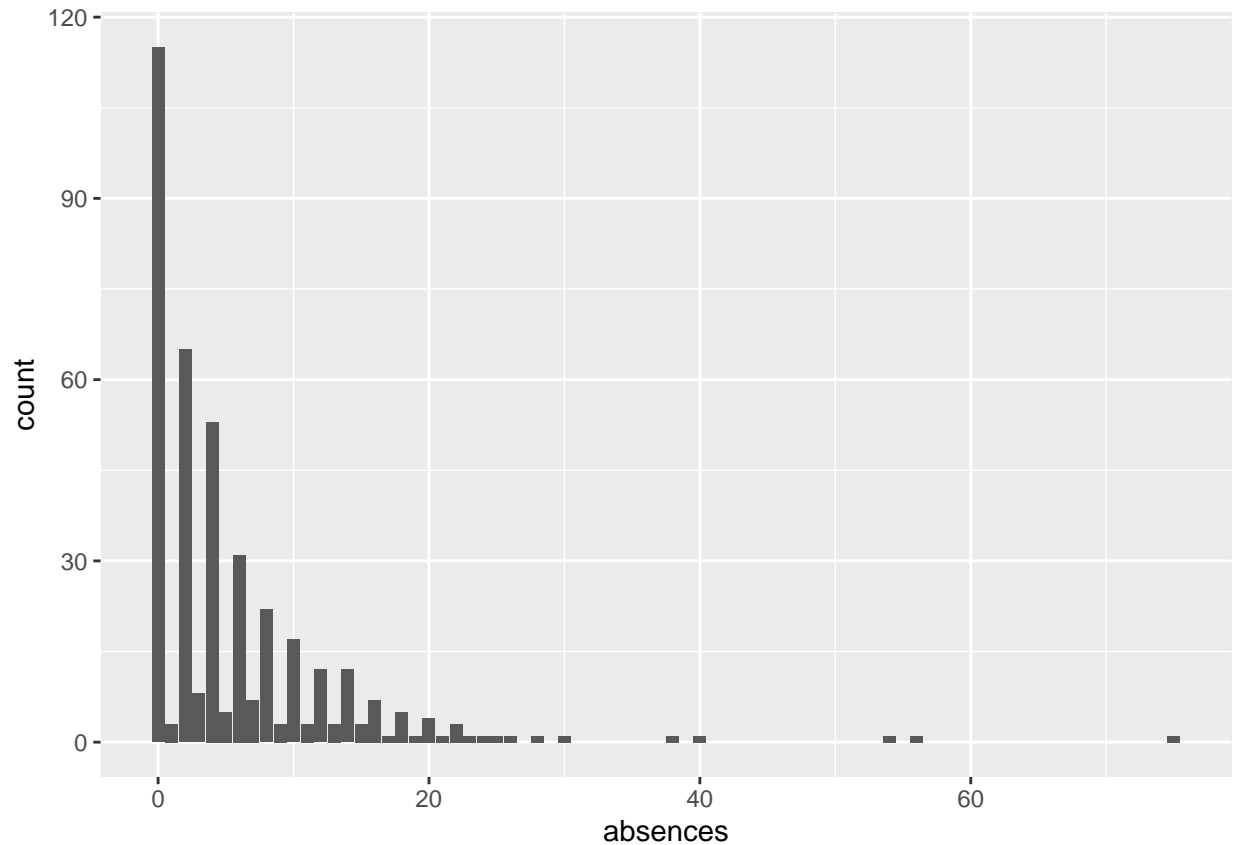
```
unique(student$absences)
```

```
## [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))
```



```
ggsave("display.30.absences.png")
```

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

```
# i believe that we can keep these as numerical :
```

```
student$absences = as.integer(student$absences)
```

```
#####
```

```
#####
```

