

# Projet Analyse de données

Rudio et Léo-Paul

2023-05-09

## Présentation du projet et du jeu de données

Le jeu de données est constitués d'informations sur la vie d'étudiants dans une université du Portugal. Ces informations vont de leur résultats universitaires, leur vie familiale à leur consommation d'alcool. Le jeu a été construit à partir d'une enquête menée auprès d'étudiant en mathématiques et en portugais.

L'objectif serait alors d'analyser le jeu de données afin de comprendre les facteurs qui impactent la réussite scolaire de ces étudiants. L'intérêt du jeu est la grande variété de facteurs proposée qui permet de couvrir un maximum d'hypothèses, notamment celle sur la consommation d'alcool proposée directement par le nom du jeu de données.

Voici les variables présentes dans ce jeu de données ;

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

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

These grades are related with the course subject, Math or Portuguese: - **G1** - first period grade (numeric: from 0 to 20) - **G2** - second period grade (numeric: from 0 to 20) - **G3** - final grade (numeric: from 0 to 20, output target)

Au cours de ce projet, nous nous concentrons sur la variable G3 qui est la variable de sortie représentant la note finale des élèves. Il s'agirait donc d'un problème de régression sur la variables G3 ou même plus généralement un problème de classification.

Voici les étapes que nous allons suivre :

1. Identifier les variables significatives
2. Appliquer des méthodes de classification sur la réussite scolaire
3. Effectuer une regression linéaires pour prédire G3
4. Comparer des méthodes de machine learning pour prédire G3

## 1.Chargement des données

```
# Chargement de la base de données
df.mat=read.table("student-mat.csv",sep=";",header=TRUE,as.is = FALSE)
df.por=read.table("student-por.csv",sep=";",header=TRUE,as.is = FALSE)

# Etudiants qui appartiennent aux deux cours
both= merge(df.mat,df.por,by=c("school","sex","age","address","famsize","Pstatus","Medu","Fedu","Mjob",
                                "Fjob","reason"))

# Concaténation des deux dataframes
df = rbind(df.mat,df.por)
head(df)
```

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

```
## absences G1 G2 G3
## 1      6  5  6  6
## 2      4  5  5  6
## 3     10  7  8 10
## 4      2 15 14 15
## 5      4  6 10 10
## 6     10 15 15 15
```

## 2. Nettoyage et vérification des données

Le jeu est composé de 33 variables dont 17 qualitatives et 16 quantitatives. On calcule la moyenne pour chaque élève, et on rajoute une variable pour la réussite scolaire.

```
print(str(df))
```

```
## 'data.frame': 1044 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 : 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 ...
## NULL
```

```
print(nrow(df))
```

```
## [1] 1044
```

```

# factor
# df$famrel=factor(df$famrel)
df$Dalc=factor(df$Dalc)
df$Walc=factor(df$Walc)
# df$freetime=factor(df$freetime)
# df$Medu=factor(df$Medu)
# df$Fedu=factor(df$Fedu)
# df$straveltime=factor(df$traveltime)
# df$studytime=factor(df$studytime)
df$goout=factor(df$goout)
df$health=factor(df$health)

## On calcule la moyenne des étudiants
df$Moy = (df$G1+df$G2+df$G3)/3

## On rajoute la réussite scolaire comme variable qualitative que nous devons prédire.
# df$RS = factor(df$Moy>=10)
df$RS = "admis"
# df$RS[df$Moy<10]="admis par conseil"
df$RS[df$Moy<10]="redoublement"
df$RS[df$Moy<8.50]="exclusion"
df$RS=factor(df$RS)
head(df)

```

```

##   school sex age address famsize Pstatus Medu Fedu   Mjob   Fjob   reason
## 1    GP   F  18      U    GT3      A    4    4  at_home  teacher  course
## 2    GP   F  17      U    GT3      T    1    1  at_home   other  course
## 3    GP   F  15      U    LE3      T    1    1  at_home   other  other
## 4    GP   F  15      U    GT3      T    4    2  health  services  home
## 5    GP   F  16      U    GT3      T    3    3   other   other  home
## 6    GP   M  16      U    LE3      T    4    3  services   other reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2          2          0        yes    no   no          no
## 2   father          1          2          0        no    yes  no          no
## 3   mother          1          2          3        yes    no  yes          no
## 4   mother          1          3          0        no    yes  yes          yes
## 5   father          1          2          0        no    yes  yes          no
## 6   mother          1          2          0        no    yes  yes          yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1    yes    yes      no      no      4          3    4    1    1    3
## 2    no    yes      yes      no      5          3    3    1    1    3
## 3    yes    yes      yes      no      4          3    2    2    3    3
## 4    yes    yes      yes      yes      3          2    2    1    1    5
## 5    yes    yes      no      no      4          3    2    1    2    5
## 6    yes    yes      yes      no      5          4    2    1    2    5
##   absences G1 G2 G3      Moy      RS
## 1      6  5  6  6  5.666667  exclusion
## 2      4  5  5  6  5.333333  exclusion
## 3     10  7  8 10  8.333333  exclusion
## 4      2 15 14 15 14.666667    admis
## 5      4  6 10 10  8.666667 redoublement
## 6     10 15 15 15 15.000000    admis

```

### 3. Exploration des données : études des variables

Cette partie consiste à appliquer des méthodes de statistiques descriptives afin de mieux comprendre le jeu de données. On se concentre sur l'analyse de la distribution des variables et leur corrélation avec les résultats scolaires.

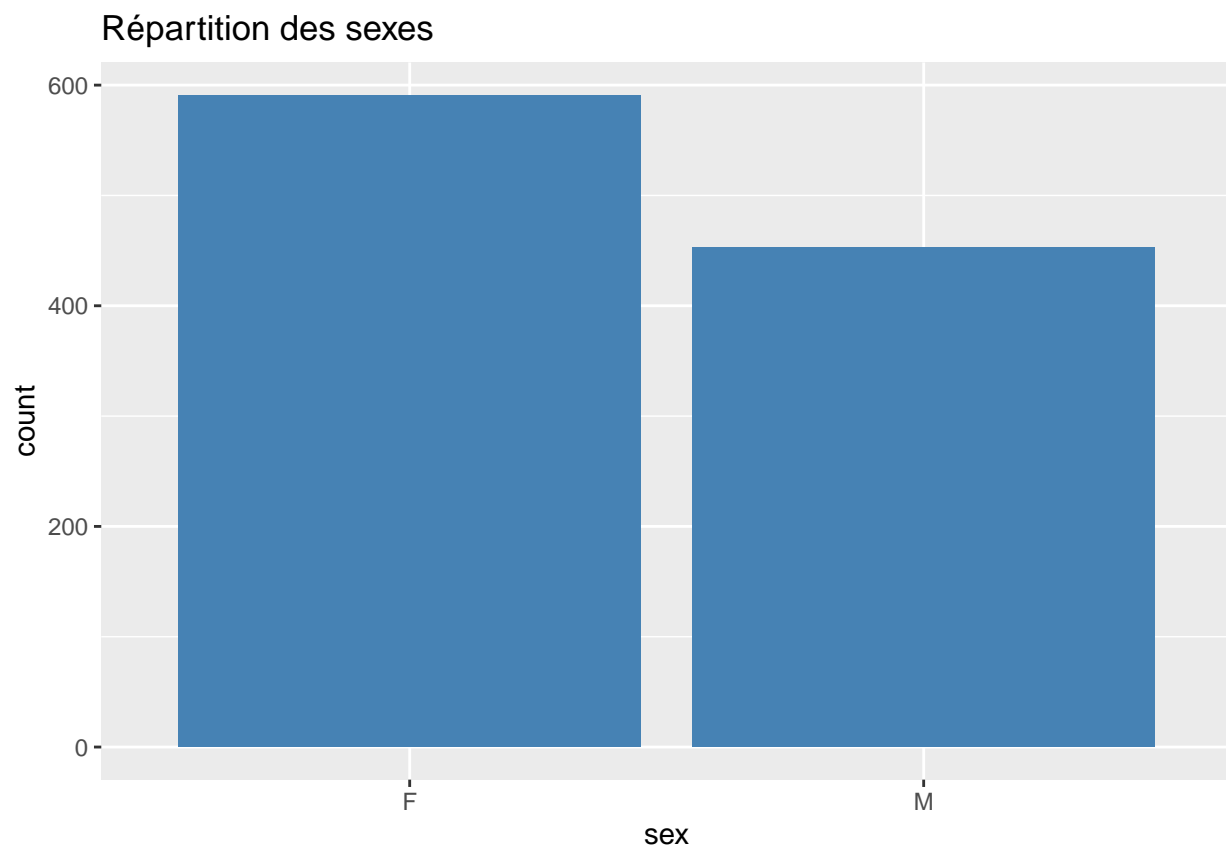
#### Les variables qualitatives

##### Le sexe des étudiants

D'après le diagramme, le dataset est plutôt équilibré en terme d'hommes et de femmes, il y a même plus de femmes que d'hommes dans ce lycée. On étudie ensuite le lien entre le sexe et les notes en effectuant une ANOVA1. D'après le test de Fisher,  $p\text{-value} > 5\%$  donc il n'y a pas d'effet du sexe sur les notes. D'après le test d'indépendance de Chi2 avec l'admission, le sexe des élèves n'a pas de lien avec leur réussite scolaire.

```
# Distribution
library(ggplot2)

##
## Attachement du package : 'ggplot2'
## L'objet suivant est masqué depuis 'package:randomForest':
##
##      margin
ggplot(df, aes(x = sex)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Répartition des sexes")
```



```

# Lien avec la moyenne
summary(lm(Moy ~ sex,data=df))

##
## Call:
## lm(formula = Moy ~ sex, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0152  -2.0152  -0.0152   2.1722   8.1722
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.3486     0.1324  85.706  <2e-16 ***
## sexM         -0.1874     0.2010  -0.932   0.351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.219 on 1042 degrees of freedom
## Multiple R-squared:  0.0008335, Adjusted R-squared:  -0.0001254
## F-statistic: 0.8693 on 1 and 1042 DF,  p-value: 0.3514

# Lien avec la réussite
chisq.test(df$sex,df$RS)

```

```

##
## Pearson's Chi-squared test
##
## data:  df$sex and df$RS
## X-squared = 1.1035, df = 2, p-value = 0.5759

```

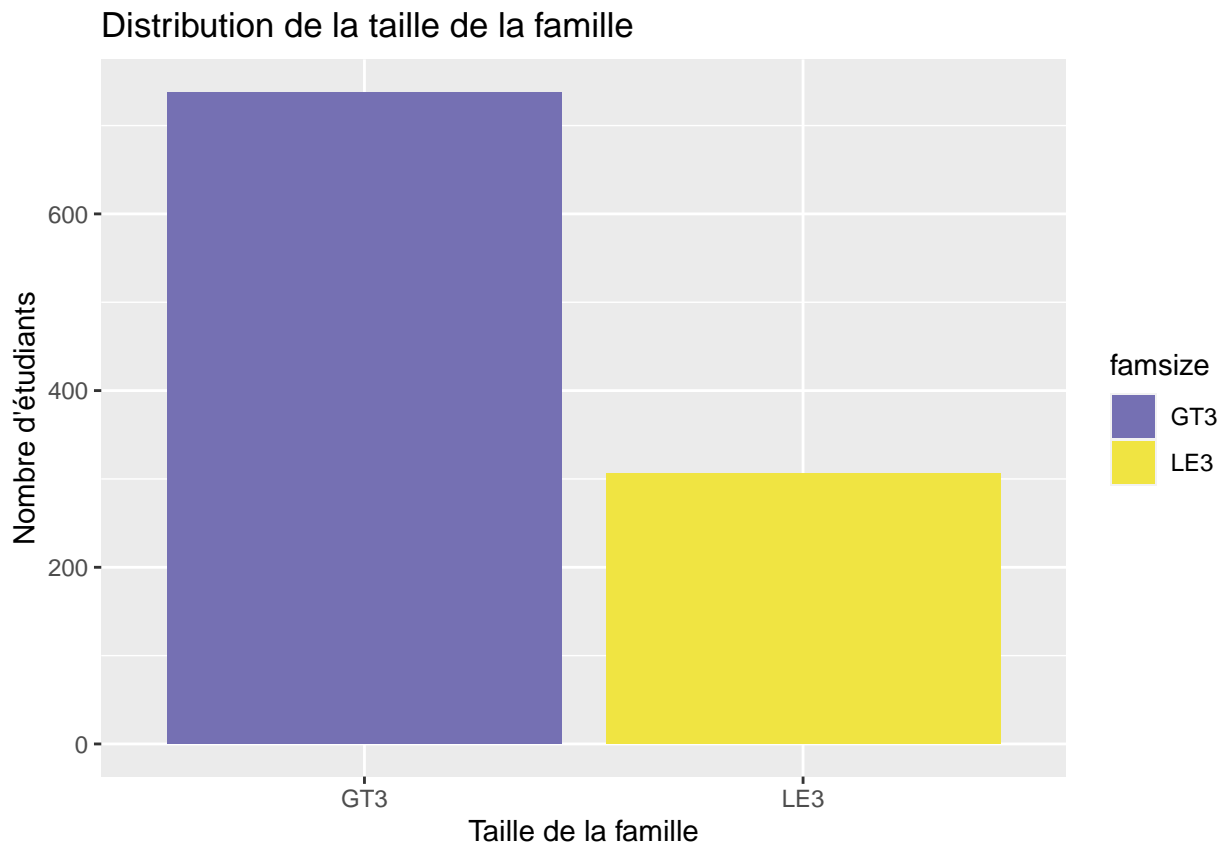
## La taille de la famille

On a deux fois plus de grandes familles que de petites familles. D'après le test de Fisher, il y a bien un impact de la taille de la famille sur les notes. Le test d'indépendance avec la réussite indique cependant que la taille de la famille n'est pas liée à la réussite scolaire.

```

# Distribution
ggplot(data = df, aes(x = famsize, fill = famsize)) +
  geom_bar() +
  labs(title = "Distribution de la taille de la famille",
       x = "Taille de la famille", y = "Nombre d'étudiants") +
  scale_fill_manual(values = c("#7570b3", "#F0E442"))

```



```
# Lien avec les notes
summary(lm(Moy ~ famsize,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ famsize, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.9096 -1.9096 -0.1391  2.1942  8.1942
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.1391     0.1183   94.15  <2e-16 ***
## famsizeLE3    0.4371     0.2185    2.00  0.0457 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.214 on 1042 degrees of freedom
## Multiple R-squared:  0.003825,    Adjusted R-squared:  0.002869
## F-statistic: 4.001 on 1 and 1042 DF,  p-value: 0.04573
```

```
# Lien avec la réussite
chisq.test(df$famsize,df$RS)
```

```
##
## Pearson's Chi-squared test
##
```

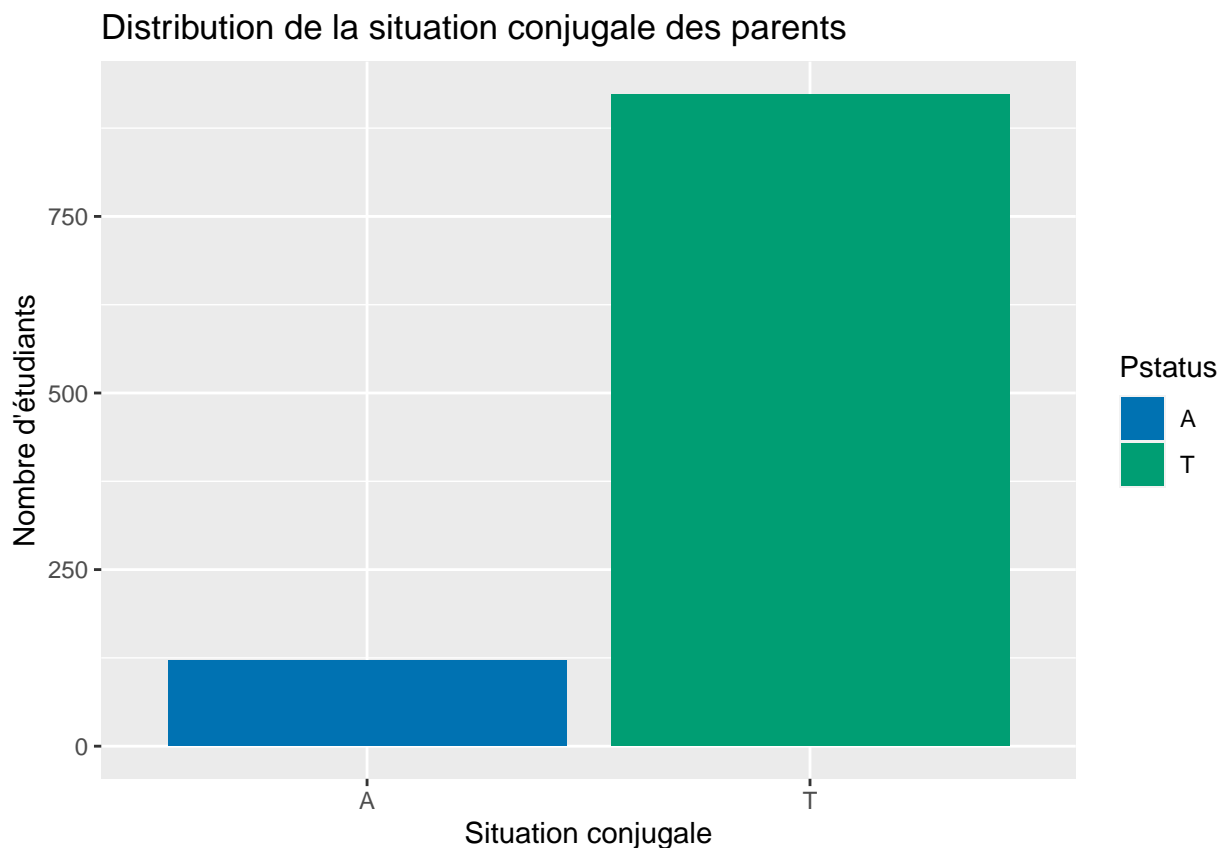
```
## data: df$famsize and df$RS
## X-squared = 4.5986, df = 2, p-value = 0.1003
```

### Situation familiale : séparation des parents

Le jeu est très déséquilibré au sujet de la situation famille : il y a 4 fois plus d'étudiants qui ont leurs parents qui vivent ensemble. De plus, le test de Fisher indique que la situation familiale n'a pas d'impact sur les notes. Le test de Chi2 soutient que le status des parents et la réussite scolaire sont indépendants.

#### # Distribution

```
ggplot(data = df, aes(x = Pstatus, fill = Pstatus)) +
  geom_bar() +
  scale_fill_manual(values = c("#0072B2", "#009E73")) +
  labs(title = "Distribution de la situation conjugale des parents",
       x = "Situation conjugale", y = "Nombre d'étudiants")
```



#### # Lien avec les notes

```
summary(lm(Moy ~ Pstatus,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ Pstatus, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0744  -1.9155   0.0845   2.0845   8.0845
##
## Coefficients:
```



```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.4077     0.2927   38.97  <2e-16 ***
## PstatusT     -0.1589     0.3113   -0.51    0.61
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.22 on 1042 degrees of freedom
## Multiple R-squared:  0.0002499, Adjusted R-squared:  -0.0007095
## F-statistic: 0.2605 on 1 and 1042 DF, p-value: 0.6099
# Lien avec la réussite
chisq.test(df$Pstatus,df$RS)

##
## Pearson's Chi-squared test
##
## data:  df$Pstatus and df$RS
## X-squared = 0.9565, df = 2, p-value = 0.6199
```

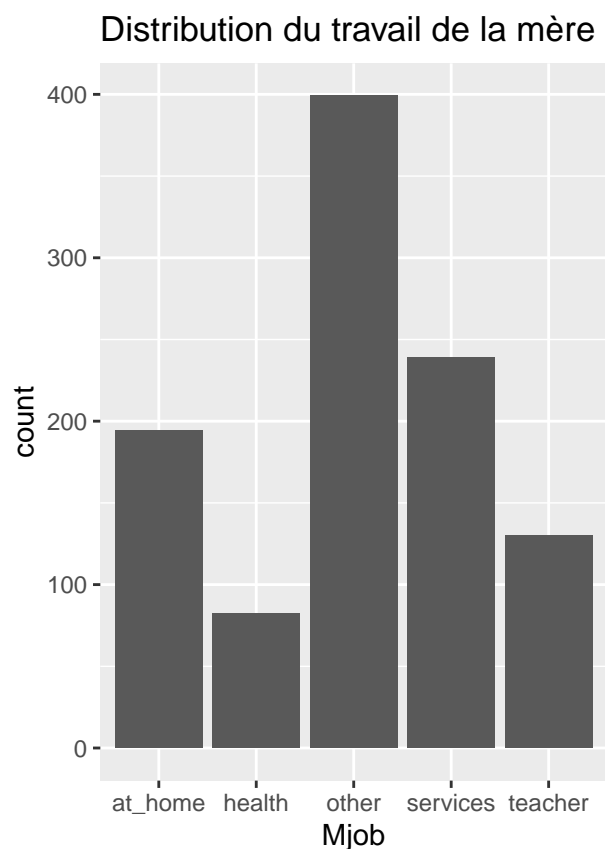
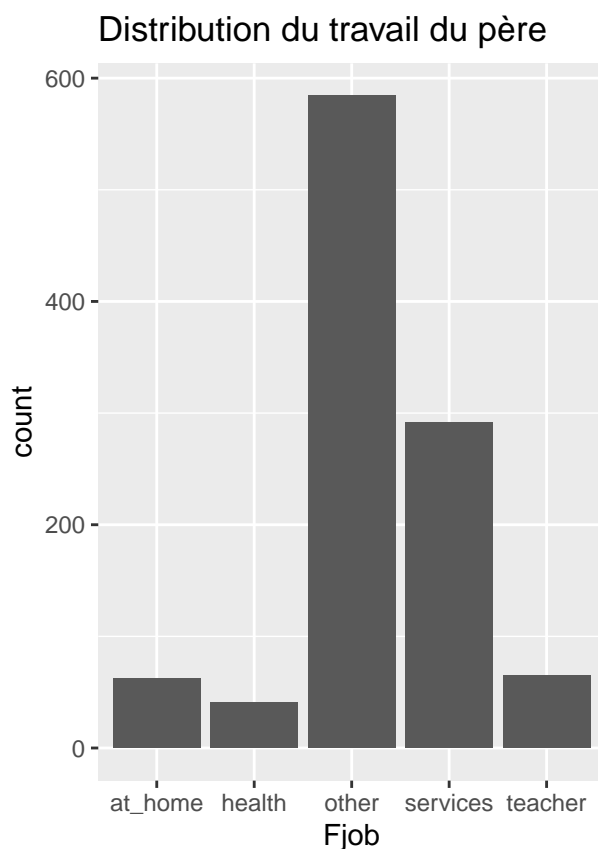
## Travail des parents

