# Devoir 5 - Solutions

Patrick Boily

2023-04-10

### Preliminaires

```
library(tidyverse)  # pour avoir acces a select() et |>
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## Q41

Considérons l'ensemble de données `Autos.xlsx` se retrouvant sur Brightspace. Le prédicteur est `Type` ($X$, type de véhicule); la réponse est `CC.q` ($Y$, consommation de carburant quotidienne moyenne, en L). En utilisant un encodage de variable nominale, déterminez un modèle de régression de $Y$ en fonction de $X$. Est-ce un bon modèle? Justifiez votre réponse.

**Solution:** on commence par aller chercher l'ensemble de donnees en question.

```r
Autos <- readxl::read_excel("Autos.xlsx") |> select(Type,CC.q)
str(Autos)
```

```
## tibble [996 x 2] (S3: tbl_df/tbl/data.frame)
##  $ Type: chr [1:996] "PUPC" "PUPC" "PUPC" "PUPC" ...
##  $ CC.q: num [1:996] 49 33 44 22 38 31 28 19 31 19 ...
```

```r
x = factor(Autos$Type)
y = Autos$CC.q
```

Il y a 4 niveaux de variable categorielle pour `Type`:

```r
levels(x)
```

```
## [1] "MVAN" "PUPC" "VPAS" "VUS"
```

Ensuite, on effectue l'ajustement:

```r
mod.1 = lm(y ~ x)
summary(mod.1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.117 -4.504 -1.555  3.445 40.883
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5039     0.5272  12.338  < 2e-16 ***
## xPUPC         1.6130     0.6959   2.318  0.02066 *
## xVPAS        -1.9493     0.5911  -3.298  0.00101 **
## xVUS         -0.1216     0.6715  -0.181  0.85636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.941 on 992 degrees of freedom
## Multiple R-squared:  0.04931,    Adjusted R-squared:  0.04644
## F-statistic: 17.15 on 3 and 992 DF,  p-value: 7.272e-11
```

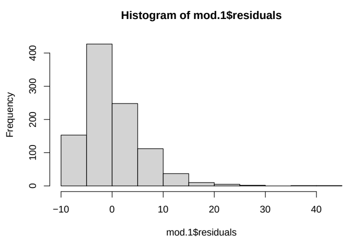
Le modele est:

$$\hat{Y} = 6.5039 + 1.6130 \cdot \mathcal{I}(x = \texttt{PUPC}) - 1.9493 \cdot \mathcal{I}(x = \texttt{VPAS}) - 0.1216 \cdot \mathcal{I}(x = \texttt{VUS}).$$

Les parametres $b_0$, $b_1$, et $b_2$ sont significatifs a un niveau de confiance $\alpha = 0.05$ (la regression elle-meme est significative, avec une valeur$-p$ de $P(F(3, 992) > 17.15) = 7.272e - 11)$, mais le coefficient de determination est tres faible, avec une valeur de $R^2 = 0.049$.

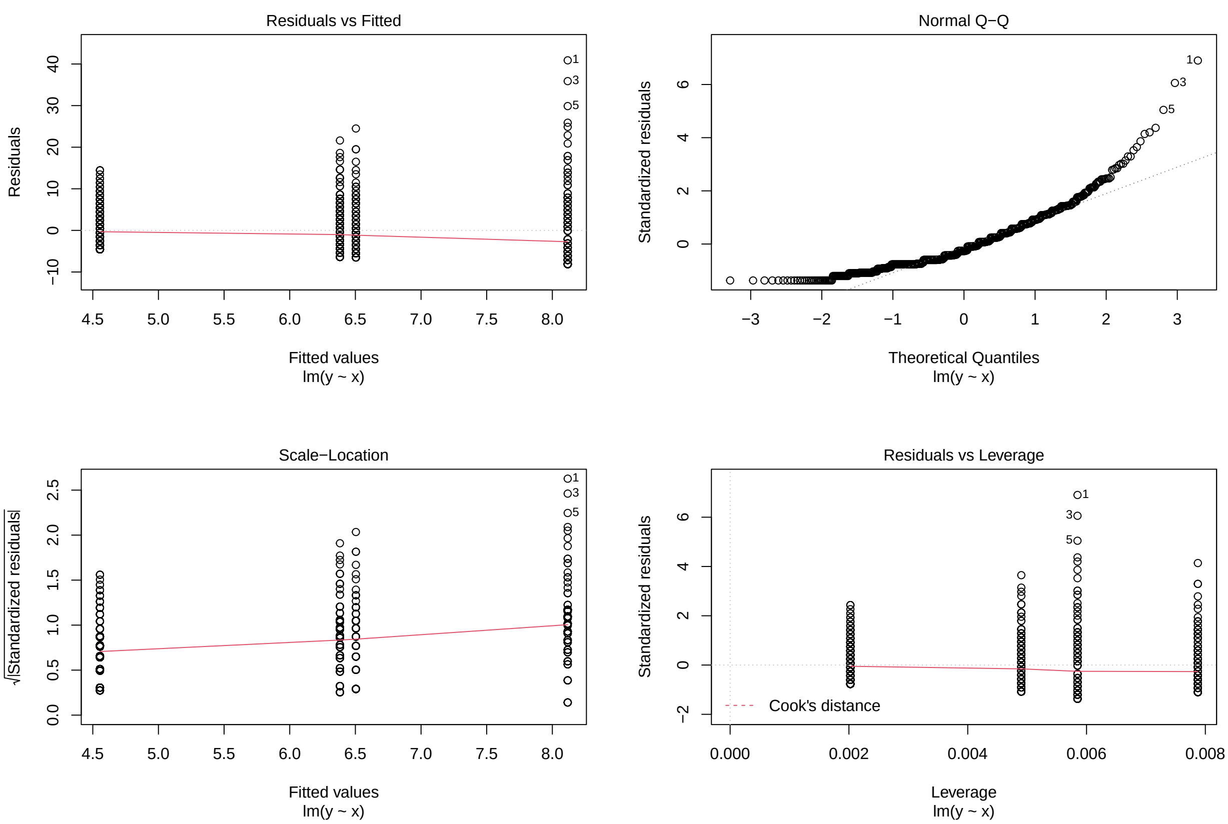
Est-ce un bon modele? On peut commencer par verifier si les residus sont compatibles avec la structure d'erreur requise.