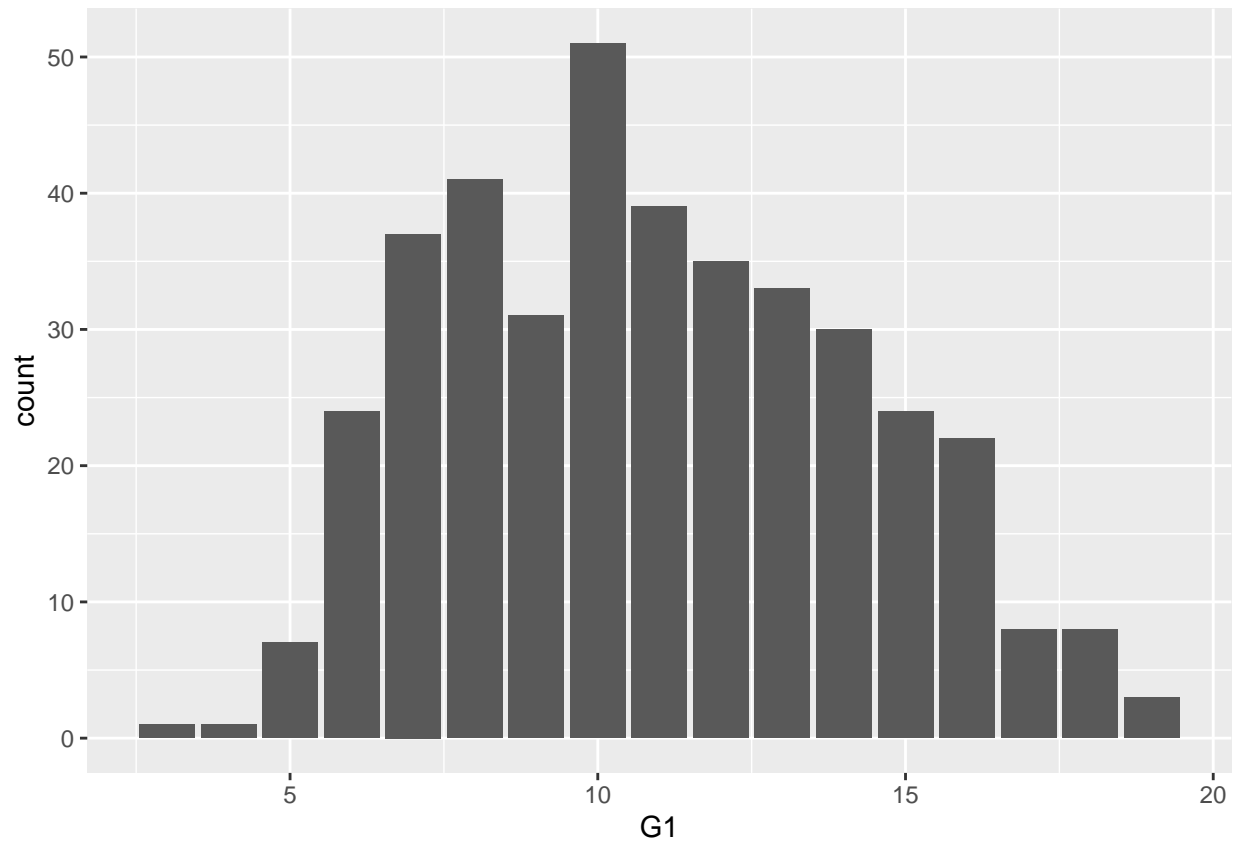
```
# $ G1 : int 5 5 7 15 6 15 12 6 16 14 ...
```

```
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))
```



```
ggsave("display.0.G1.png")
```

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

```
# i believe that we can keep these as numerical, although we may not need it :
```

```
student$G1 = as.factor(student$G1)
```

```
#####
```