Dans les deux cas, others et services sont les catégories qui dominent. Une différence notable est la que la proportion de femme au-foyer est bien plus élevée que celle des hommes. D'après le test de Fisher, le travail de la mère a un impact sur les notes, contrairement à celui du père. Les résultats des test de Chis2 suivent les résultats des test de Fisher : le travail de la mère et la réussite scolaire sont bien corrélés mais celui du père n'a pas d'impact.

```
#Distributions
g2=ggplot(data = df, aes(x = Mjob)) +
  geom_bar() +
  labs(title = "Distribution du travail de la mère") +
  scale_fill_manual(values = c("#7570b3", "#0072B2", "#E69F00", "#009E73", "#F0E442"))

g1=ggplot(data = df, aes(x = Fjob)) +
  geom_bar() +
  labs(title="Distribution du travail du père") +
  scale_fill_manual(values = c("#7570b3", "#0072B2", "#E69F00", "#009E73", "#F0E442"))

grid.arrange(g1, g2, ncol = 2)
```



*# Lien avec les notes*

```
summary(lm(Moy ~ Medu+Fedu,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ Medu + Fedu, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.265  -1.732   0.068   2.126   7.852
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.4233     0.2595  36.316 < 2e-16 ***
## Medu           0.5214     0.1125   4.635 4.02e-06 ***
## Fedu           0.2037     0.1150   1.771  0.0769 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.133 on 1041 degrees of freedom
## Multiple R-squared:  0.05434,    Adjusted R-squared:  0.05252
## F-statistic: 29.91 on 2 and 1041 DF,  p-value: 2.344e-13
```

*# Lien avec la réussite*

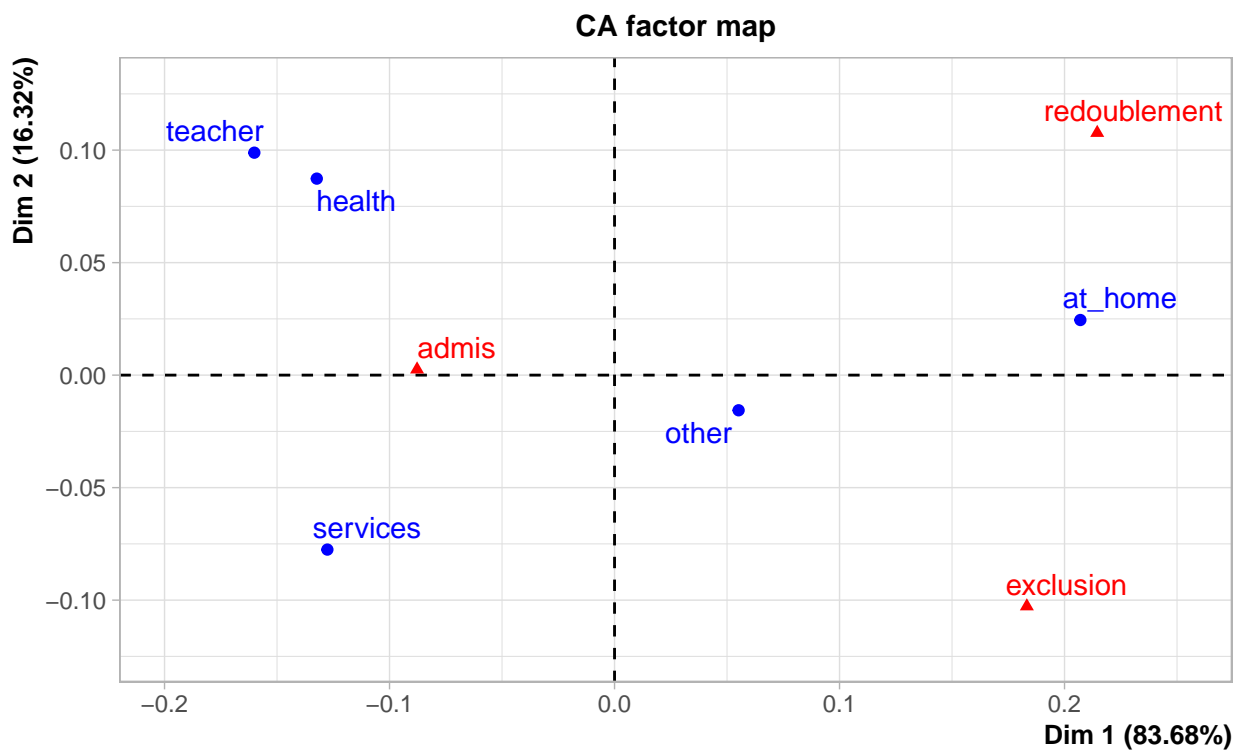
```
chisq.test(df$Mjob,df$RS)
```

```
##
## Pearson's Chi-squared test
```

```
##
## data: df$Mjob and df$RS
## X-squared = 21.736, df = 8, p-value = 0.005429
chisq.test(df$Fjob,df$RS)
```

```
##
## Pearson's Chi-squared test
##
## data: df$Fjob and df$RS
## X-squared = 7.4964, df = 8, p-value = 0.4841
```

```
# AFC sur le travail de la mère
df.Mjob = data.frame(df$Mjob,df$RS)
table.Mjob = table(df.Mjob)
res = CA(table.Mjob)
```

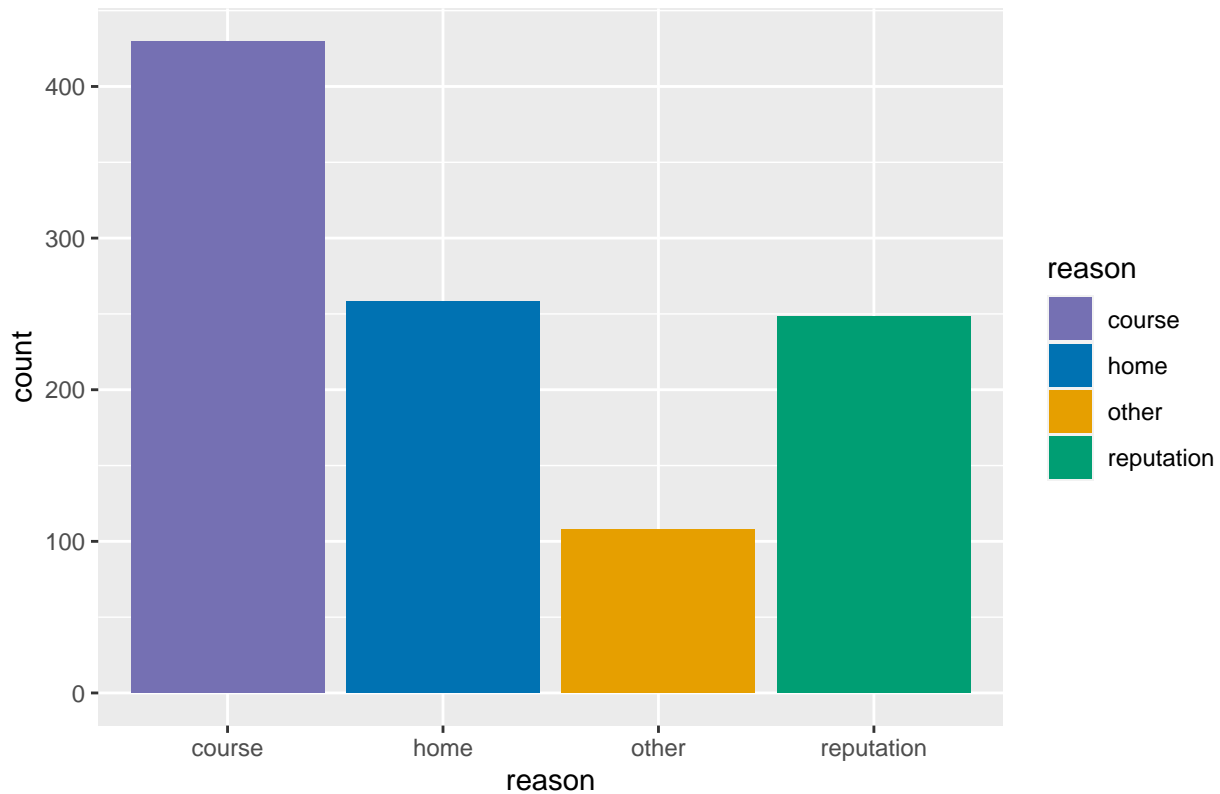


### Les raisons du choix d'école

D'après le digramme circulaire, seule "other" possède un petit effectif alors que "course" domine. Ainsi, les élèves vont majoritairement en cours car ils les apprécient. D'après l'ANOVA1, il est clair que la raison d'aller en cours impacte les notes des étudiants ( $p\text{-value} < 5\%$ ). Cela paraît cohérent étant donné que cela détermine leur motivation à avoir de bonnes notes. De la même manière, la raison est bien corrélée avec la réussite scolaire, ce qui paraît bien cohérent.

```
# Distribution
ggplot(data = df, aes(x = reason, fill = reason)) +
  geom_bar() +
  labs(title="Distribution du travail du père") +
  scale_fill_manual(values = c("#7570b3", "#0072B2", "#E69F00", "#009E73", "#F0E442"))
```

## Distribution du travail du père



```
# Lien avec les notes
summary(lm(Moy~ reason,data=df))
```

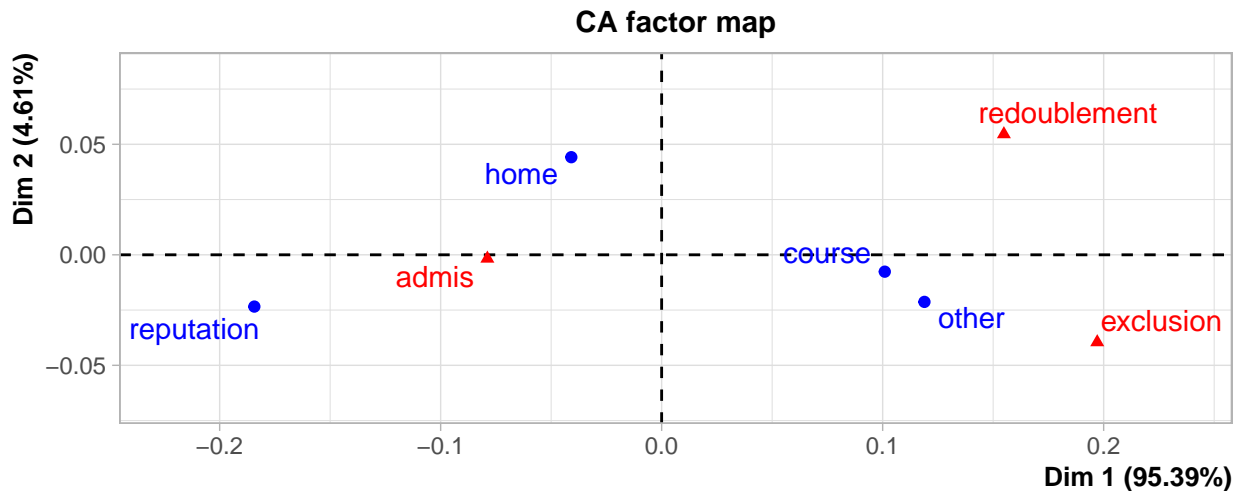
```
##
## Call:
## lm(formula = Moy ~ reason, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3858  -1.8791  -0.0052   2.1209   7.7876
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.87907    0.15372  70.771 < 2e-16 ***
## reasonhome     0.45943    0.25103   1.830  0.0675 .
## reasonother    -0.03956    0.34309  -0.115  0.9082
## reasonreputation 1.17335    0.25417   4.616 4.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.188 on 1040 degrees of freedom
## Multiple R-squared:  0.02209,    Adjusted R-squared:  0.01927
## F-statistic: 7.832 on 3 and 1040 DF,  p-value: 3.587e-05
```

```
# Lien avec la réussite
chisq.test(df$reason,df$RS)
```

```
##
```

```
## Pearson's Chi-squared test
##
## data: df$reason and df$RS
## X-squared = 15.479, df = 6, p-value = 0.01684

# AFC sur le travail de la mère
df.reason = data.frame(df$reason,df$RS)
table.reason = table(df.reason)
res = CA(table.reason)
```



## Les relations

Il y a environ deux fois plus de jeunes célibataires que de jeunes en couple. On peut penser qu'être en couple réduit le temps passé à étudier et rajoute des distractions, donc il devrait avoir un impact négatif sur les notes. D'après le test de Fisher, la p-value est fortement inférieure à 5%, donc on rejette  $H_0$ : il y a bien un lien entre situation romantique et notes, ce qui rejoint bien l'idée de départ. Il serait donc intéressant d'étudier la distribution des notes selon la situation romantique. D'après les boxplots, les différences sont assez minimes, même si on peut apercevoir que les notes des célibataires sont légèrement meilleures. Cependant, la présence de relation amoureuse n'a pas d'impact sur la réussite scolaire. Ainsi, être en couple fait baisser la moyenne mais n'est pas un facteur d'échec.

```
# Distribution
gr1=ggplot(df, aes(x = romantic)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution des personnes en couple",
        x = "Couple", y = "Nombre d'étudiants")

# Lien avec les notes
summary(lm(Moy~ romantic,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ romantic, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1486  -1.9455   0.1222   2.1847   7.8514
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

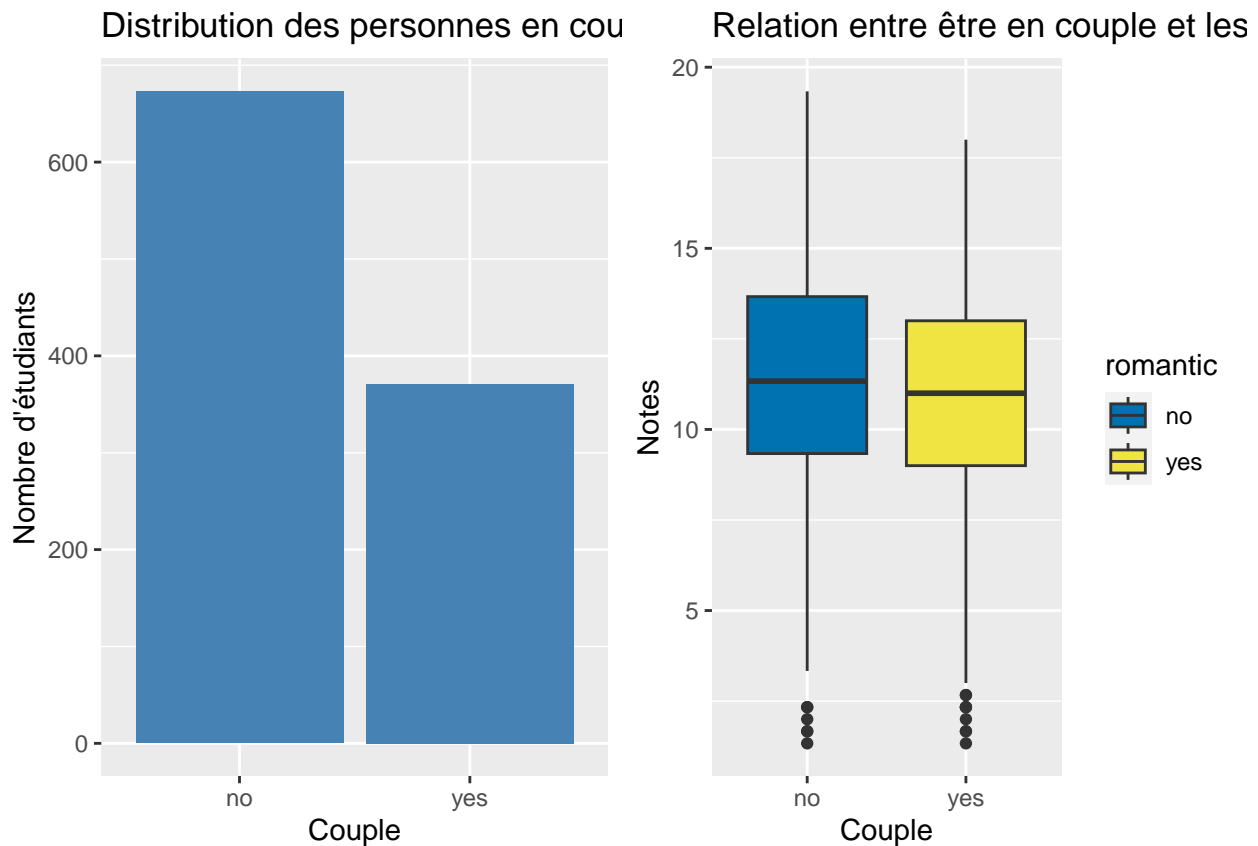
```
## (Intercept) 11.4819      0.1236  92.871 < 2e-16 ***
## romanticyes -0.6041      0.2074  -2.913  0.00366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.207 on 1042 degrees of freedom
## Multiple R-squared:  0.008077,    Adjusted R-squared:  0.007125
## F-statistic: 8.485 on 1 and 1042 DF,  p-value: 0.003658
```

```
yes = df$Moy[df$romantic=='yes']
no = df$Moy[df$romantic=='no']
```

```
# Boxplot des notes
```

```
gr2=ggplot(data = df, aes(x = romantic, y = Moy, fill = romantic)) +
  geom_boxplot() +
  scale_fill_manual(values = c("#0072B2", "#F0E442")) +
  labs(title = "Relation entre être en couple et les notes",
       x = "Couple", y = "Notes")
```

```
grid.arrange(gr1, gr2, ncol = 2)
```



```
# Lien avec la réussite
chisq.test(df$romantic,df$RS)
```

```
##
## Pearson's Chi-squared test
##
## data: df$romantic and df$RS
```

```
## X-squared = 5.5477, df = 2, p-value = 0.06242
```

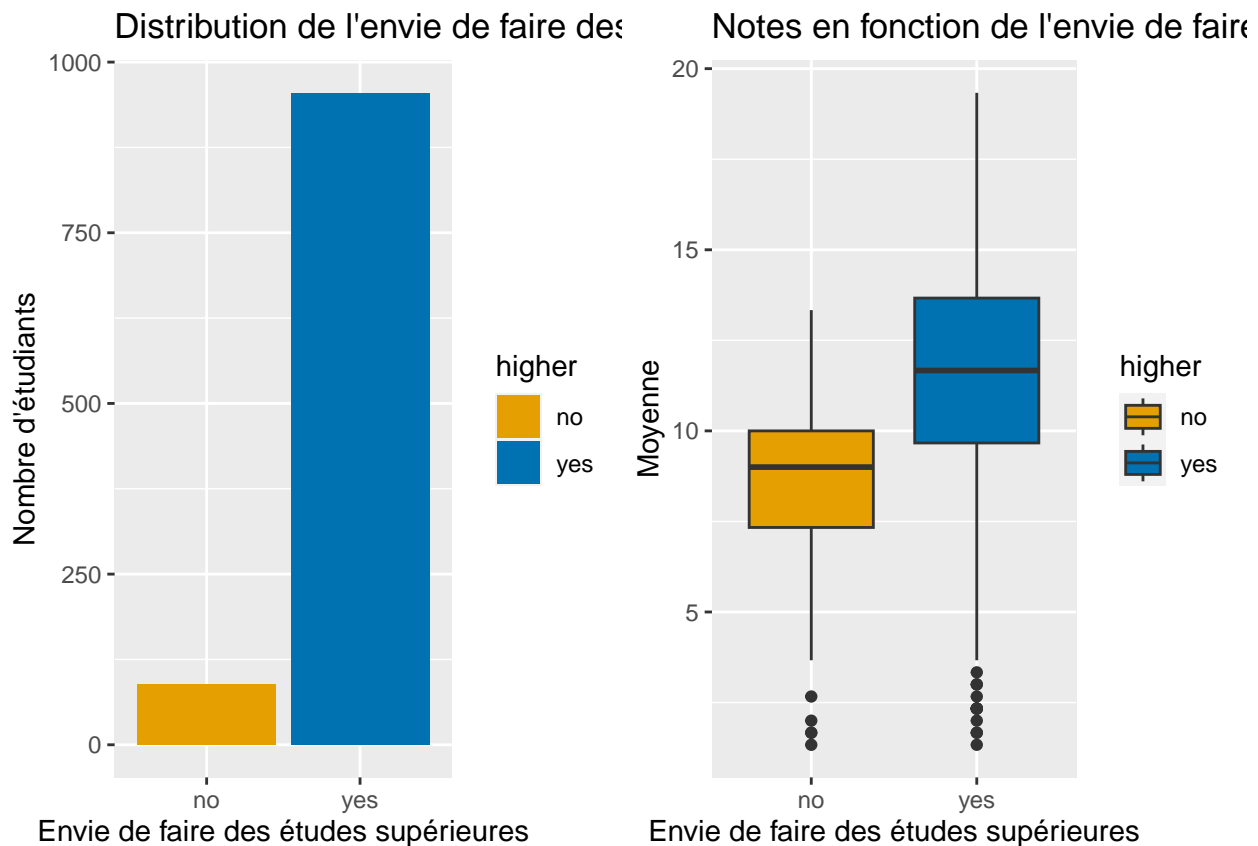
### Volonté de faire des études supérieures

On observe qu'au moins 80% des élèves veulent continuer leur études après le lycée, ce qui est plutôt rassurant. De plus, d'après le test de Fisher, les deux variables sont corrélées. On peut également annoncer que ceux qui veulent faire des études supérieures tendent à avoir de meilleures notes grâce au test unilatéral. A priori, la volonté de faire des études supérieures est corrélée à la réussite scolaire. Donc, ceux qui veulent poursuivre leurs études auront de meilleures notes et tendance à ne pas être en échec.

```
# distribution
g1=ggplot(df, aes(x = higher, fill = higher)) +
  geom_bar() +
  labs(title = "Distribution de l'envie de faire des études supérieures",
       x = "Envie de faire des études supérieures", y = "Nombre d'étudiants") +
  scale_fill_manual(values = c("#E69F00", "#0072B2"))

# Boxplot des notes en fonction de l'envie de faire des études supérieures
g2=ggplot(df, aes(x = higher, y = Moy, fill = higher)) +
  geom_boxplot() +
  labs(title = "Notes en fonction de l'envie de faire des études supérieures",
       x = "Envie de faire des études supérieures", y = "Moyenne") +
  scale_fill_manual(values = c("#E69F00", "#0072B2"))

grid.arrange(g1, g2, ncol = 2)
```



```
# Lien avec les notes
summary(lm(Moy ~ higher, data=df))
```

```
##
## Call:
## lm(formula = Moy ~ higher, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1930  -1.8597   0.1403   2.1403   7.8070
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.4869     0.3293  25.775  <2e-16 ***
## higheryes     3.0395     0.3443   8.829  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.106 on 1042 degrees of freedom
## Multiple R-squared:  0.0696, Adjusted R-squared:  0.06871
## F-statistic: 77.95 on 1 and 1042 DF,  p-value: < 2.2e-16

yes = df$Moy[df$higher=='yes']
no = df$Moy[df$higher=='no']

# Lien avec la réussite
chisq.test(df$higher,df$RS)

##
## Pearson's Chi-squared test
##
## data:  df$higher and df$RS
## X-squared = 66.594, df = 2, p-value = 3.461e-15

# AFC sur la volonté de faire des études sup
df.higher = data.frame(df$higher,df$RS)
table.higher = table(df.higher)
res = CA(table.higher)
```

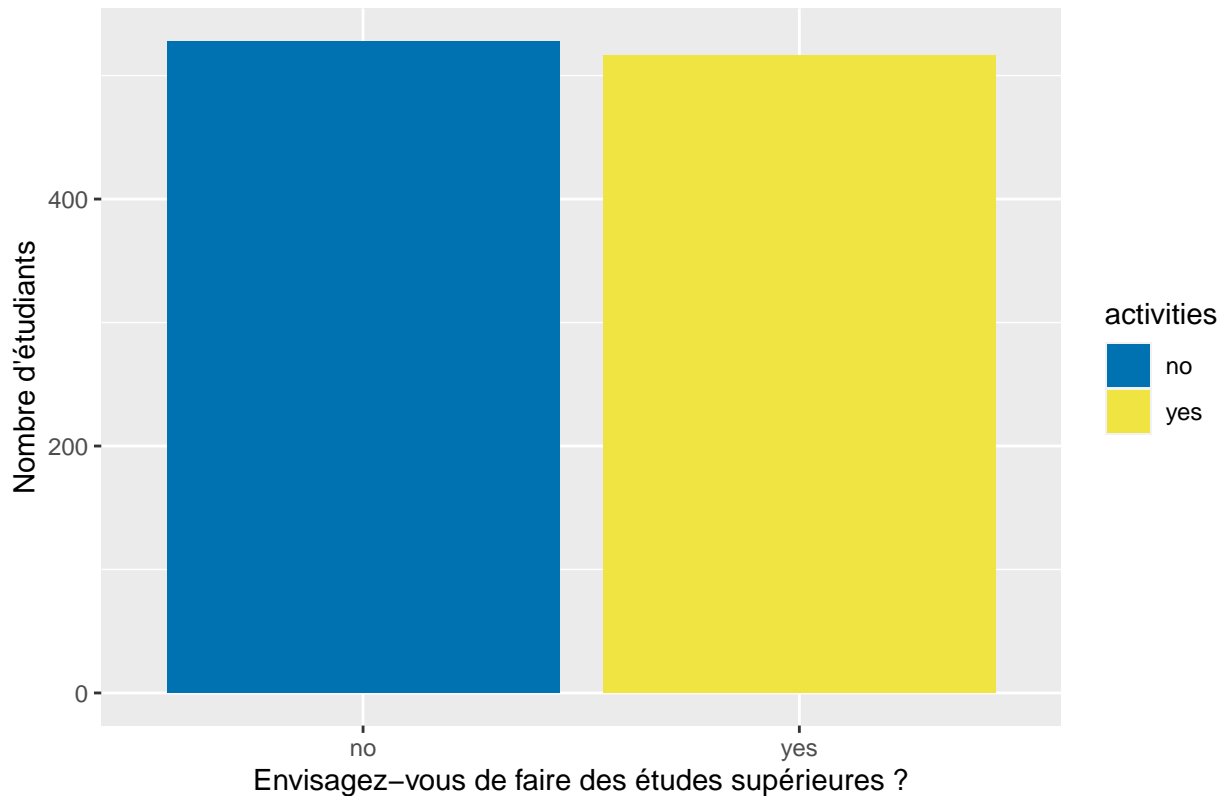
## Activités extrascolaires

On a autant d'élèves qui pratiquent des activités extrascolaires que d'élèves qui n'en pratiquent pas, ce qui est plutôt intéressant. De plus, le test de Fisher indique plutôt qu'il n'y a pas de liens entre les activités extrascolaires et les notes, ce qui est plutôt surprenant étant donné que l'on aurait tendance à penser que les étudiants ayant des activités, ont moins de temps pour étudier. Dans la même lignée, les activités sont plutôt indépendantes de la réussite d'après le test de Chi2.

```
# Distribution
ggplot(df, aes(x = activities, fill = activities)) +
  geom_bar() +
  labs(title = "Distribution de de la pratique d'activités extrascolaires",
       x = "Envisagez-vous de faire des études supérieures ?",
       y = "Nombre d'étudiants") +
  scale_fill_manual(values = c("#0072B2", "#F0E442"))
```



## Distribution de de la pratique d'activités extrascolaires



```
# Lien avec les notes
summary(lm(Moy ~ activities,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ activities, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1085  -2.0966  -0.0966   2.2248   7.8915
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    11.0966     0.1399  79.292  <2e-16 ***
## activitiesyes     0.3453     0.1991   1.734   0.0831 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.216 on 1042 degrees of freedom
## Multiple R-squared:  0.002879, Adjusted R-squared:  0.001922
## F-statistic: 3.008 on 1 and 1042 DF, p-value: 0.08313
```

```
# Lien avec la réussite
chisq.test(df$activities,df$RS)
```

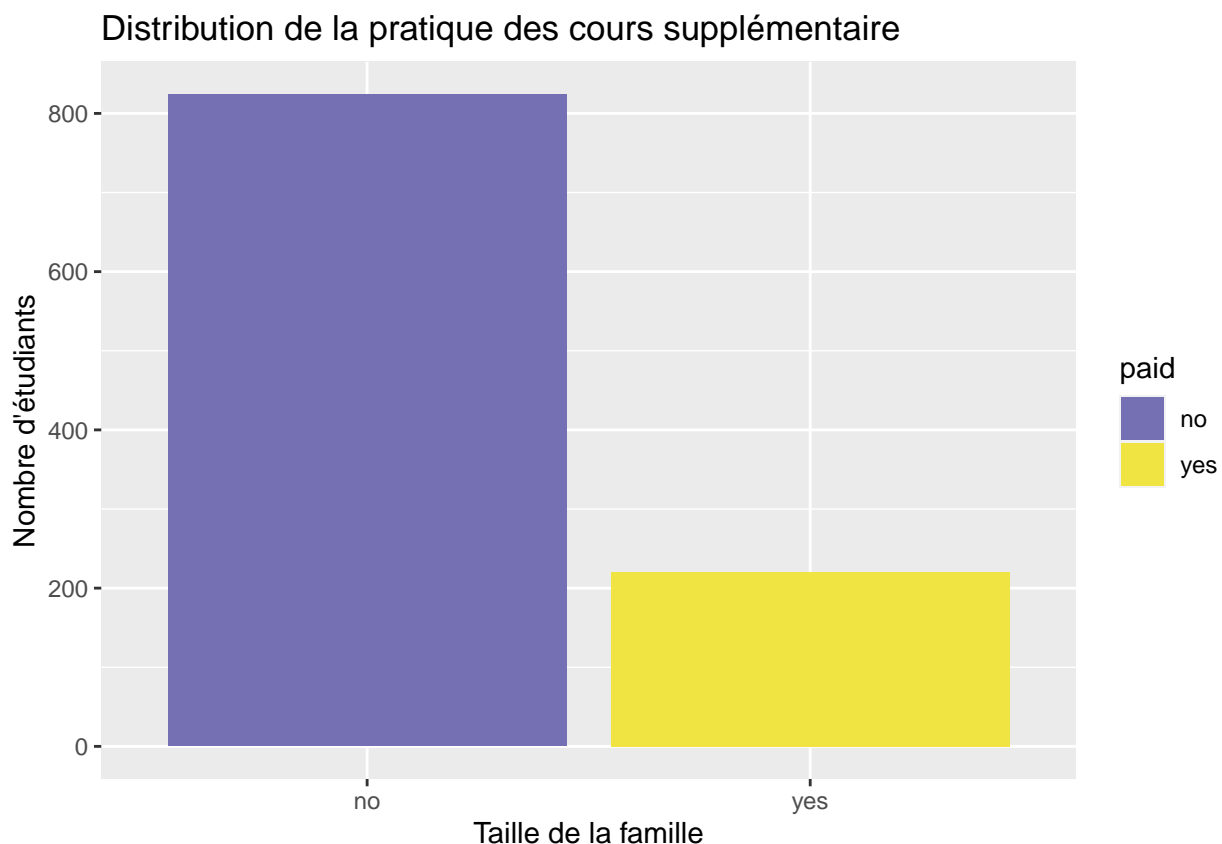
```
##
## Pearson's Chi-squared test
##
```

```
## data: df$activities and df$RS
## X-squared = 2.5236, df = 2, p-value = 0.2831
```

### Cours supplémentaires

Il y a bien plus d'élèves qui ne suivent pas de cours supplémentaires que d'élèves qui en suivent. Cette distribution est cohérente avec l'idée qu'on peut se faire. Le test de Fisher indique plutôt que les suivis de cours supplémentaires n'a pas d'impact sur la moyenne. De même, le suivi de cours supplémentaire n'est pas lié à la réussite.

```
# Distribution
ggplot(data = df, aes(x = paid, fill = paid)) +
  geom_bar() +
  labs(title = "Distribution de la pratique des cours supplémentaire",
       x = "Taille de la famille", y = "Nombre d'étudiants") +
  scale_fill_manual(values = c("#7570b3", "#F0E442"))
```



```
# Lien avec la moyenne
summary(lm(Moy ~ paid, data=df))
```

```
##
## Call:
## lm(formula = Moy ~ paid, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.9951 -1.9951  0.0049  2.0049  8.0049
##
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.3285      0.1121  101.05  <2e-16 ***
## paidyes      -0.2906      0.2442   -1.19    0.234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.218 on 1042 degrees of freedom
## Multiple R-squared:  0.001357,    Adjusted R-squared:  0.0003986
## F-statistic: 1.416 on 1 and 1042 DF,  p-value: 0.2344
# Lien avec la réussite
chisq.test(df$paid,df$RS)

##
## Pearson's Chi-squared test
##
## data:  df$paid and df$RS
## X-squared = 4.8571, df = 2, p-value = 0.08816
```

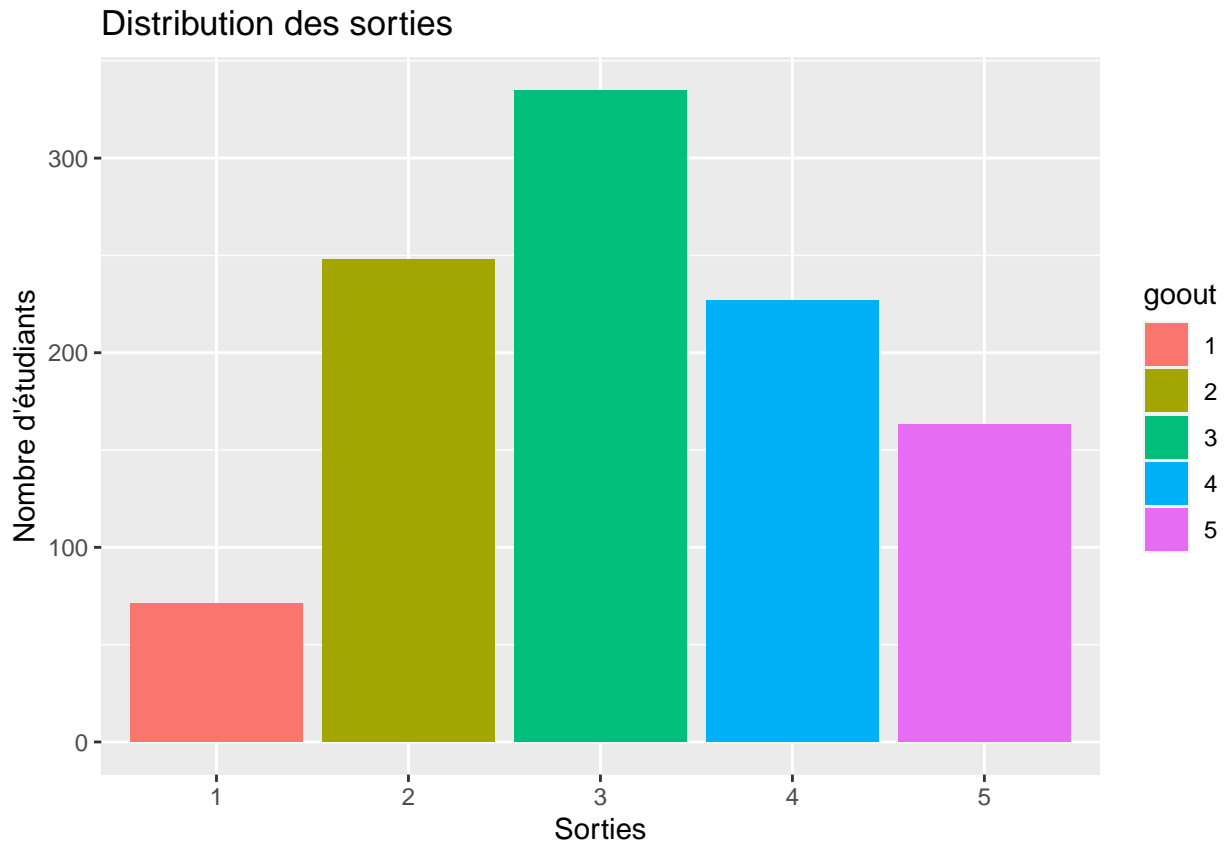
## Les variables qualitatives à modalités numériques

### Les sorties

On remarque que les élèves maintiennent leur vie sociale. La grosse majorité sont intermédiaires en termes de sorties ce qui est quand même rassurant. Il y a quand même plus de personnes qui sortent vraiment beaucoup que de personnes qui ne sortent pas. Le test de Fisher indique les sorties sont très corrélées au notes et le test de Chi2 montre que la réussite scolaire est aussi corrélée aux sorties. Ainsi, on retrouve des résultats qui semblent cohérents et représentatifs de la vie étudiante.

Etant donné, la corrélation entre RS et goout, on peut effectuer une AFC pour préciser. On peut remarquer que ceux qui sortent peu-moyennement auront tendance à être admis alors que ce qui ne sortent pas (retrait/exclusion social) vont plutôt redoubler et les autres vont avoir tendances à se faire exclure. On obtient donc des résultats qui semblent plutôt pertinents.

```
# Distribution
ggplot(data = df, aes(x =goout, fill = goout)) +
  geom_bar() +
  labs(title = "Distribution des sorties",
       x = "Sorties", y = "Nombre d'étudiants")
```



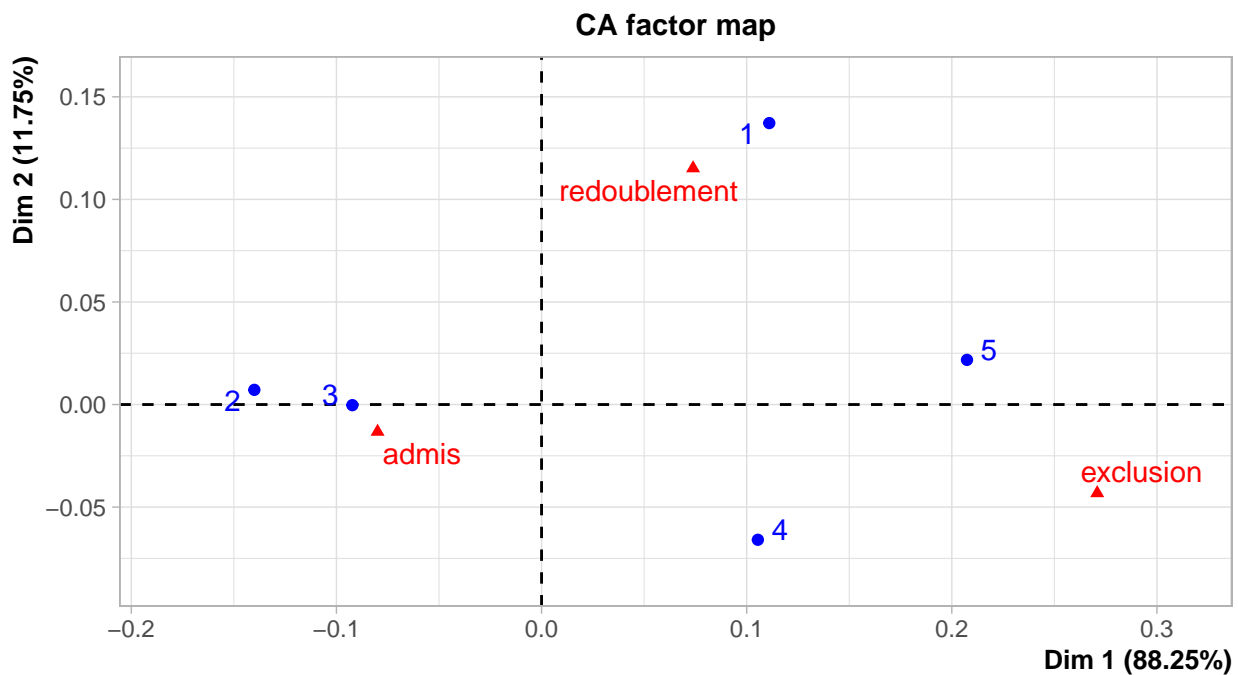
```
# Lien avec la moyenne
summary(lm(Moy ~ goout,data=df))
```

```
##
## Call:
## lm(formula = Moy ~ goout, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5887  -1.8876  -0.0015   2.1124   7.6652
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.5493     0.3770  27.980 < 2e-16 ***
## goout2        1.3727     0.4276   3.210  0.00137 **
## goout3        1.0049     0.4151   2.421  0.01564 *
## goout4        0.4522     0.4320   1.047  0.29548
## goout5       -0.1853     0.4517  -0.410  0.68178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.177 on 1039 degrees of freedom
## Multiple R-squared:  0.02957,    Adjusted R-squared:  0.02583
## F-statistic: 7.915 on 4 and 1039 DF,  p-value: 2.766e-06
```

```
# Lien avec la réussite
chisq.test(df$goout,df$RS)
```

```
##
## Pearson's Chi-squared test
##
## data: df$goout and df$RS
## X-squared = 20.537, df = 8, p-value = 0.008485
```

```
# AFC sur les sorties
df.goout = data.frame(df$goout,df$RS)
table.goout = table(df.goout)
res = CA(table.goout)
```

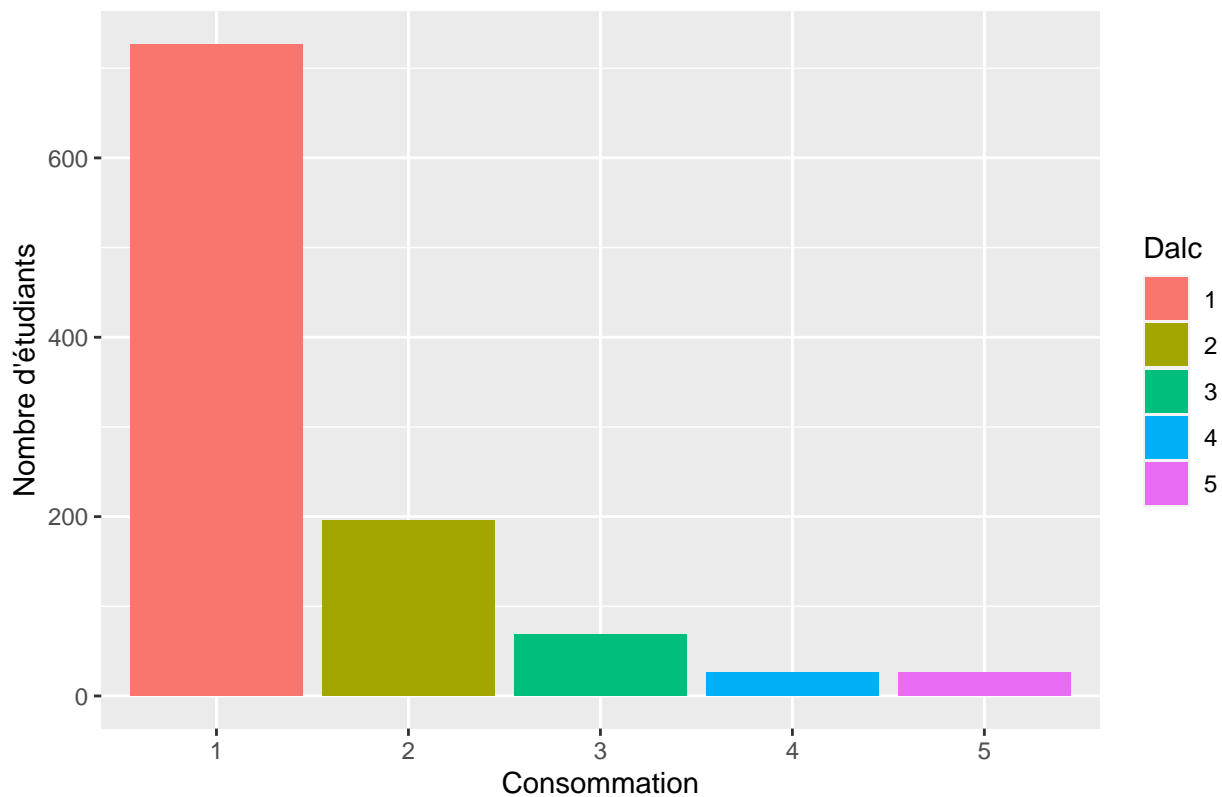


### La consommation d'alcool

On s'intéresse enfin à la feature "principale" de ce jeu de données, la consommation d'alcool des étudiants.