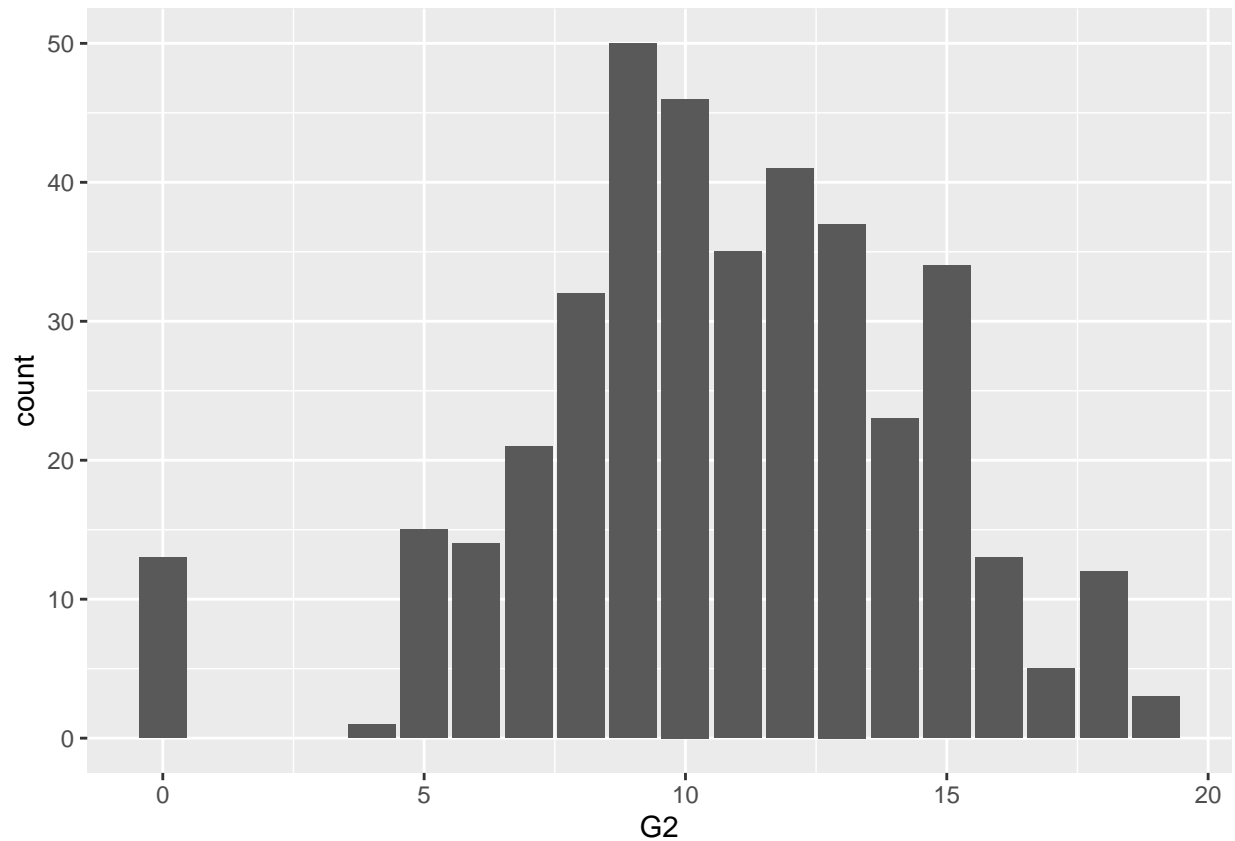
```
#####
```

```
# $ 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))
```



```
ggsave("display.0.G2.png")
```

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

```
# i believe that we can keep these as numerical, although we may not need it :
```

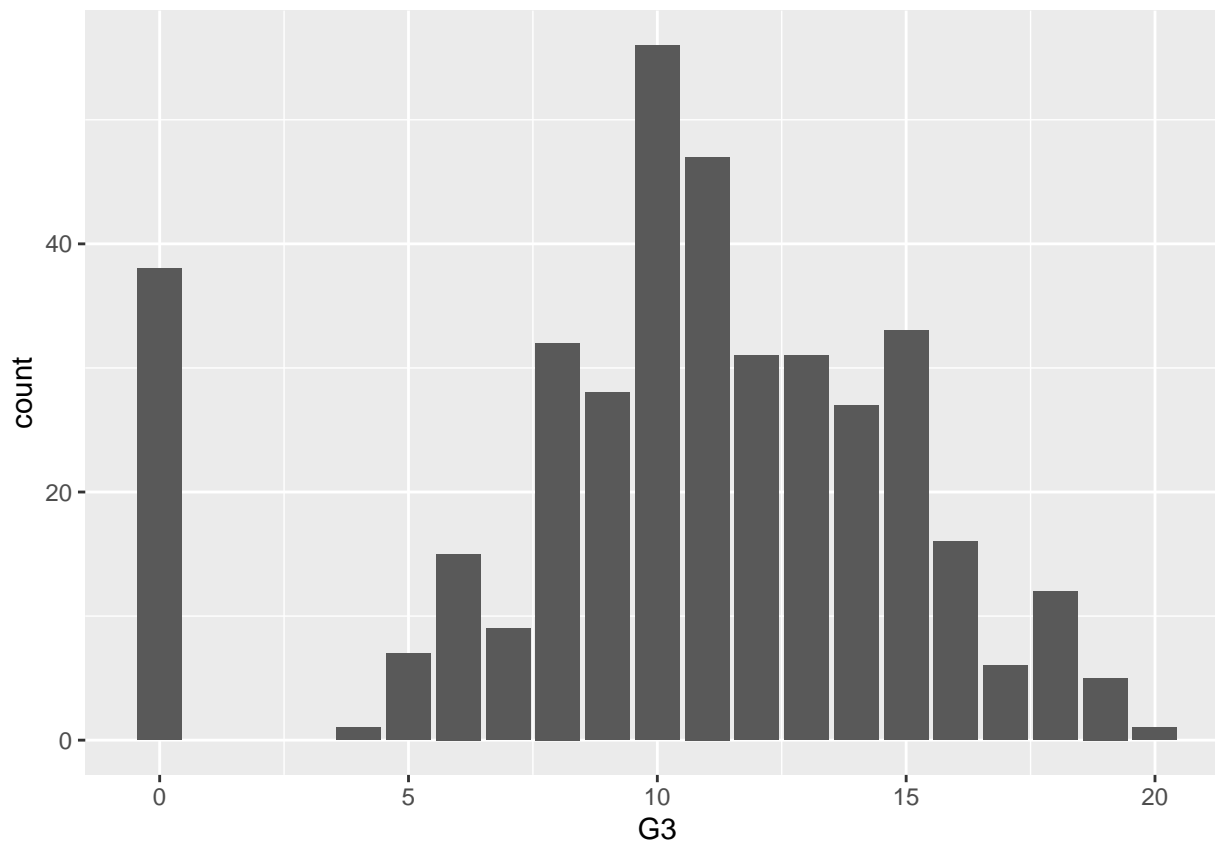
```
student$G2 = as.factor(student$G2)
```

```
#####
#####
# $ 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))
```



```
ggsave("display.0.G3.png")
```