```
# Distribution
ggplot(data = df, aes(x =Dalc, fill = Dalc)) +
  geom_bar() +
  labs(title = "Distribution de la consommation d'alcool en semaine",
       x = "Consommation", y = "Nombre d'étudiants")
```

## Distribution de la consommation d'alcool en semaine



```
# Lien avec la moyenne
summary(lm(Moy ~ Dalc,data=df))
```

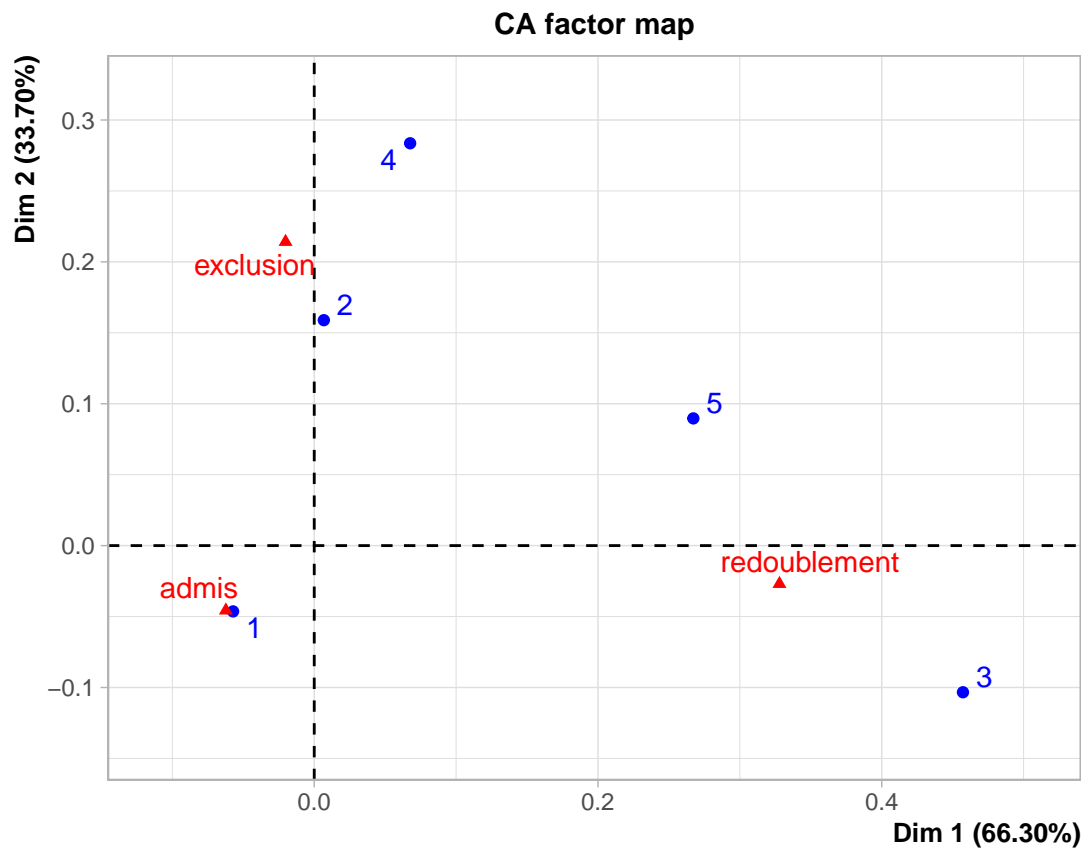
```
##
## Call:
## lm(formula = Moy ~ Dalc, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.244  -1.911   0.089   2.089   7.756
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5777    0.1181  98.001  < 2e-16 ***
## Dalc2        -0.8889    0.2564  -3.467  0.000547 ***
## Dalc3        -0.8917    0.4013  -2.222  0.026475 *
## Dalc4        -2.0008    0.6358  -3.147  0.001696 **
## Dalc5        -1.3982    0.6358  -2.199  0.028079 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.185 on 1039 degrees of freedom
## Multiple R-squared:  0.02443,    Adjusted R-squared:  0.02068
## F-statistic: 6.505 on 4 and 1039 DF,  p-value: 3.594e-05
```

```
# Lien avec la réussite
chisq.test(df$Dalc,df$RS)
```

```
## Warning in chisq.test(df$Dalc, df$RS): Chi-squared approximation may be
## incorrect
```

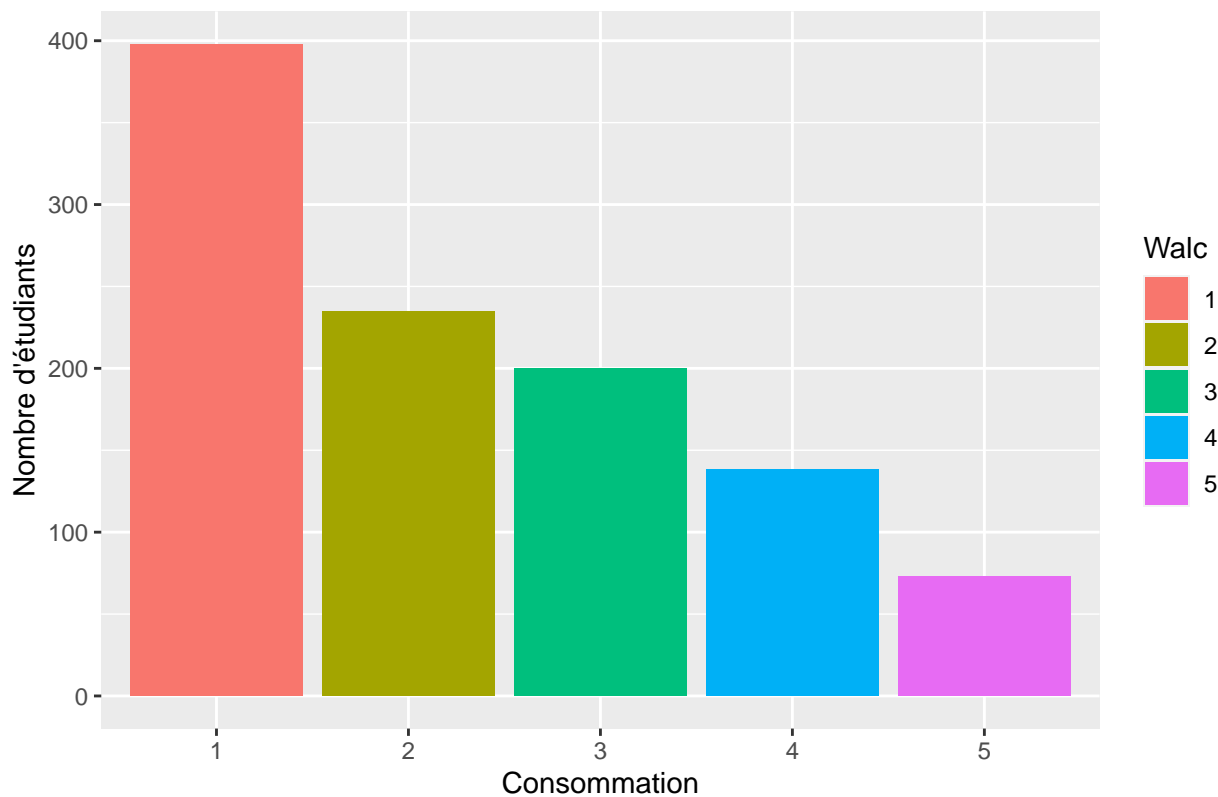
```
##
## Pearson's Chi-squared test
##
## data: df$Dalc and df$RS
## X-squared = 28.342, df = 8, p-value = 0.0004134
```

```
# AFC sur les sorties
df.Dalc = data.frame(df$Dalc, df$RS)
table.Dalc = table(df.Dalc)
res = CA(table.Dalc)
```



```
# Distribution
ggplot(data = df, aes(x = Walc, fill = Walc)) +
  geom_bar() +
  labs(title = "Distribution de la consommation d'alcool le week-end",
        x = "Consommation", y = "Nombre d'étudiants")
```

## Distribution de la consommation d'alcool le week-end



```
# Lien avec la moyenne
summary(lm(Moy ~ Walc,data=df))
```

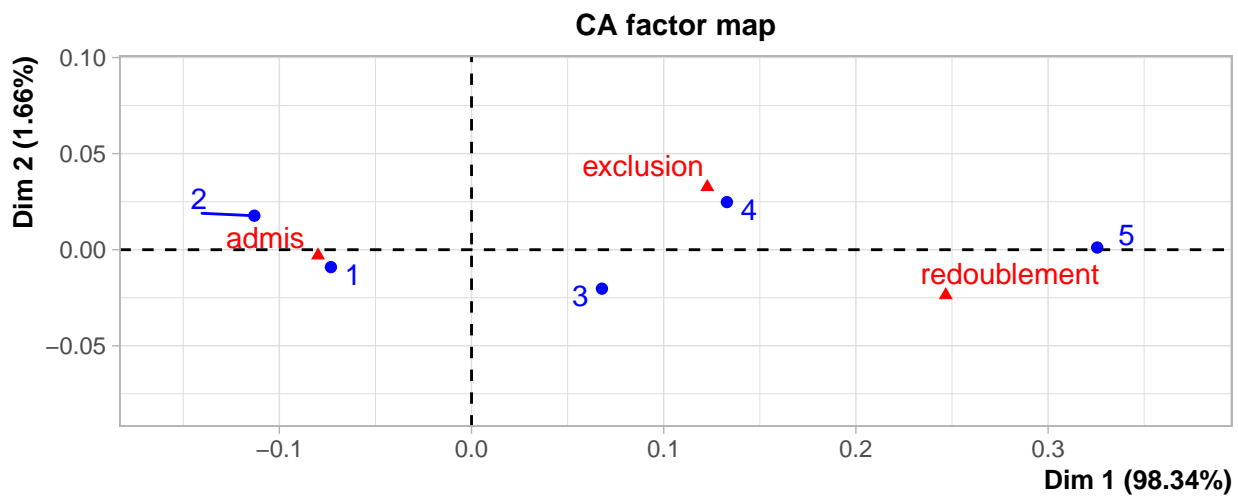
```
##
## Call:
## lm(formula = Moy ~ Walc, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2831  -1.9050   0.0503   2.0614   7.7169
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.6164     0.1600  72.593 < 2e-16 ***
## Walc2        -0.1398     0.2626  -0.532 0.594577
## Walc3        -0.3781     0.2767  -1.366 0.172115
## Walc4        -1.1768     0.3154  -3.731 0.000201 ***
## Walc5        -1.2831     0.4065  -3.157 0.001642 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.192 on 1039 degrees of freedom
## Multiple R-squared:  0.0201, Adjusted R-squared:  0.01633
## F-statistic: 5.328 on 4 and 1039 DF,  p-value: 0.0003004
```

```
# Lien avec la réussite
chisq.test(df$Walc,df$RS)
```



```
##
## Pearson's Chi-squared test
##
## data: df$Walc and df$RS
## X-squared = 16.5, df = 8, p-value = 0.03576
```

```
# AFC sur les sorties
df.Walc = data.frame(df$Walc,df$RS)
table.Walc = table(df.Walc)
res = CA(table.Walc)
```



## Les variables quantitatives

```
data=df
data_quanti=data[c(3,7,8,13,14,15,25,26,27,28,29,30,31,32,33)]
data_quanti_mat=df.mat[c(3,7,8,13,14,15,25,26,27,28,29,30,31,32,33)]
data_quanti_por=df.por[c(3,7,8,13,14,15,25,26,27,28,29,30,31,32,33)]
head(data_quanti)
```

```
##   age Medu Fedu traveltime studytime failures freetime goout Dalc Walc health
## 1  18   4    4          2          2          0          3    4    1    1    3
## 2  17   1    1          1          2          0          3    3    1    1    3
## 3  15   1    1          1          2          3          3    2    2    3    3
## 4  15   4    2          1          3          0          2    2    1    1    5
## 5  16   3    3          1          2          0          3    2    1    2    5
## 6  16   4    3          1          2          0          4    2    1    2    5
##   absences G1 G2 G3
## 1         6 5 6 6
## 2         4 5 5 6
## 3        10 7 8 10
## 4         2 15 14 15
## 5         4 6 10 10
## 6        10 15 15 15
```

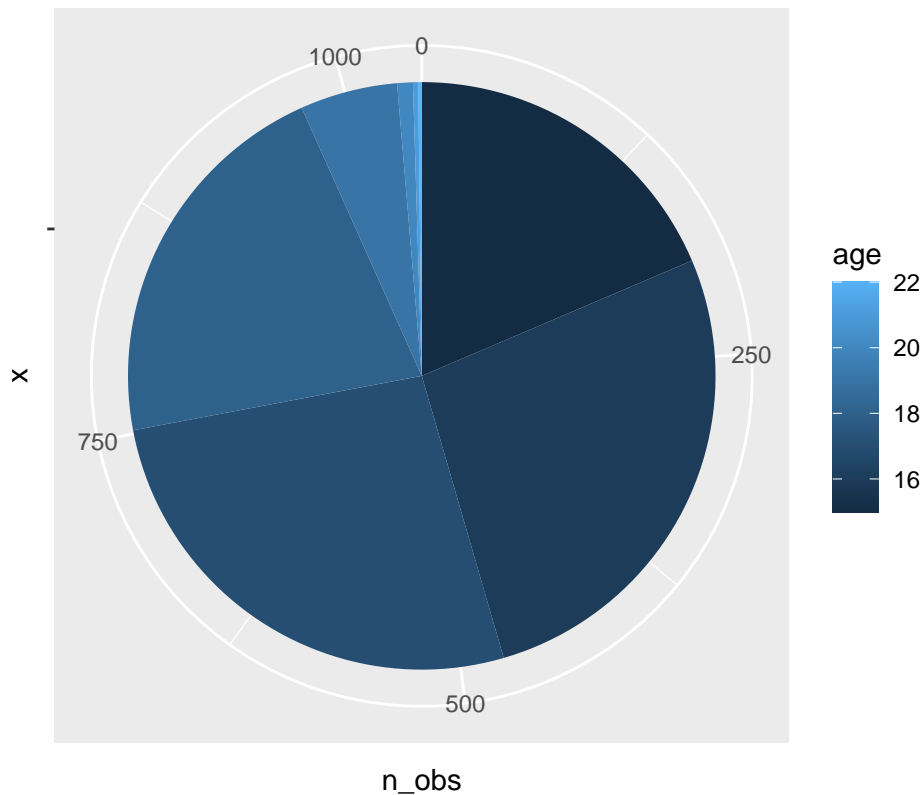
## L'âge des élèves

```
attach(data_quanti)
data_age=data_quanti
```

```
data_age=summarise(group_by(data_age,age),n_obs=n()) #on groupe par âge avec le nombre de personnes dan.

#création du camembert
ggplot(data = data_age, aes(x = "", y = n_obs, fill = age)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Répartition par âge toutes filière confondue")
```

### Répartition par âge toutes filière confondue



La couleur la plus claire correspond à l'âge le plus grand (22 ans), dès que l'on passe à une couleur plus foncée, on diminue l'âge de 1. On voit clairement ici que la majorité des étudiants ont entre 15 et 19 ans.

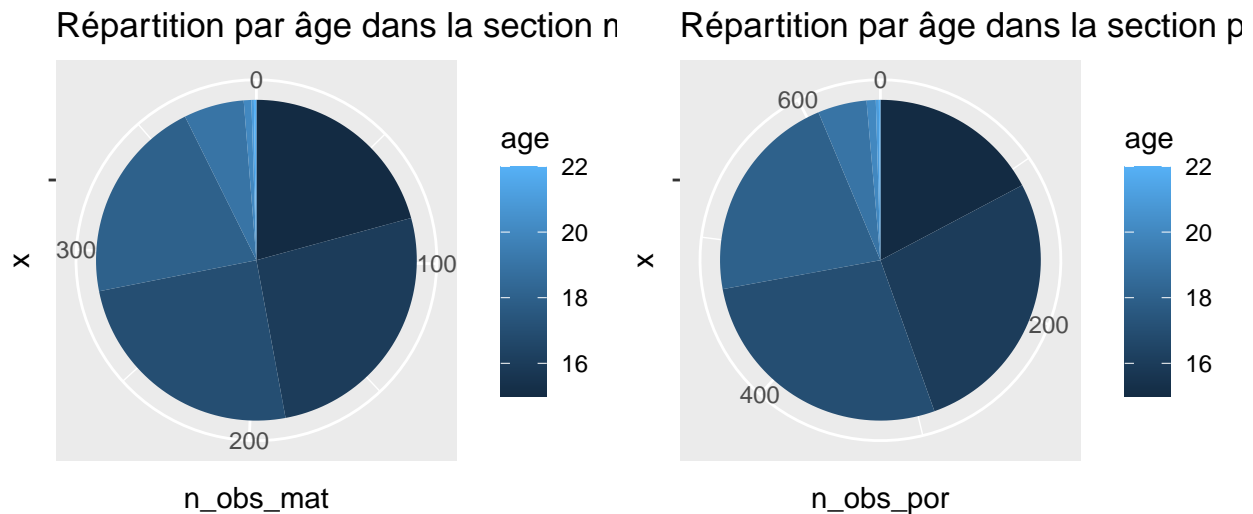
```
data_age_mat=data_quanti_mat
data_age_por=data_quanti_por

data_age_mat=summarise(group_by(data_age_mat,age),n_obs_mat=n()) #on groupe par âge avec le nombre de p
data_age_por=summarise(group_by(data_age_por,age),n_obs_por=n())

#création du camembert
p1=ggplot(data = data_age_mat, aes(x = "", y = n_obs_mat, fill = age)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Répartition par âge dans la section maths")

p2=ggplot(data = data_age_por, aes(x = "", y = n_obs_por, fill = age)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Répartition par âge dans la section portugais")
```

```
grid.arrange(p1, p2, ncol = 2)
```



On voit que la répartition semble être la grossièrement la même, en effet:

```
data_age_mat <- data_age_mat %>%
  mutate(proportion = n_obs_mat / sum(n_obs_mat))

data_age_por <- data_age_por %>%
  mutate(proportion = n_obs_por / sum(n_obs_por))

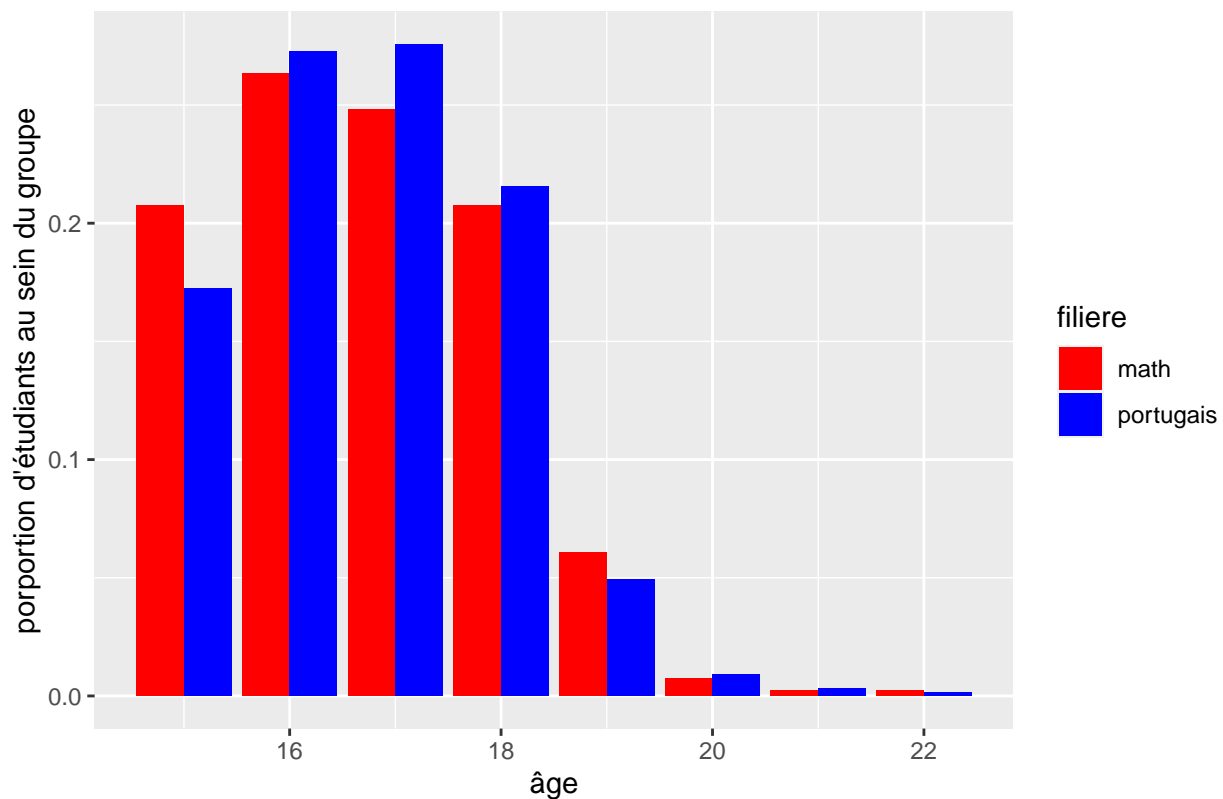
#on renome de la même manière les colonnes du nombre d'étudiants pour chaque observation
data_age_mat$filiere=c(rep("math",nrow(data_age_mat)))
data_age_por$filiere=c(rep("portugais",nrow(data_age_por)))
colnames(data_age_mat)[colnames(data_age_mat) == "n_obs_mat"] <- "n_obs"
colnames(data_age_por)[colnames(data_age_por) == "n_obs_por"] <- "n_obs"

#on concatène les deux datas frame
data_age=rbind(data_age_mat,data_age_por)

#Création du graphique

ggplot(data_age, aes(x = age, y = proportion, fill = filiere)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Comparaison des âges dans chaque filière", x ="âge", y = "porportion d'étudiants au sein")
scale_fill_manual(values = c("red", "blue"))
```

## Comparaison des âges dans chaque filière



voit que la répartition d'âge est la même dans chaque filière

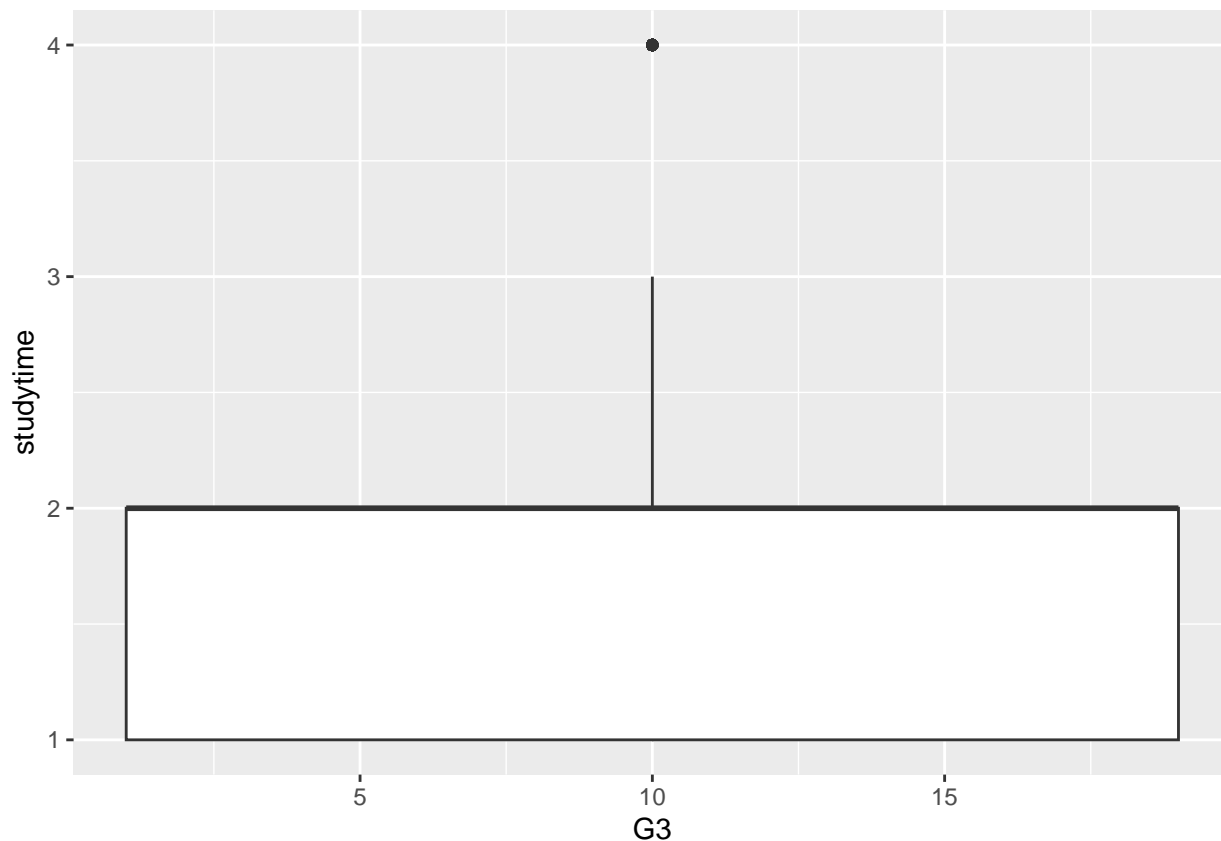
## Quantité de travail

```
data_stud=data_quanti
data_stud=summarise(group_by(data_stud,studytime),n_obs=n()) #on groupe par temps d'étude par semaine

data_stud$studytime[data_stud$studytime == 1] <- "<2 hours"
data_stud$studytime[data_stud$studytime == 2] <- "2 to 5 hours"
data_stud$studytime[data_stud$studytime == 3] <- "5 to 10 hours"
data_stud$studytime[data_stud$studytime == 4] <- ">10 hours"

ggplot(data_quanti, aes(x = G3, y = studytime)) +
  geom_boxplot()

## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



```
# ggplot(data_stud, aes(x = "", y = n_obs, fill = factor(studytime))) +
#   geom_bar(stat = "identity", width = 1) +
#   coord_polar(theta = "y") +
#   labs(title = "Répartition des temps d'étude toutes filières confondues") +
#   scale_fill_discrete(name = "Temps d'étude", labels = c("<=2 hours", "2 to 5 hours", "5 to 10 hours"))
```

On voit clairement que les étudiants travaillent majoritairement moins de 2h00 ou entre 5h00 et 10h00 par semaines.

```
#creation data frame stud pour le groupe portugais
data_stud_por=data_quant_i_por
data_stud_por=summarise(group_by(data_stud_por,studytime),n_obs_por=n()) #on groupe par temps d'étude p

data_stud_por$studytime[data_stud_por$studytime == 1] <- "<2 hours"
data_stud_por$studytime[data_stud_por$studytime == 2] <- "2 to 5 hours"
data_stud_por$studytime[data_stud_por$studytime == 3] <- "5 to 10 hours"
data_stud_por$studytime[data_stud_por$studytime == 4] <- ">10 hours"

#creation data frame stud pour le groupe mat b
data_stud_mat=data_quant_i_mat
data_stud_mat=summarise(group_by(data_stud_mat,studytime),n_obs_mat=n()) #on groupe par temps d'étude p

data_stud_mat$studytime[data_stud_mat$studytime == 1] <- "<2 hours"
data_stud_mat$studytime[data_stud_mat$studytime == 2] <- "2 to 5 hours"
data_stud_mat$studytime[data_stud_mat$studytime == 3] <- "5 to 10 hours"
data_stud_mat$studytime[data_stud_mat$studytime == 4] <- ">10 hours"

#création des camemberts pour les deux sections
```

```

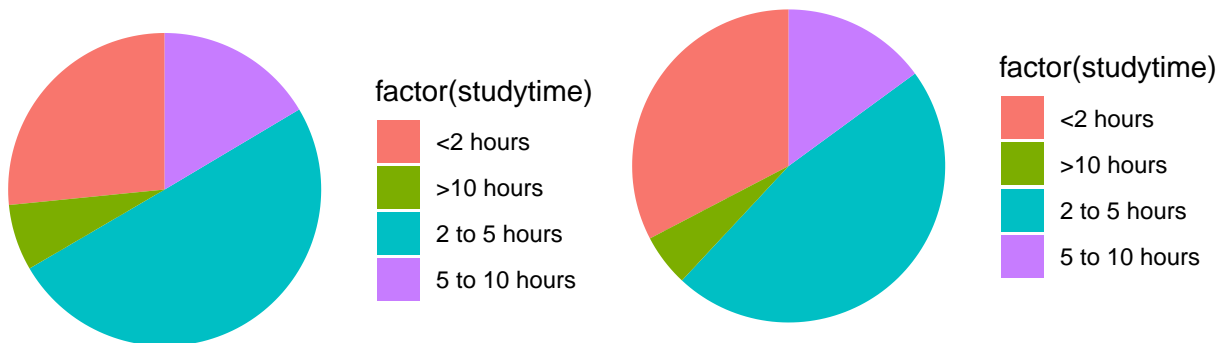
p1=ggplot() +
  # Premier camembert
  geom_bar(data = data_stud_mat, aes(x = "", y = n_obs_mat, fill = factor(studytime)), stat = "identity", coord_polar(theta = "y")) +
  theme_void() +
  labs(title = "Temps d'étude par semaine dans la section maths (à gauche) et portugaise (à droite)")

  # Deuxième camembert
p2=ggplot() +
  geom_bar(data = data_stud_por, aes(x = "", y = n_obs_por, fill = factor(studytime)), stat = "identity", coord_polar(theta = "y")) +
  theme_void()

grid.arrange(p1, p2, ncol = 2)

```

Temps d'étude par semaine dans la section maths (à gauche) et portugaise (à droite)



data\_stud\_mat

```

## # A tibble: 4 x 2
##   studytime    n_obs_mat
##   <chr>         <int>
## 1 <2 hours      105
## 2 2 to 5 hours  198
## 3 5 to 10 hours   65
## 4 >10 hours     27

```

On voit qu'il y a plus de personnes qui travaillent moins de deux heures par semaine dans la section portugaise tandis qu'il y a moins de personnes qui travaillent plus de 10h00 dans cette même section. Le nombre d'étudiants travaillant entre 5 et 10 heures semble être à peu près le même. En effet:

```

#on calcul la proportion pour pouvoir comparer
data_stud_mat <- data_stud_mat %>%
  mutate(proportion = n_obs_mat / sum(n_obs_mat))

data_stud_por <- data_stud_por %>%
  mutate(proportion = n_obs_por / sum(n_obs_por))

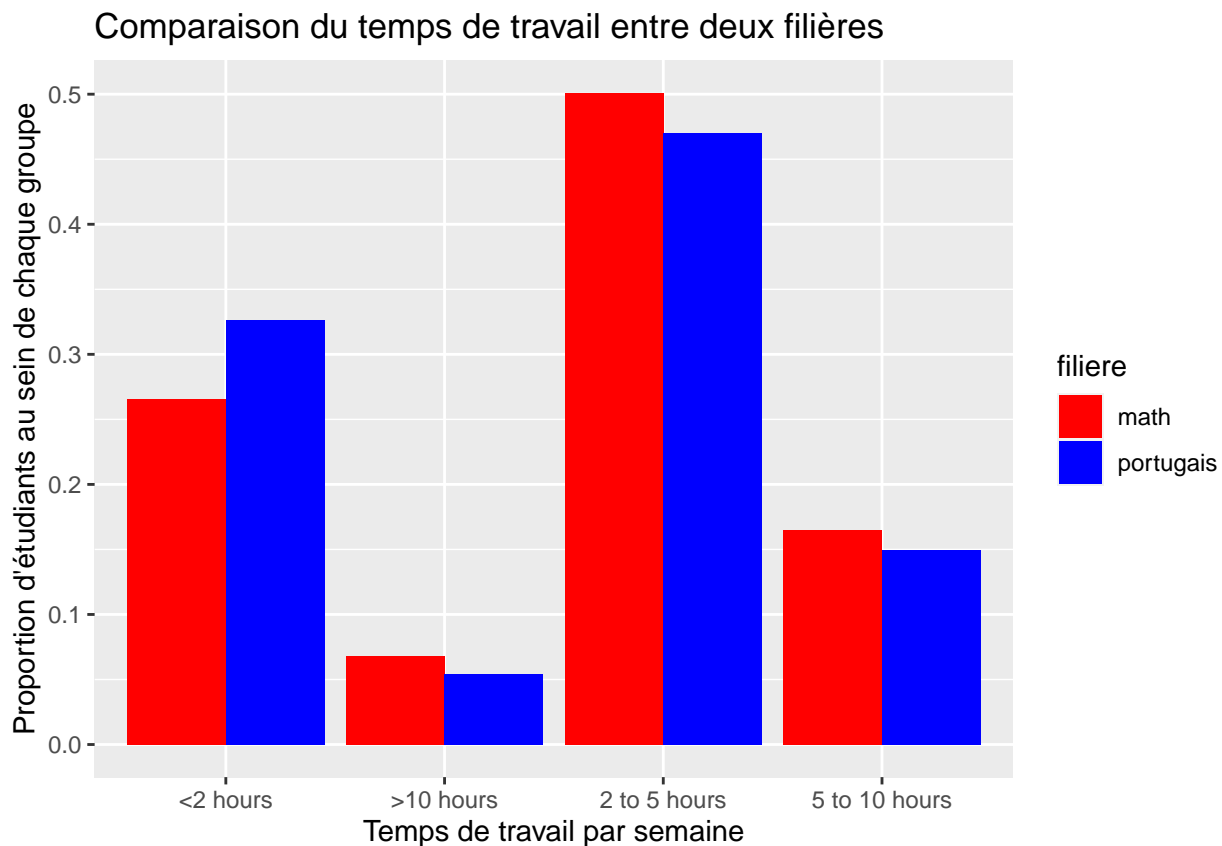
#on renome de la même manière les colonnes du nombre d'étudiants pour chaque observation
data_stud_mat$filieres=c(rep("math",nrow(data_stud_mat)))
data_stud_por$filieres=c(rep("portugais",nrow(data_stud_por)))
colnames(data_stud_mat)[colnames(data_stud_mat) == "n_obs_mat"] <- "n_obs"
colnames(data_stud_por)[colnames(data_stud_por) == "n_obs_por"] <- "n_obs"

```

```
#on concatène les deux datas frame
data_stud=rbind(data_stud_mat,data_stud_por)

#Création du graphique

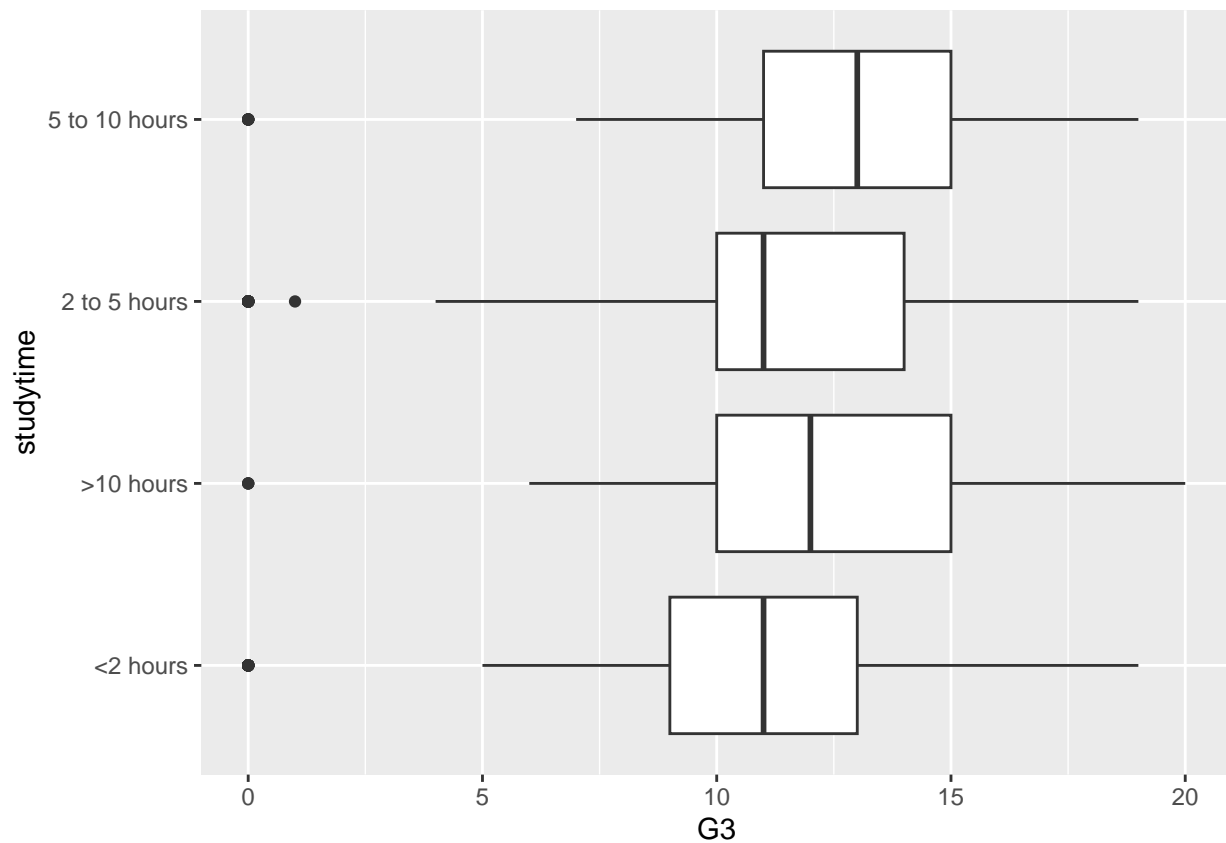
ggplot(data_stud, aes(x = studytime, y = proportion, fill = filiere)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Comparaison du temps de travail entre deux filières", x = "Temps de travail par semaine",
  scale_fill_manual(values = c("red", "blue"))
```



On s'aperçoit donc que les élèves dans la filière mathématiques travaillent plus

```
data_quanti$studytime[data_quanti$studytime == 1] <- "<2 hours"
data_quanti$studytime[data_quanti$studytime == 2] <- "2 to 5 hours"
data_quanti$studytime[data_quanti$studytime == 3] <- "5 to 10 hours"
data_quanti$studytime[data_quanti$studytime == 4] <- ">10 hours"

ggplot(data_quanti, aes(x = G3, y = studytime)) +
  geom_boxplot()
```



On voit que globalement, les élèves qui travaillent plus ont de meilleures notes (comportement bizarre à vérifier)

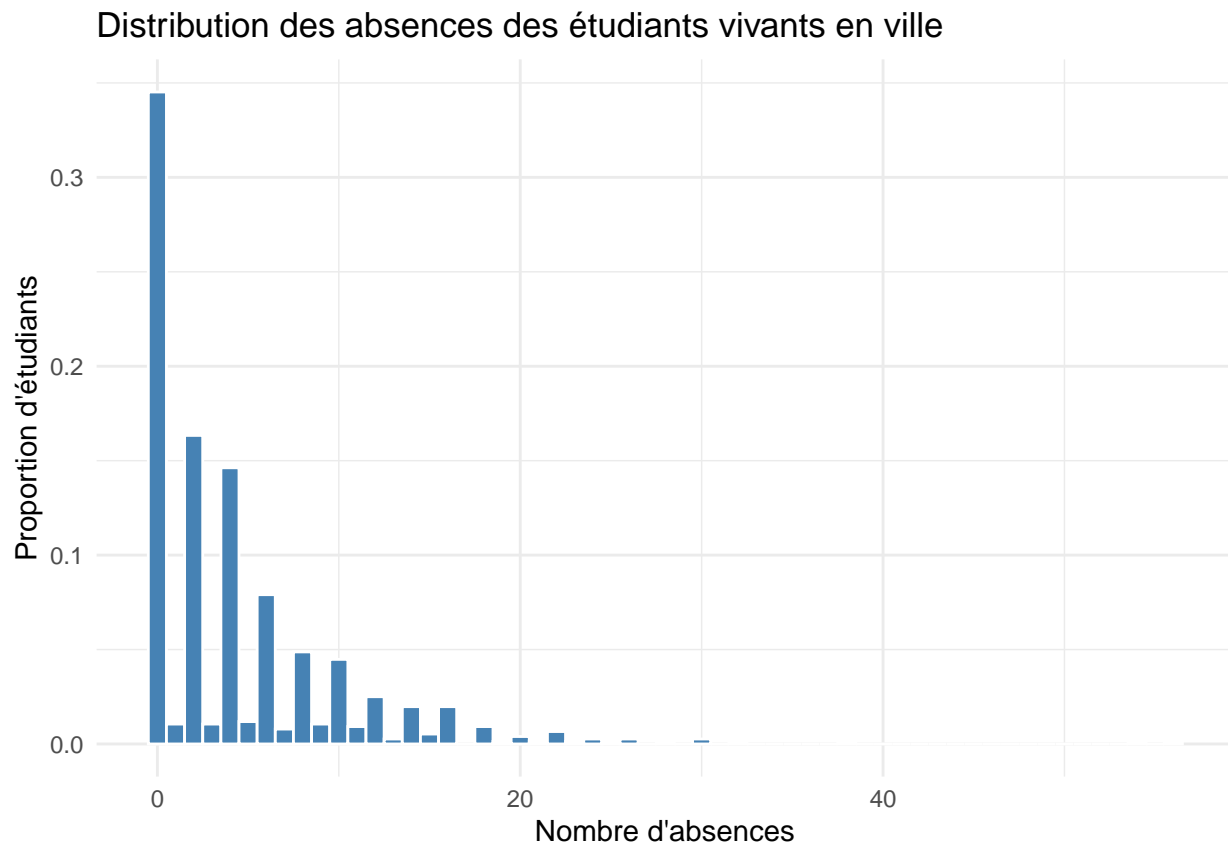
### Absences des étudiants

```
ggplot(df[df$address == 'U',], aes(x=absences)) +
  geom_histogram(aes(y = ..count.. / sum(..count..)), binwidth=1, fill="steelblue", color="white") +
  labs(title="Distribution des absences des étudiants vivants en ville",
        x="Nombre d'absences", y="Proportion d'étudiants") +
  theme_minimal()
```

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.