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

```
# 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 ...
```

```
class(student)
```

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

3. DATA FILTERING

```
## the OUTPUT VARIABLES is G3
## we may remove G1 and G2
## and other features that are nit 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 STUDENT3, 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 TRANSFORMATION

```
## in order to REMOVE the RECORDS where the GRADE 3 is > 2 :
## a new piece of R code (in contrast to the previous version)

dim(student3)

## [1] 395 6

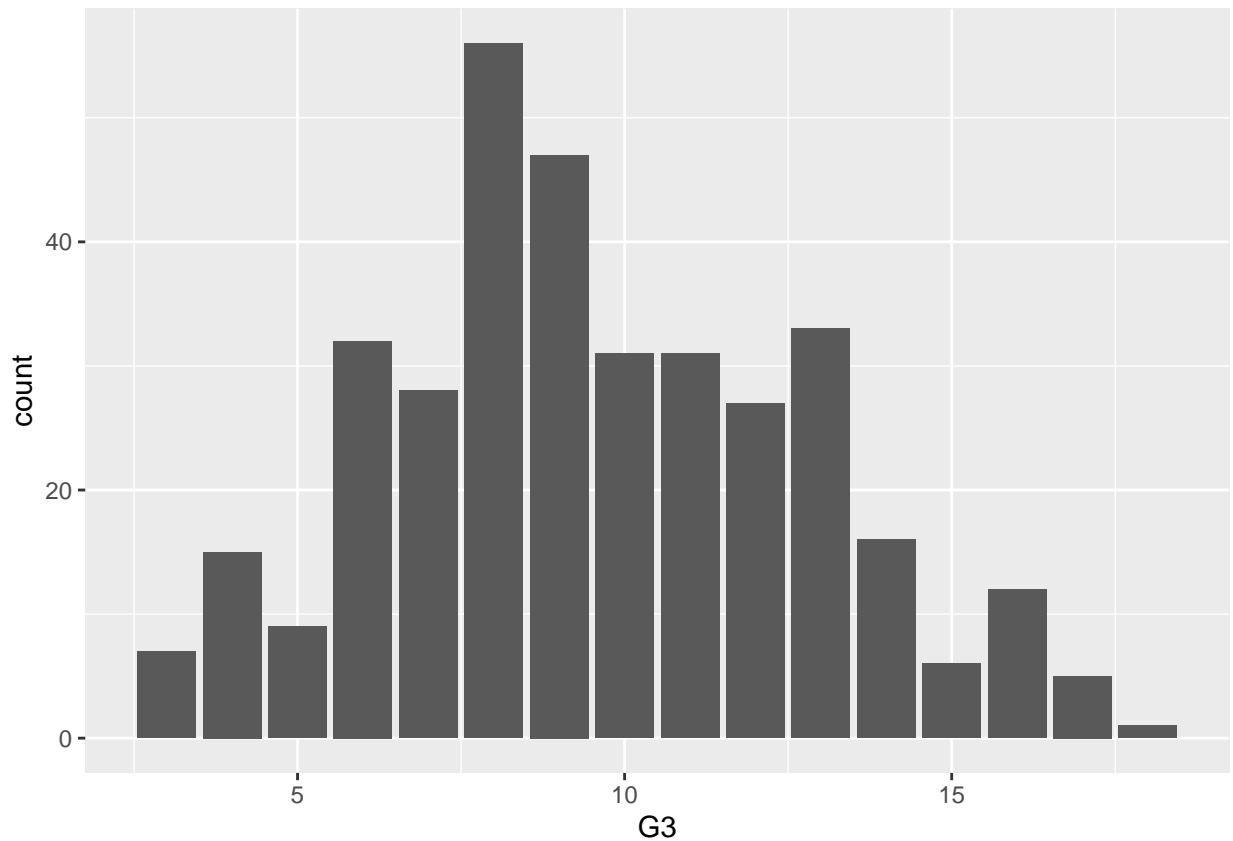
student3$G3 = as.integer(student3$G3)