## i Please use `after\_stat(count)` instead.





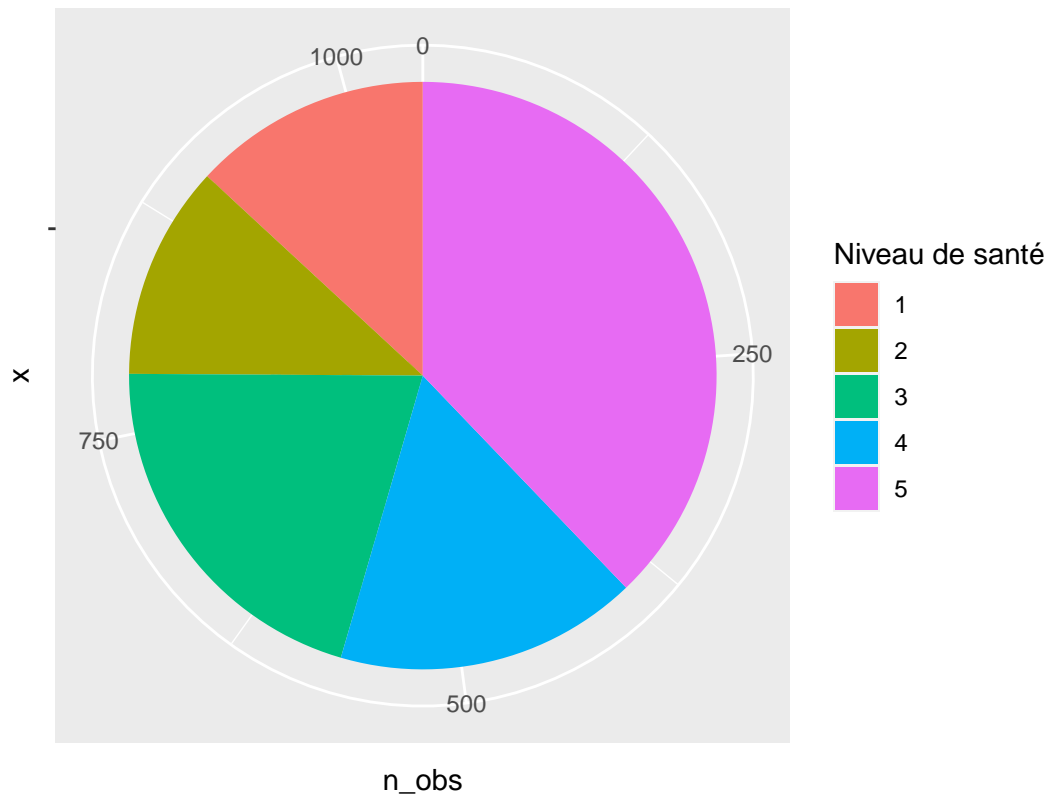
## Variables qualitatives à modalités numériques

### Santé des étudiants

```
data_health=data_quant1
data_health=summarise(group_by(data_health,health),n_obs=n())

ggplot(data_health, aes(x = "", y = n_obs, fill = factor(health))) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Santé des étudiants") +
  scale_fill_discrete(name = "Niveau de santé", labels = c(1,2,3,4,5))
```

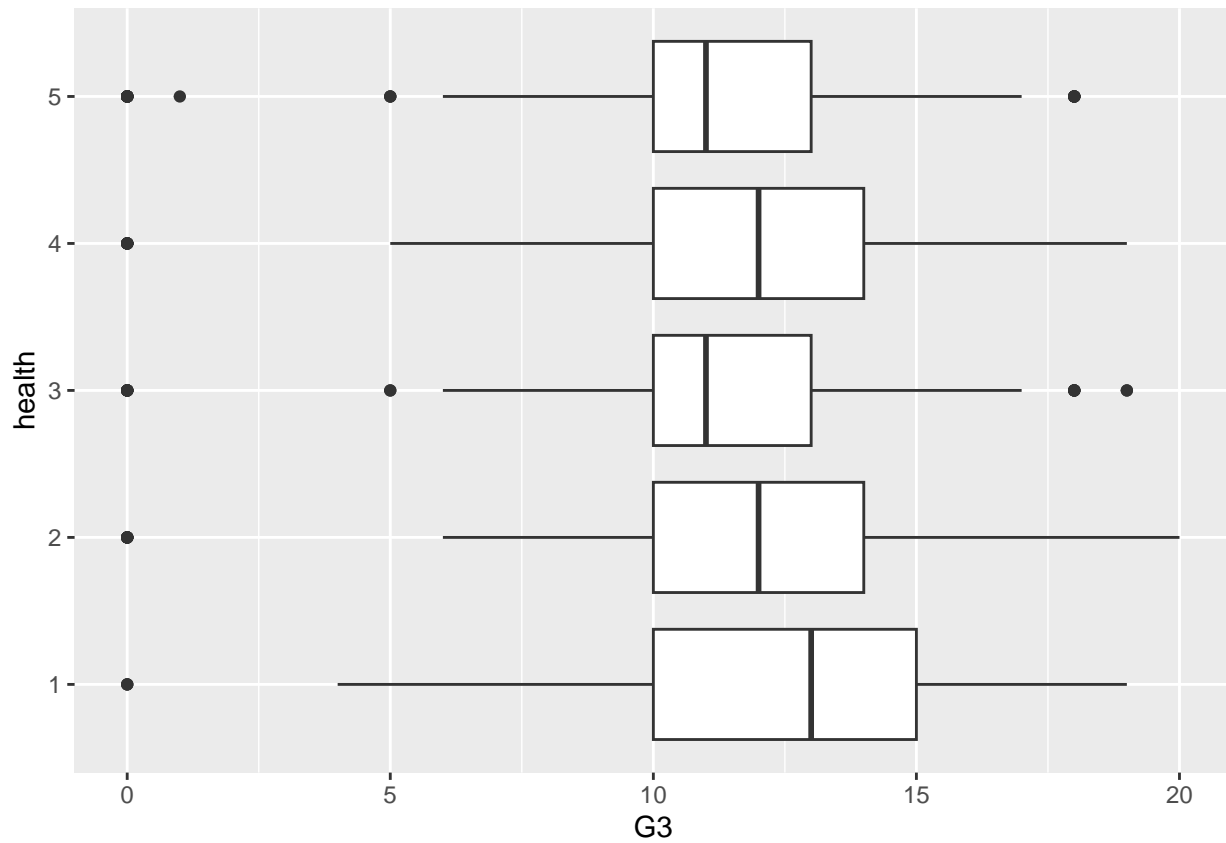
## Santé des étudiants



part des étudiant sont en bonne santé

On voit que la plu-

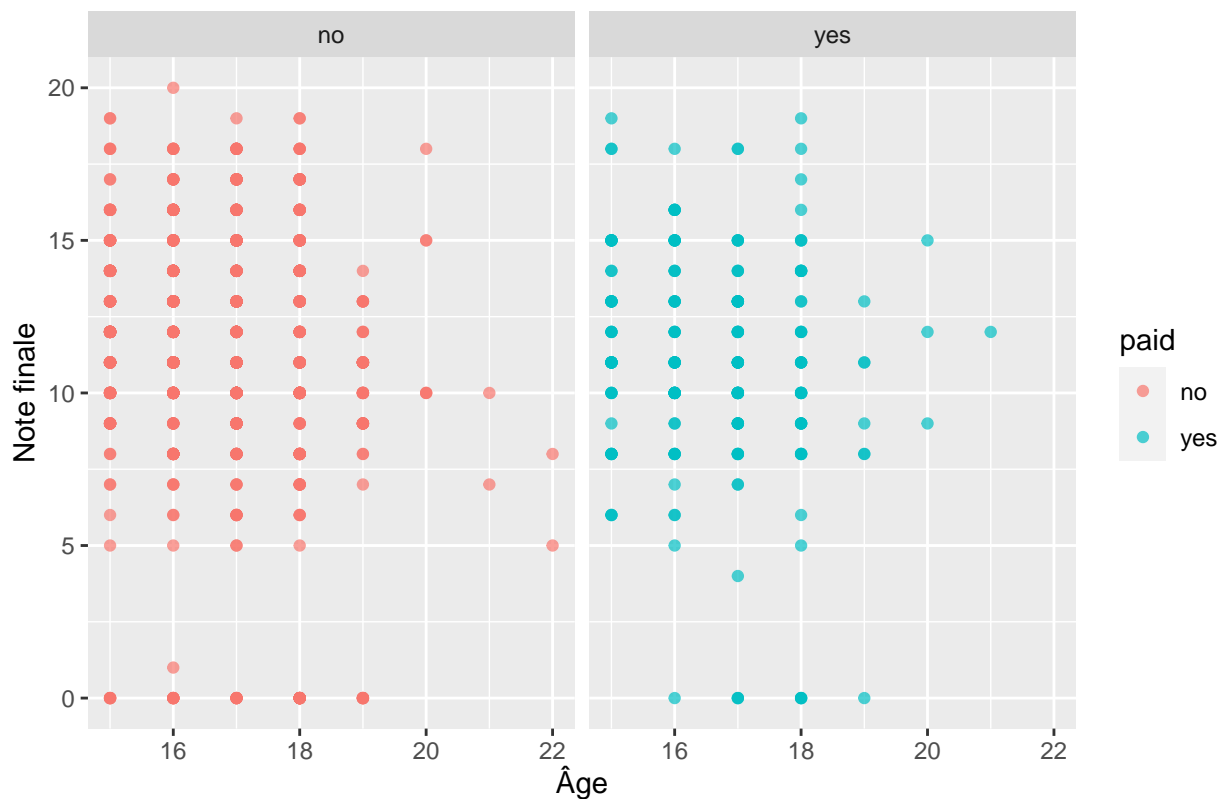
```
data_quanti$health=factor(data_quanti$health)
ggplot(data_quanti, aes(x = G3, y = health)) +
  geom_boxplot()
```



On voit que les étudiants en meilleure santé ont une meilleure réussite

```
ggplot(df, aes(x = age, y = G3, color = paid)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~paid) +
  labs(title = "Distribution de l'âge et de la note finale en fonction cours particuliers et de l'âge",
        x = "Âge", y = "Note finale")
```

Distribution de l'âge et de la note finale en fonction cours particuliers et de l'



Curieusement, les résultats semblent meilleur pour ceux qui n'ont pas pris de cours

## 4. Machine Learning : Classification de la réussite scolaire

Dans cette partie, nous nous concentrons sur la mise en place de méthodes de classification afin de prédire la variable RS (réussite scolaire). Nous nous intéresserons essentiellement à la comparaison des résultats de chacune des méthodes. Les méthodes utilisées seront évaluées avec leur accuracy et leur courbe ROC.

### a) Séparation du jeu de données

Ici, nous découpons notre dataset en jeu d'entraînement et jeu de test. Le ratio utilisé est  $\frac{1}{5}$  pour le jeu de test. Tout d'abord on modifie notre jeu de données pour le préparer pour la classification en retirant les notes.

```
# Suppression des colonnes
X = subset(df, select = -c(G1,G2,G3,Moy) )

set.seed(1)
n <- nrow(X)
p <- ncol(X)-1
test.ratio <- .2 # ratio of test/train samples
n.test <- round(n*test.ratio)
n.test

## [1] 209

tr <- sample(1:n,n.test)
df.test <- X[tr,]
df.train <- X[-tr,]
```

## b) LDA

```
res_lda=lda(df.train$RS ~., data=df.train)
pred_lda <- predict(res_lda,newdata=df.test)$posterior[,2]
```

*# Table de confusion*

```
table(df.test$RS,predict(res_lda,newdata=df.test)$class)
```

```
##
##          admis exclusion redoublement
## admis          137         1         5
## exclusion        17        14         2
## redoublement     23         4         6
```

*# Courbe ROC*

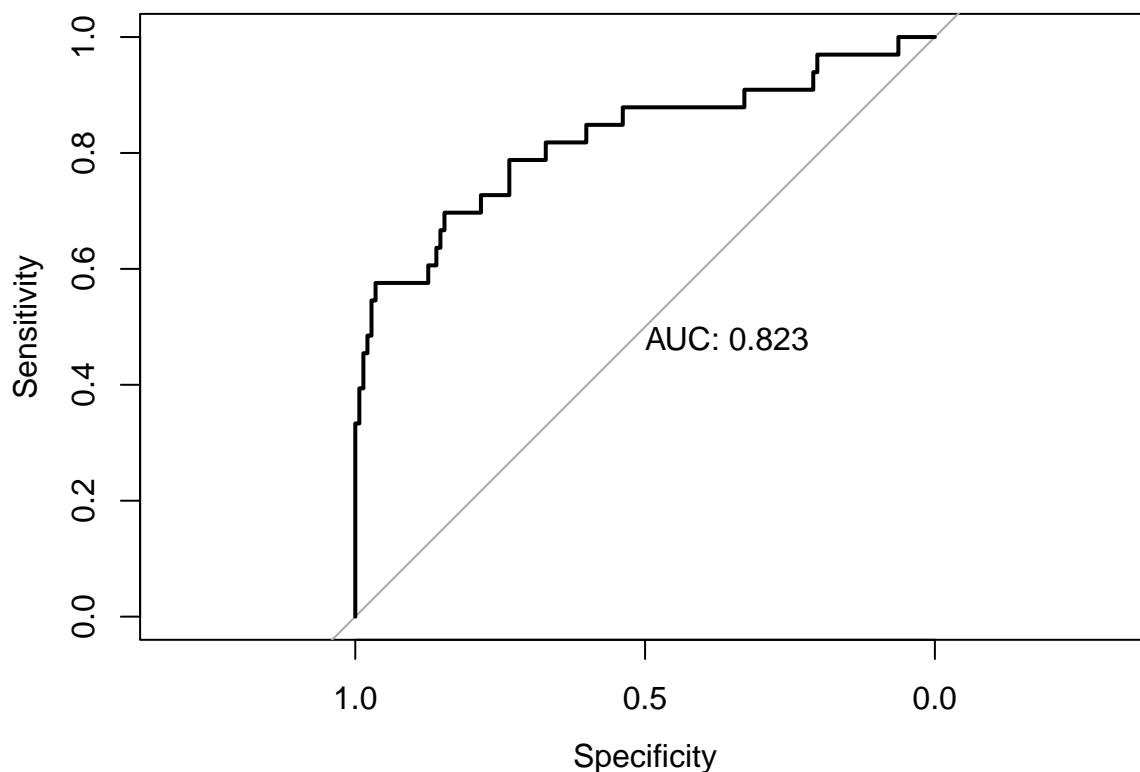
```
ROC_lda <- roc(df.test$RS, pred_lda)
```

```
## Warning in roc.default(df.test$RS, pred_lda): 'response' has more than two
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
```

```
## Setting levels: control = admis, case = exclusion
```

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

```
plot(ROC_lda, print.auc=TRUE, print.auc.y = 0.5)
```



```
ROC_lda$auc
```

```
## Area under the curve: 0.8226
```

*# Accuracy*

```
accuracy_lda = mean(df.test$RS==predict(res_lda,newdata=df.test)$class)
print("accuracy lda = ")
```

```
## [1] "accuracy lda = "  
print(accuracy_lda)
```

```
## [1] 0.7511962
```

### c) QDA

```
res_qda = qda(df.train$RS~., data=df.train)  
pred_qda <- predict(res_qda,newdata=df.test)$posterior[,2]
```

```
# Table de confusion  
table(df.test$RS,predict(res_qda,newdata=df.test)$class)
```

```
##  
##          admis exclusion redoublement  
##  admis          128         12         3  
##  exclusion        12         17         4  
##  redoublement     19          7         7
```

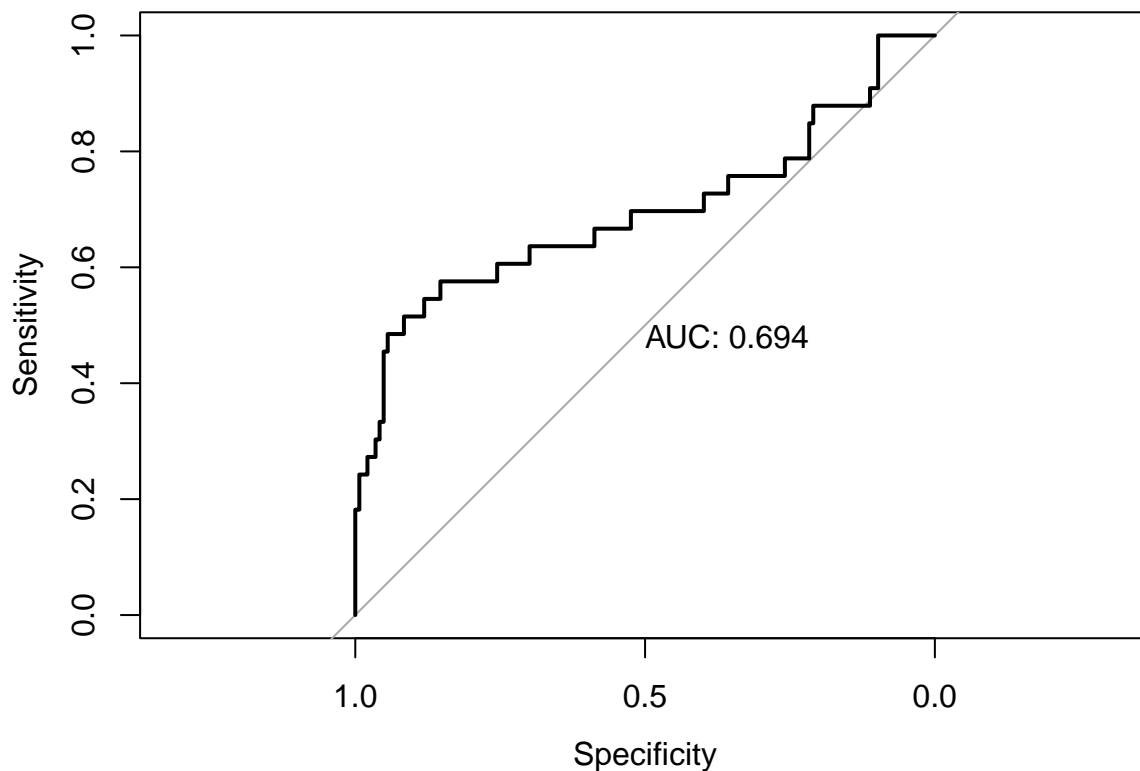
```
# Courbe ROC  
ROC_qda <- roc(df.test$RS, pred_qda)
```

```
## Warning in roc.default(df.test$RS, pred_qda): 'response' has more than two  
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
```

```
## Setting levels: control = admis, case = exclusion
```

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

```
plot(ROC_qda, print.auc=TRUE, print.auc.y = 0.5)
```



```
ROC_qda$auc
```

```
## Area under the curve: 0.6944
```

```
# Accuracy
```

```
accuracy_qda = mean(df.test$RS==predict(res_qda,newdata=df.test)$class)
```

```
print("accuracy qda = ")
```

```
## [1] "accuracy qda = "
```

```
print(accuracy_qda)
```

```
## [1] 0.7272727
```

## d) Stepwise

```
stepwise_lda=stepclass(RS~., data=df.train, method="lda", direction="backward")
```

```
## `stepwise classification', using 10-fold cross-validated correctness rate of method lda'.
```

```
## 835 observations of 30 variables in 3 classes; direction: backward
```

```
## stop criterion: improvement less than 5%.
```

```
## Warning in cv.rate(vars = start.vars, data = data, grouping = grouping, :  
## error(s) in modeling/prediction step
```

```
## correctness rate: 0; starting variables (30): school, sex, age, address, famsize, Pstatus, Medu, Fe
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```

```
## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,  
## : error(s) in modeling/prediction step
```





```

## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,
## : error(s) in modeling/prediction step

## Warning in cv.rate(trymodel, data = data, grouping = grouping, method = method,
## : error(s) in modeling/prediction step

##
##   hr.elapsed min.elapsed sec.elapsed
##      0.000      0.000      2.441
stepwise_lda

## method      : lda
## final model : RS ~ school + sex + age + address + famsize + Pstatus + Medu +
##   Fedu + Mjob + Fjob + reason + guardian + traveltime + studytime +
##   failures + schoolsup + famsup + paid + activities + nursery +
##   higher + internet + romantic + famrel + freetime + goout +
##   Dalc + Walc + health + absences
## <environment: 0x5643f0e0ce88>
##
## correctness rate = 0
res_stepwise_lda = lda(stepwise_lda$formula, data=df.train)

pred_lda_step <- predict(res_stepwise_lda,newdata=df.test)$posterior[,2]

# Table de confusion
table(df.test$RS, predict(res_stepwise_lda,newdata=df.test)$class)

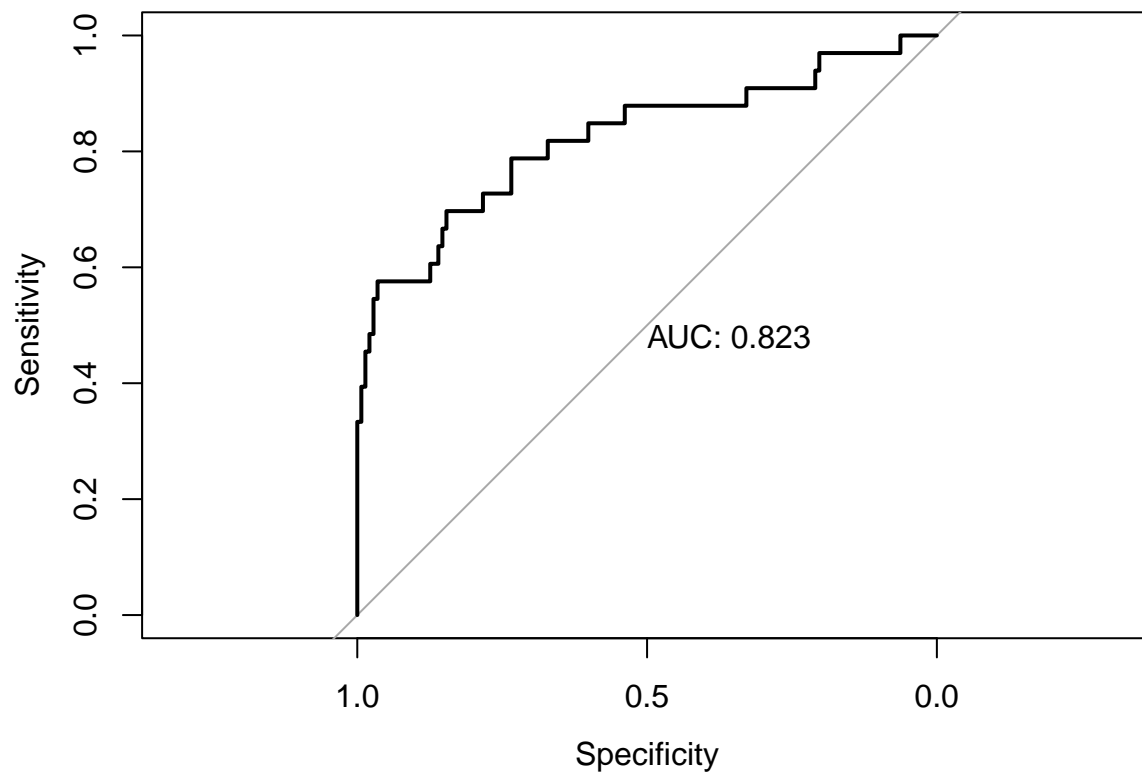
##
##           admis exclusion redoublement
##   admis           137          1          5
##   exclusion          17         14          2
##   redoublement        23          4          6

# Courbe ROC
ROC_lda_step <- roc(df.test$RS, pred_lda)

## Warning in roc.default(df.test$RS, pred_lda): 'response' has more than two
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead

## Setting levels: control = admis, case = exclusion
## Setting direction: controls < cases
plot(ROC_lda_step, print.auc=TRUE, print.auc.y = 0.5)

```



```
ROC_lda_step$auc
```

```
## Area under the curve: 0.8226
```

```
# Accuracy
```

```
accuracy_lda_stepwise = mean(df.test$RS== predict(res_stepwise_lda,newdata=df.test)$class)
print("accuracy lda stepwise = ")
```

```
## [1] "accuracy lda stepwise = "
```

```
print(accuracy_lda_stepwise)
```

```
## [1] 0.7511962
```

## e) Random Forest

```
res_RF <- randomForest(RS~.,df.train)
res_RF
```

```
##
```

```
## Call:
```

```
## randomForest(formula = RS ~ ., data = df.train)
```

```
##           Type of random forest: classification
```

```
##           Number of trees: 500
```

```
## No. of variables tried at each split: 5
```

```
##
```

```
##           OOB estimate of  error rate: 31.98%
```

```
## Confusion matrix:
```

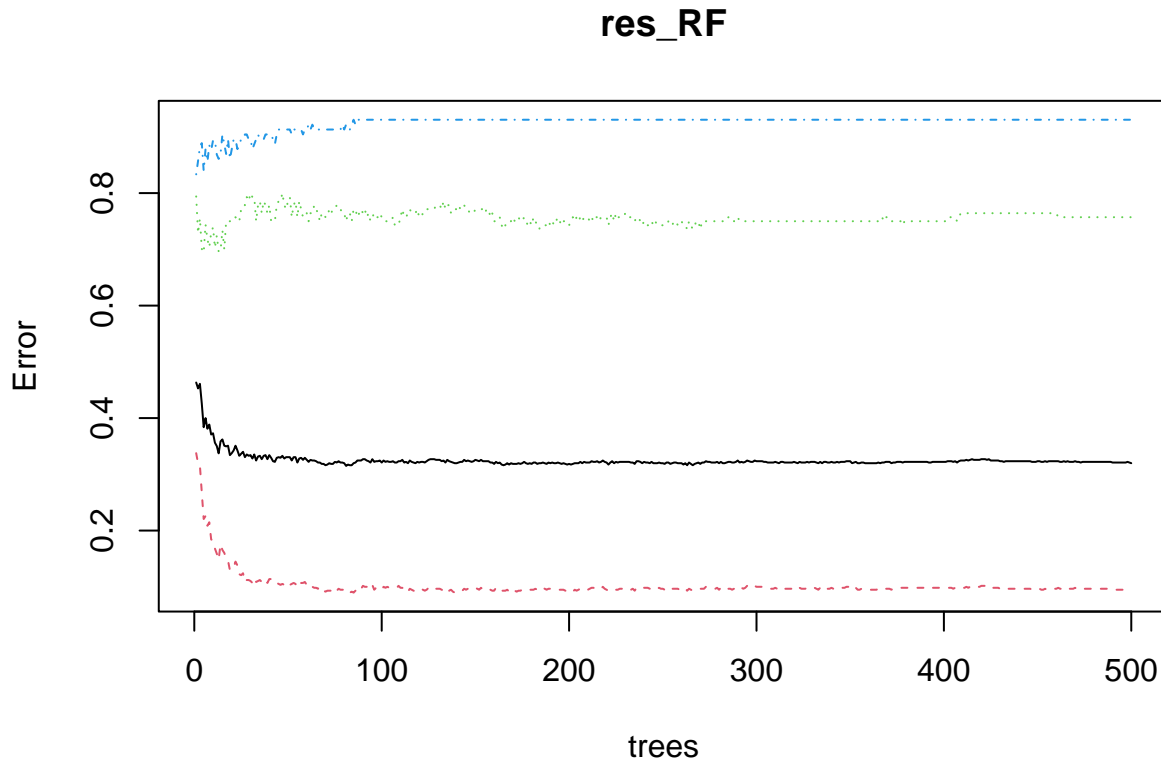
```
##           admis exclusion redoublement class.error
```

```
## admis           526          34           20 0.09310345
```

```
## exclusion          95          34           11 0.75714286
```

```
## redoublement      87      20      8 0.93043478
```

```
plot(res_RF)
```



```
## prédiction :
```

```
pred_RF <- predict(res_RF,newdata=df.test)
```

```
## Table confusion et accuracy :
```

```
table(df.test$RS, predict(res_RF,newdata=df.test,type="class"))
```

```
##
```

```
##      admis exclusion redoublement
```

```
##  admis      135      7      1
```

```
##  exclusion    16     14     3
```

```
##  redoublement  26      6      1
```

```
## aire sous courbe ROC
```

```
pred_RF = predict(res_RF, df.test, type="prob")[,2]
```

```
ROC_RF <- roc(df.test$RS, pred_RF)
```

```
## Warning in roc.default(df.test$RS, pred_RF): 'response' has more than two
```

```
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
```

```
## Setting levels: control = admis, case = exclusion
```

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

```
ROC_RF$auc
```

```
## Area under the curve: 0.8535
```

```
## Accuracy
```

```
accuracy_RF = mean(df.test$RS==predict(res_RF,newdata=df.test,type="class"))
```

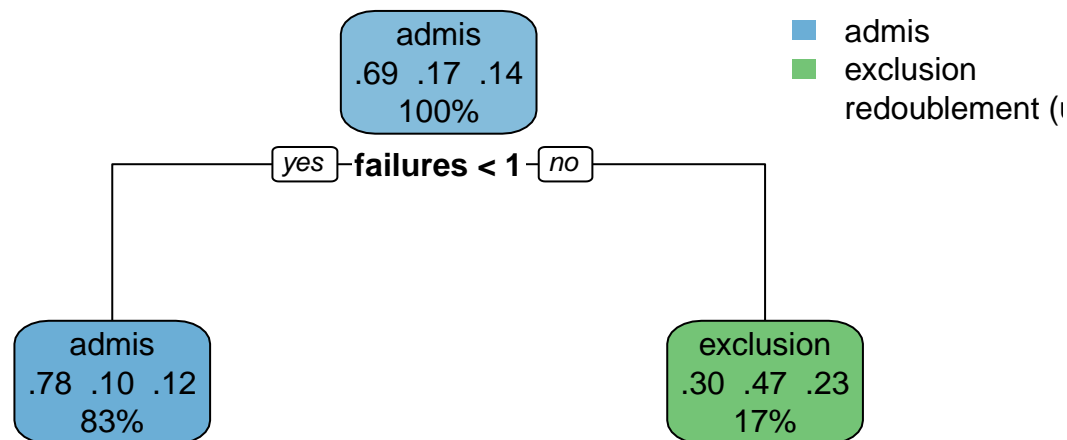
```
print("accuracy RF = ")
```

```
## [1] "accuracy RF = "
print(accuracy_RF)
```

```
## [1] 0.7177033
```

## f) CART

```
arbre = rpart(df.train$RS~.,df.train,control=rpart.control(minsplit=5,cp=0.025))
cp.opt = arbre$cptable[which.min(arbre$cptable[, "xerror"]), "CP"]
res_cart = prune(arbre,cp=cp.opt)
rpart.plot(res_cart)
```



```
## prédiction :
pred_cart <- predict(res_cart,newdata=df.test)[,2]
```

```
## Table confusion et accuracy :
table(df.test$RS, predict(res_cart,newdata=df.test,type="class"))
```

```
##
##          admis exclusion redoublement
##  admis          133         10          0
##  exclusion         13         20          0
##  redoublement        24          9          0
```

```
## aire sous courbe ROC
pred_cart = predict(res_cart, df.test, type="prob")[,2]
ROC_cart <- roc(df.test$RS, pred_cart)
```

```
## Warning in roc.default(df.test$RS, pred_cart): 'response' has more than two
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
## Setting levels: control = admis, case = exclusion
## Setting direction: controls < cases
```

```
ROC_cart$auc
```

```
## Area under the curve: 0.7681
```

```
## Accuracy
accuracy_cart = mean(df.test$RS==predict(res_cart,newdata=df.test,type="class"))
print("accuracy cart = ")
```

```
## [1] "accuracy cart = "
print(accuracy_cart)
```

```
## [1] 0.7320574
```

h) Adaboost

```
# fit.adaboost=gbm(as.numeric(RS)-1 ~., df.train, distribution = "adaboost")
# fit.adaboost
#
# ### Calibrer B=n.tree par cross-validation :
# fit.adaboost=gbm(as.numeric(RS)-1 ~., df.train, distribution = "adaboost",cv.folds = 5, shrinkage = 0
# gbm.perf(fit.adaboost)
# B.opt = gbm.perf(fit.adaboost, method="cv")
#
# ## prédiction :
# pred_adaboost = predict(fit.adaboost, newdata=df.test, type = "response", n.trees = B.opt)
# class = 1*(pred_adaboost>1/2)
#
# ## Table confusion et accuracy :
# table(df.test$RS, class)
#
# ## Accuracy
# accuracy_adaboost = mean(as.numeric(df.test$RS)-1==class)
# print("accuracy adaboost = ")
# print(accuracy_adaboost)
#
# ## aire sous courbe ROC
# ROC_adaboost <- roc(df.test$RS, pred_adaboost)
# ROC_adaboost$auc
```

## i) Regression Logistique

```
### Modèle
logit.train <- glm(RS ~ ., family = binomial , data=df.train)
```

```
## prédiction :
pred_logit <- predict(logit.train,newdata=df.test)
class = 1*(pred_logit>1/2)
```

```
## Table confusion et accuracy :
table(df.test$RS, class)
```

```
##           class
##           0   1
##   admis      137  6
##   exclusion   19 14
##   redoublement 23 10
```

```
## aire sous courbe ROC
ROC_logit <- roc(df.test$RS, pred_logit)
```

```
## Warning in roc.default(df.test$RS, pred_logit): 'response' has more than two
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
## Setting levels: control = admis, case = exclusion
```

```

## Setting direction: controls < cases
## Accuracy
accuracy_logit = mean(as.numeric(df.test$RS)-1==class)
print("accuracy regression logistique = ")

## [1] "accuracy regression logistique = "
print(accuracy_logit)

## [1] 0.722488
ROC_logit$auc

## Area under the curve: 0.8339
# # régression logistique Lasso
# library(glmnet)
# res_Lasso <- glmnet(as.matrix(df.train[,-1]),df.train$RS,family='binomial')
# plot(res_Lasso, label = TRUE) # en abscisse : norme des coefficients
# plot(res_Lasso, xvar = "lambda", label = TRUE) # en abscisse : log(lambda)
# # sum(coef(res_Lasso, s=exp())!=0)
#
# cvLasso <- cv.glmnet(as.matrix(df.train[,-1]),df.train$RS,family="binomial", type.measure = "class")
# plot(cvLasso)
# cvLasso$lambda.min
# coef(res_Lasso, s=cvLasso$lambda.min)
#
# #prédiction
# class_logit_lasso=predict(cvLasso, newx = as.matrix(df.test[,-1]), s = 'lambda.min', type = "class")
#
# #Table de confusion et accuracy
# table(df.test$RS, class_logit_lasso)
# pred_logit_lasso=predict(cvLasso, newx = as.matrix(df.test[,-1]), s = 'lambda.min', type = "response")
#
# accuracy_logit_lasso = mean(df.test$RS==class_logit_lasso)
# print("accuracy regression logistique lasso= ")
# print(accuracy_logit_lasso)
#
# #pred_logit_lasso
# ROC_logit_lasso = roc( df.test$RS, pred_logit_lasso)
# ROC_logit_lasso$auc

```

## Comparaison

```

result=matrix(NA, ncol=5, nrow=2)
rownames(result)=c('accuracy', 'AUC')
colnames(result)=c('lda', 'qda', 'cart', 'RF',"logit")
result[1,]= c(accuracy_lda, accuracy_qda, accuracy_cart, accuracy_RF,accuracy_logit)
result[2,]=c(ROC_lda$auc, ROC_qda$auc, ROC_cart$auc, ROC_RF$auc,ROC_logit$auc)
result

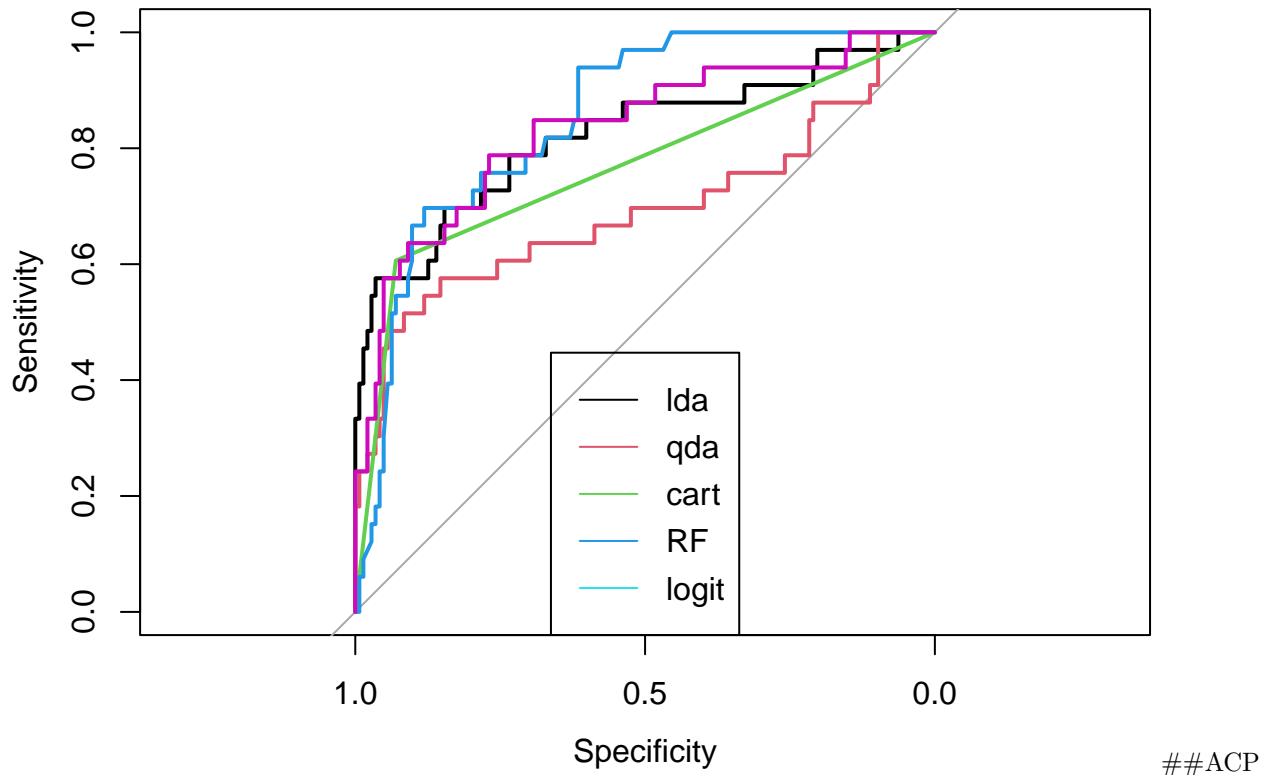
##              lda      qda      cart      RF      logit
## accuracy 0.7511962 0.7272727 0.7320574 0.7177033 0.7224880
## AUC      0.8226319 0.6944268 0.7680653 0.8534647 0.8338631

```

```
apply(result,1, which.max )
```

```
## accuracy      AUC
##           1      4
```

```
plot(ROC_lda, xlim=c(1,0))
plot(ROC_qda, add=TRUE, col=2)
plot(ROC_cart, add=TRUE, col=3)
plot(ROC_RF, add=TRUE, col=4)
plot(ROC_logit, add=TRUE, col=6)
legend('bottom', col=1:5, paste(c('lda', 'qda', 'cart', 'RF',"logit")), lwd=1)
```



```
data_quanti=df[,c(3,13,14,15,31,32,33,34)]
head(data_quanti)
```

```
##   age traveltime studytime failures G1 G2 G3      Moy
## 1  18          2         2        0  5  6  6  5.666667
## 2  17          1         2        0  5  5  6  5.333333
## 3  15          1         2        3  7  8 10  8.333333
## 4  15          1         3        0 15 14 15 14.666667
## 5  16          1         2        0  6 10 10  8.666667
## 6  16          1         2        0 15 15 15 15.000000
```

On va par la suite transformer lorsque cela est possible certaines variables qualitatives en variables quantitatives afin de pouvoir réaliser une ACP dessus. Pour les variables studytime et traveltime, des intervalles nous sont données, on prend donc pour chaque niveau le milieu de l'intervalle. Pour les valeurs extrêmes, 1 et 4, on choisit arbitrairement une borne supérieure ou inférieure (15H00 pour studytime et 3h00 pour traveltime pour ce qu'il s'agit des bornes supérieures et 0h00 pour les deux bornes inférieures)

```
##on convertit studytime et travel time en variables quantitatives (on prend le milieu des segments)
for (i in 1:nrow(data_quanti)){
```

```

if (data_quanti$studytime[i]==2){
  data_quanti$studytime[i]=210
}
if (data_quanti$studytime[i]==1){
  data_quanti$studytime[i]=120
}
if (data_quanti$studytime[i]==3){
  data_quanti$studytime[i]=450
}
if (data_quanti$studytime[i]==4){
  data_quanti$studytime[i]=750
}
if(data_quanti$travelttime[i]==1){
  data_quanti$travelttime[i]=7.5
}
if(data_quanti$travelttime[i]==2){
  data_quanti$travelttime[i]=22.5
}
if(data_quanti$travelttime[i]==3){
  data_quanti$travelttime[i]=45
}
if(data_quanti$travelttime[i]==4){
  data_quanti$travelttime[i]=120
}

}
head(data_quanti)

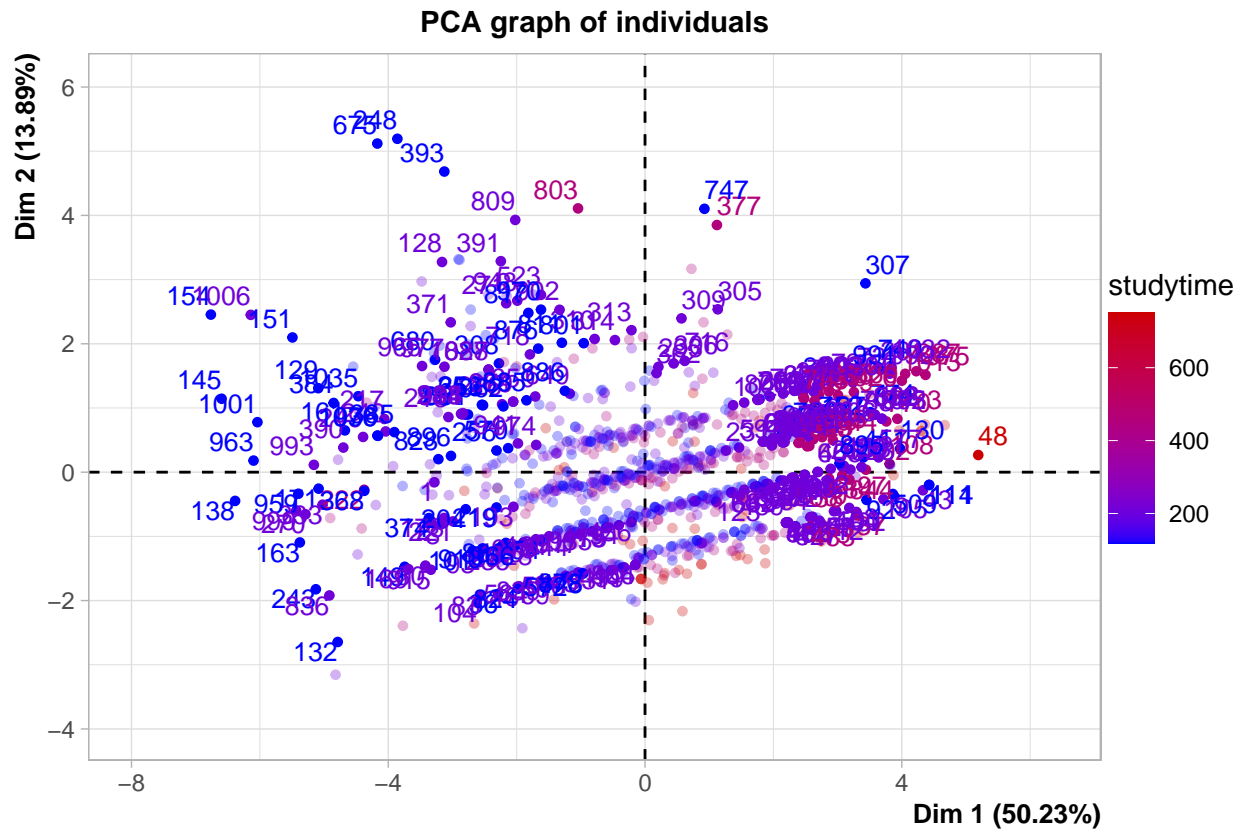
##   age travelttime studytime failures G1 G2 G3      Moy
## 1  18         22.5        210         0  5  6  6  5.666667
## 2  17          7.5        210         0  5  5  6  5.333333
## 3  15          7.5        210         3  7  8 10  8.333333
## 4  15          7.5        450         0 15 14 15 14.666667
## 5  16          7.5        210         0  6 10 10  8.666667
## 6  16          7.5        210         0 15 15 15 15.000000