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.png")
```

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

```
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)
```

5. TRAINING AND TEST SETS

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

```
indxTrain <- createDataPartition(student3$RESULT,  
                                  p = .75,  
                                  list = FALSE)
```

```

training <- student3[indxTrain,]
#training

testing <- student3[-indxTrain,]
#testing

dim(student3)

## [1] 356  6
dim(training)

## [1] 268  6
dim(testing)

## [1] 88  6

```

6. PRE-PROCESSING THE DATA

```

## PRE-PROCESSING the DATA

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

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

preProcValues

## Created from 268 samples and 5 variables
##
## Pre-processing:
##   - centered (5)
##   - ignored (0)
##   - scaled (5)
names(trainX)

## [1] "age"          "traveltime" "studytime"  "failures"   "absences"
dim(trainX)

## [1] 268  5
names(training)

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

```

7. PERFORMING THE TRAINING

```

## PERFORMING the TRAINING

```

```

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

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

## The output of kNN fit

knnFit

## k-Nearest Neighbors
##
## 268 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: 241, 242, 240, 241, 242, 241, ...
## Resampling results across tuning parameters:
##
##  k   Accuracy   Kappa
##  5  0.5772182  0.14485215
##  7  0.5764516  0.13982731
##  9  0.5549349  0.09877736
## 11  0.5763058  0.14289109
## 13  0.5812915  0.15306078
## 15  0.5774895  0.14440753
## 17  0.5725139  0.13257052
## 19  0.5698514  0.12495619
## 21  0.5759293  0.13846462
## 23  0.5908493  0.16813866
## 25  0.5996337  0.18736077
## 27  0.6008683  0.18920423
## 29  0.6133598  0.21357071
## 31  0.6058167  0.19503496
## 33  0.5936440  0.17130745
## 35  0.5961131  0.17677884
## 37  0.5860942  0.15703032
## 39  0.5911850  0.16751914
## 41  0.5873898  0.15947133
## 43  0.5898996  0.16166239
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 29.

png("the.results.knn.FIT.png")
plot(knnFit)
dev.off()

## pdf
## 2

```

8. MAKING THE PREDICTIONS

```
## Making the PREDICTIONS :  
  
knnPredict <- predict(knnFit, newdata = testing)
```

9. THE CONFUSION MATRIX

```
## COMPUTING the CONFUSION MATRIX :  
  
confusionMatrix(knnPredict, testing$RESULT)  
  
## Confusion Matrix and Statistics  
##  
##           Reference  
## Prediction NO_PASS PASS  
##    NO_PASS      33   18  
##    PASS       15   22  
##  
##           Accuracy : 0.625  
##           95% CI : (0.5153, 0.726)  
##    No Information Rate : 0.5455  
##    P-Value [Acc > NIR] : 0.08137  
##  
##           Kappa : 0.239  
##  
##    Mcnemar's Test P-Value : 0.72772  
##  
##           Sensitivity : 0.6875  
##           Specificity : 0.5500  
##           Pos Pred Value : 0.6471  
##           Neg Pred Value : 0.5946  
##           Prevalence : 0.5455  
##           Detection Rate : 0.3750  
##           Detection Prevalence : 0.5795  
##           Balanced Accuracy : 0.6188  
##  
##           'Positive' Class : NO_PASS  
##  
mean(knnPredict == testing$RESULT)  
  
## [1] 0.625  
  
dim(student3)  
  
## [1] 356  6
```

We may aim to optimize the model by feature selection or by including new features from the data that is available.