res=PCA(data_quanti)

```

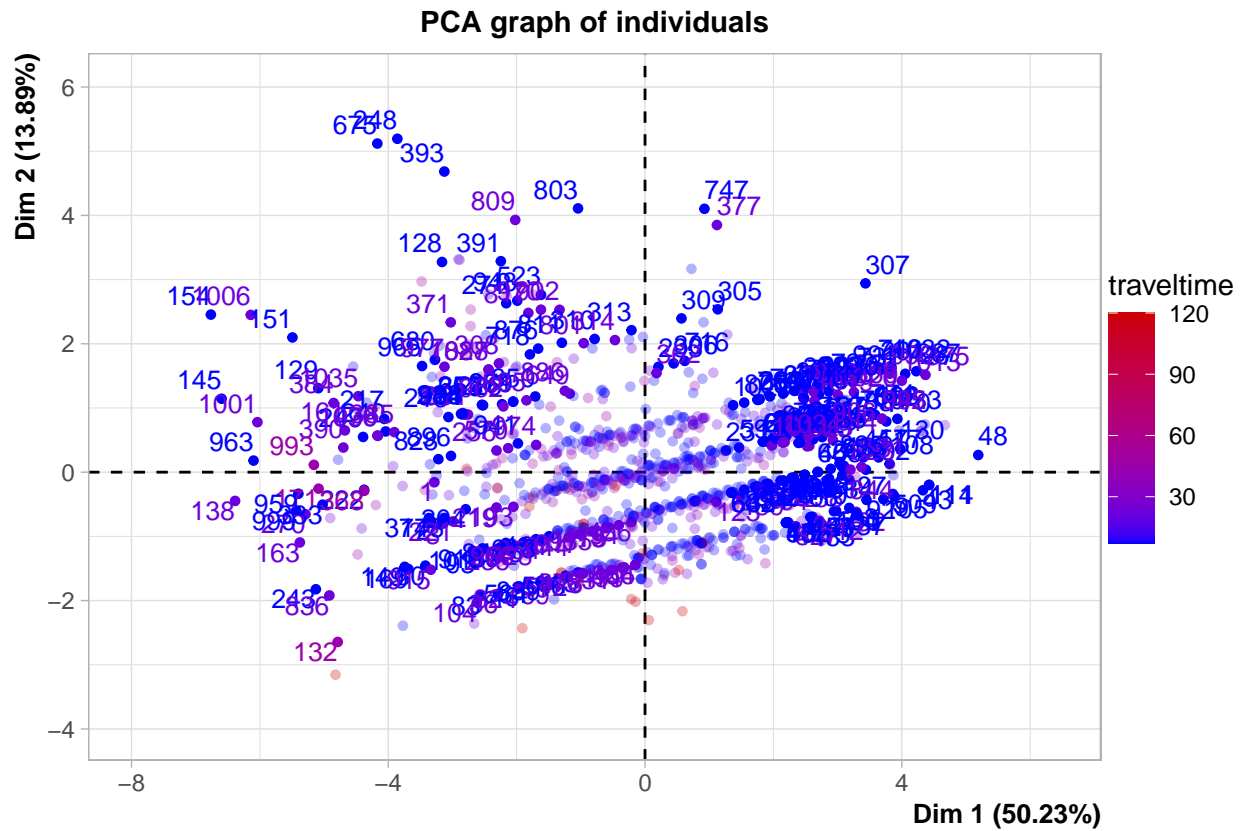






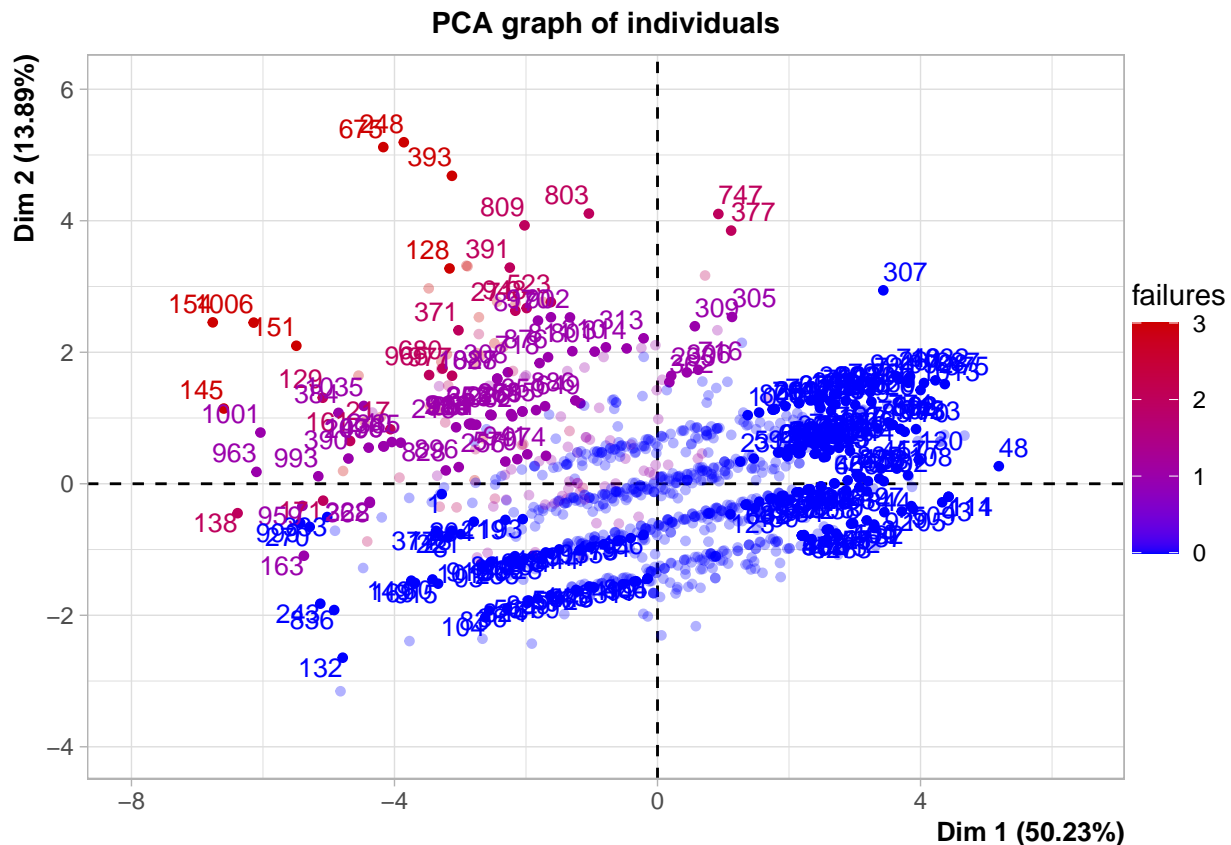
On retrouve bien que les personnes ayant le plus travaillé se situent du côté des bonnes notes.

```
plot(res, select="cos2 0.8", habillage=2, cex=0.9, choix="ind") #visualisation du temps de trajet
```



De la même manière on voit que le temps de trajet a une influence négative sur la réussite.

```
plot(res, select="cos2 0.8", habillage=4, cex=0.9, choix="ind") #visualisation des échecs
```



On voit aussi que les élèves qui ont les meilleurs résultats sont ceux qui ont le moins d'échecs. Par ailleurs l'acp ici ne semble pas très pertinente car la plupart des variables du jeu de données sont quantitatives, nous avons donc été obligés de les rendre (lorsque cela a un sens) qualitatives. Néanmoins on voit par exemple que pour ces variables transformées, leur projection est très mauvaise et ne peuvent donc pas être interprétés à l'aide de l'ACP (comme studytime et traveltime). Egalement peut être qu'il y a une meilleure de les rendre qualitatives. C'est pour quoi l'on va réaliser par la suite un anova 2 sur les variables quatitatives studytime et traveltime afin de pouvoir expliqués la variable Moy avec. ##Anova 2 sur les variables studytime et traveltime

```
#création de la data frame correspondante
data_anova=df[,c(13,14,34)]
data_anova$traveltime=factor(data_anova$traveltime)
data_anova$studytime=factor(data_anova$studytime)
attach(data_anova)
```

## Les objets suivants sont masqués depuis data\_quantiti:

```
##
##   studytime, traveltime
```

```
head(data_anova)
```

```
##   traveltime studytime      Moy
## 1          2         2  5.666667
## 2          1         2  5.333333
## 3          1         2  8.333333
## 4          1         3 14.666667
## 5          1         2  8.666667
## 6          1         2 15.000000
```

```
table(data_anova$traveltime,data_anova$studytime)
```

```
##
##      1  2  3  4
##  1 165 314 108 36
##  2 110 143  46 21
##  3  34  37   4  2
##  4   8   9   4  3
```

Le plan est trop déséquilibré pour faire un anova ##Anova 2 sur les variables romantic et Walc

```
#création de la data frame correspondante
head(df)
```

```
##  school sex age address famsize Pstatus Medu Fedu Mjob Fjob reason
## 1    GP  F  18      U    GT3      A    4    4 at_home teacher course
## 2    GP  F  17      U    GT3      T    1    1 at_home  other course
## 3    GP  F  15      U    LE3      T    1    1 at_home  other  other
## 4    GP  F  15      U    GT3      T    4    2 health services home
## 5    GP  F  16      U    GT3      T    3    3 other  other  home
## 6    GP  M  16      U    LE3      T    4    3 services other reputation
##  guardian traveltime studytime failures schoolsup famsup paid activities
## 1  mother          2          2          0      yes    no    no          no
## 2  father          1          2          0      no     yes    no          no
## 3  mother          1          2          3      yes    no    yes          no
## 4  mother          1          3          0      no     yes    yes          yes
## 5  father          1          2          0      no     yes    yes          no
## 6  mother          1          2          0      no     yes    yes          yes
##  nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1    yes    yes      no      no      4          3    4    1    1    3
## 2    no     yes      yes      no      5          3    3    1    1    3
## 3    yes    yes      yes      no      4          3    2    2    3    3
## 4    yes    yes      yes      yes      3          2    2    1    1    5
## 5    yes    yes      no      no      4          3    2    1    2    5
## 6    yes    yes      yes      no      5          4    2    1    2    5
##  absences G1 G2 G3      Moy      RS
## 1      6  5  6  6  5.666667 exclusion
## 2      4  5  5  6  5.333333 exclusion
## 3     10  7  8 10  8.333333 exclusion
## 4      2 15 14 15 14.666667 admis
## 5      4  6 10 10  8.666667 redoublement
## 6     10 15 15 15 15.000000 admis
```

```
data_anova=df[,c(28,23,34)]
data_anova$Walc=factor(data_anova$Walc)
head(data_anova)
```

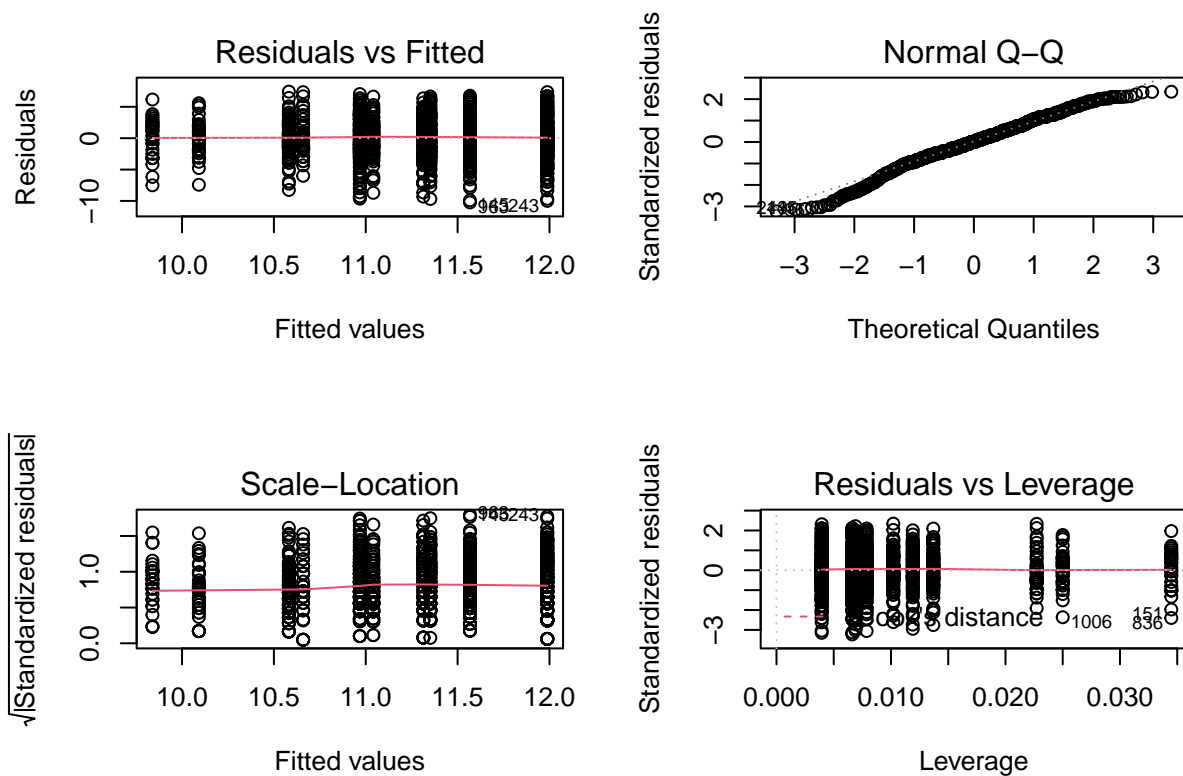
```
##  Walc romantic      Moy
## 1    1      no  5.666667
## 2    1      no  5.333333
## 3    3      no  8.333333
## 4    1     yes 14.666667
## 5    2      no  8.666667
## 6    2      no 15.000000
```

```
table(data_anova$Walc,data_anova$romantic)
```

```
##
##      no yes
##    1 253 145
##    2 151  84
##    3 127  73
##    4  98  40
##    5  44  29
```

Le modèle est complet et n'est pas trop déséquilibré.

```
res=lm(Moy~romantic*Walc,data_anova)
par(mfrow=c(2,2))
plot(res)
```



```
shapiro.test(res$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res$residuals
## W = 0.98902, p-value = 4.858e-07
```

Les données ne sont pas du tout gaussiennes.

```
head(df)
```

```
##  school sex age address famsize Pstatus Medu Fedu  Mjob  Fjob  reason
## 1    GP   F  18      U    GT3      A    4    4  at_home teacher  course
## 2    GP   F  17      U    GT3      T    1    1  at_home  other   course
## 3    GP   F  15      U    LE3      T    1    1  at_home  other   other
```

```
## 4    GP  F  15    U    GT3    T    4    2    health services    home
## 5    GP  F  16    U    GT3    T    3    3    other    other    home
## 6    GP  M  16    U    LE3    T    4    3    services    other reputation
## guardian traveltime studytime failures schoolsup famsup paid activities
## 1    mother    2    2    0    yes    no    no    no
## 2    father    1    2    0    no    yes    no    no
## 3    mother    1    2    3    yes    no    yes    no
## 4    mother    1    3    0    no    yes    yes    yes
## 5    father    1    2    0    no    yes    yes    no
## 6    mother    1    2    0    no    yes    yes    yes
## nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1    yes    yes    no    no    4    3    4    1    1    3
## 2    no    yes    yes    no    5    3    3    1    1    3
## 3    yes    yes    yes    no    4    3    2    2    3    3
## 4    yes    yes    yes    yes    3    2    2    1    1    5
## 5    yes    yes    no    no    4    3    2    1    2    5
## 6    yes    yes    yes    no    5    4    2    1    2    5
## absences G1 G2 G3    Moy    RS
## 1    6 5 6 6 5.666667    exclusion
## 2    4 5 5 6 5.333333    exclusion
## 3    10 7 8 10 8.333333    exclusion
## 4    2 15 14 15 14.666667    admis
## 5    4 6 10 10 8.666667    redoublement
## 6    10 15 15 15 15.000000    admis
```

```
data_anova=df[c(34,18,22)]
head(data_anova)
```

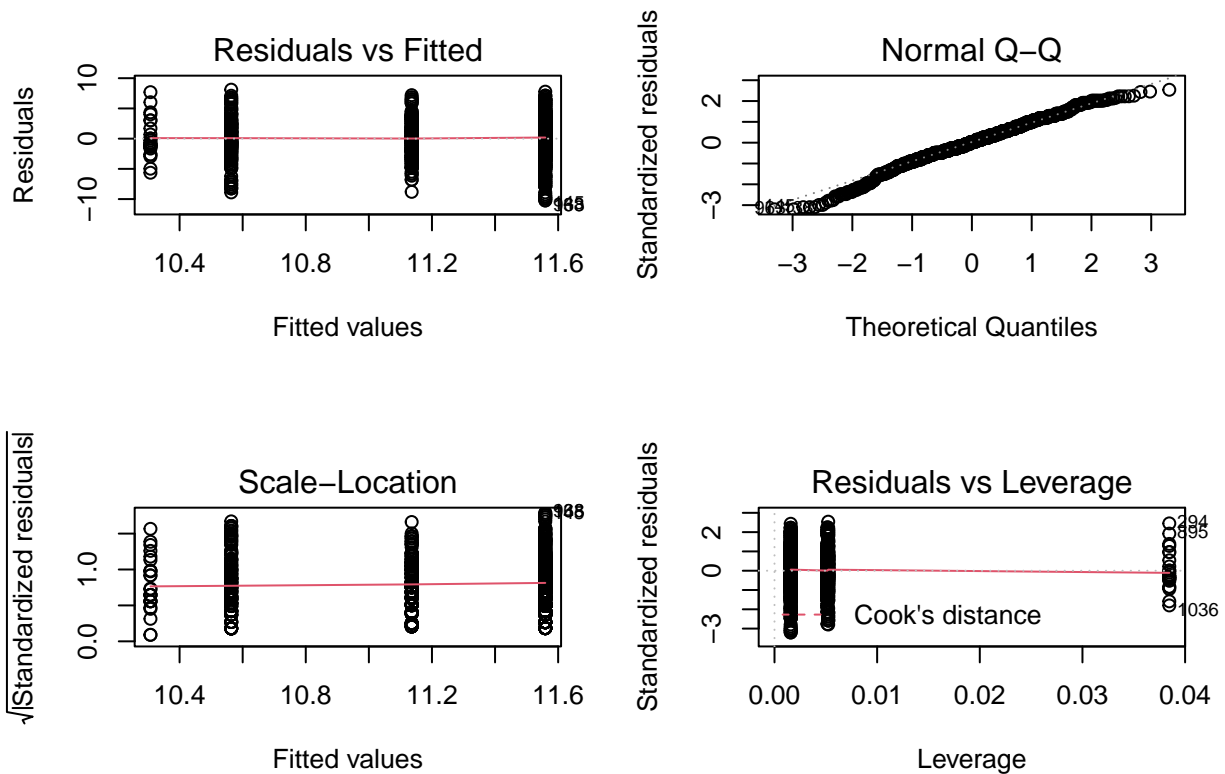
```
##          Moy paid internet
## 1  5.666667    no    no
## 2  5.333333    no    yes
## 3  8.333333    yes    yes
## 4 14.666667    yes    yes
## 5  8.666667    yes    no
## 6 15.000000    yes    yes
```

```
table(data_anova$internet,data_anova$paid)
```

```
##
##          no yes
##    no  191  26
##    yes 633 194
```

Le plan est complet et quasiment équilibré

```
res=lm(Moy~internet*paid,data_anova)
par(mfrow=c(2,2))
plot(res)
```



```
shapiro.test(res$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res$residuals
## W = 0.99028, p-value = 2.162e-06
```

On obtiens encore que les données ne sont pas gaussiennes

##Modèle linéaire Gaussien: Régression mutlipie

```
library(car) #pour utiliser VIF
```

```
## Le chargement a nécessité le package : carData
```

```
##
```

```
## Attachement du package : 'car'
```

```
## L'objet suivant est masqué depuis 'package:dplyr':
```

```
##
```

```
##      recode
```

```
reg=lm(Moy~.,data_quanti)#regression multiple pour utiliser VIF
vif(reg)#test de colinéarité
```

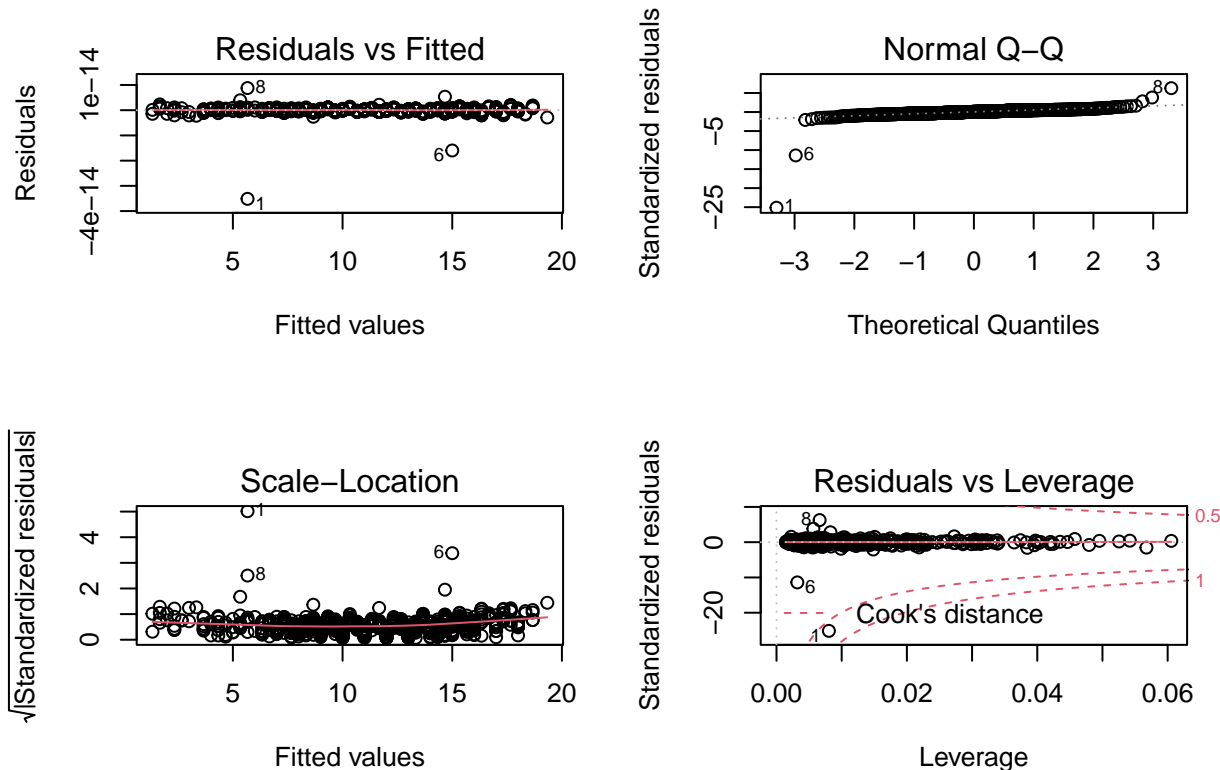
```
## Warning in summary.lm(object, ...): essentially perfect fit: summary may be
## unreliable
```

```
##      age traveltime studytime  failures      G1      G2      G3
##  1.087837  1.027492  1.048104  1.278165  3.943316  7.959427  6.069691
```

Aucune valeur n'est plus grande que 10, la matrice est donc de plein rang. On va maintenant vérifier si les résidus sont iid, gaussiens centrée et réduits



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



On voit qu'il n'y a pas de forme de trompette sur le graphe des résidus donc l'hypothèse d'homoscédasticité est vérifiée. Néanmoins il semble y avoir plusieurs points avec des résidus trop grands.

```
abs(rstudent(reg))[abs(rstudent(reg))>2]
```

```
##          1          2          4          6          8         48
## 40.270194  2.823092  3.837576 12.191489  6.387979  2.082211
```

En effet, on voit qu'il y en a huit. Il faudrait enlever le point le plus éloigné. Néanmoins, on voit en regardant le qqplot nos variables n'ont aucune chance d'être gaussiennes. En effet, avec la p-valeur du test de shapiro qui est très petite devant 5%, on rejette  $H_0$ , les données ne sont donc pas gaussiennes. Le modèle n'est donc pas adapté.

```
shapiro.test(reg$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg$residuals
## W = 0.35748, p-value < 2.2e-16
```