```
hist(mod.1$residuals)
```




C'est plus ou moins symetrique, mais il y a des residus qui sont tres eleves, voir meme aberrants. Qu'en est-il des autres diagnostiques? ... vraiment pas tres fort. Alors non, ce n'est pas un bon modele.

```
plot(mod.1)
```



## Q42

Utilisez l'ensemble de données de l'exemple dans la section 4.5.

a. Déterminez la solution du problème des moindres carrés pondérés avec $w_i = x_i^2$, $i = 1, \ldots, n$. Tracez les résultats.

b. Déterminez la solution du problème des moindres carrés pondérés avec la procédure décrite en p.37. Tracez les résultats.

c. Laquelle des deux options donne le meilleur ajustement? Justifiez votre réponse.

**Solution:** on commence par mettre les donnes dans des vecteurs.

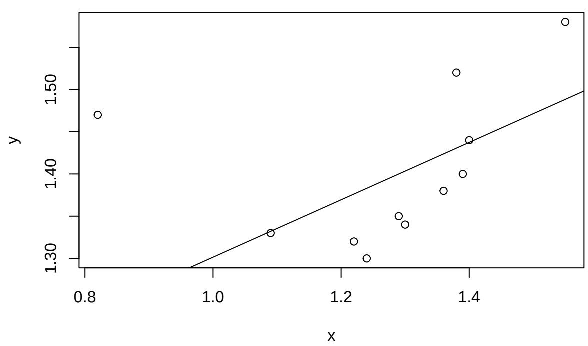
```
x = c(0.82,1.09,1.22,1.24,1.29,1.30,1.36,1.38,1.39,1.40,1.55)
y = c(1.47,1.33,1.32,1.30,1.35,1.34,1.38,1.52,1.40,1.44,1.58)
w.2 = x^2
```

a. La solution dans ce cas est:

```
mod.2 = lm(y ~ x, weights = w.2)
summary(mod.2)
```

```
##
## Call:
## lm(formula = y ~ x, weights = w.2)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10289 -0.06650 -0.04718  0.06359  0.18837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9617     0.2208   4.356  0.00183 **
## x             0.3398     0.1657   2.050  0.07057 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1069 on 9 degrees of freedom
## Multiple R-squared:  0.3184, Adjusted R-squared:  0.2427
## F-statistic: 4.204 on 1 and 9 DF,  p-value: 0.07057
```

```
plot(x,y)
abline(mod.2)
```




b. on utilise la formulation avec les $|e_i|$:

```
mod = lm(y ~ x)
w.3 = 1/lm(abs(mod$residuals) ~ x)$fitted.values^2
mod.3 <- lm(y ~ x, weights=w.3)
(MSE.w = sum(mod.3$residuals^2)/(length(x)-2))
```

```
## [1] 0.008986807
```

```
w.4 = 1/lm(abs(mod.3$residuals) ~ x)$fitted.values^2
mod.4 <- lm(y ~ x, weights=w.4)
(MSE.w = sum(mod.4$residuals^2)/(length(x)-2))
```

```
## [1] 0.01546611
```

```
w.5 = 1/lm(abs(mod.4$residuals) ~ x)$fitted.values^2
mod.5 <- lm(y ~ x, weights=w.5)
(MSE.w = sum(mod.5$residuals^2)/(length(x)-2))
```

```
## [1] 0.02532088
```

```
w.6 = 1/lm(abs(mod.5$residuals) ~ x)$fitted.values^2
mod.6 <- lm(y ~ x, weights=w.6)
(MSE.w = sum(mod.6$residuals^2)/(length(x)-2))
```

```
## [1] 0.01835132
```

```
w.7 = 1/lm(abs(mod.6$residuals) ~ x)$fitted.values^2
mod.7 <- lm(y ~ x, weights=w.7)
(MSE.w = sum(mod.7$residuals^2)/(length(x)-2))
```

```
## [1] 0.02168024
```

```r
w.8 = 1/lm(abs(mod.7$residuals) ~ x)$fitted.values^2
mod.8 <- lm(y ~ x, weights=w.8)
(MSE.w = sum(mod.8$residuals^2)/(length(x)-2))
```

```
## [1] 0.01991129
```

```r
w.9 = 1/lm(abs(mod.8$residuals) ~ x)$fitted.values^2
mod.9 <- lm(y ~ x, weights=w.9)
(MSE.w = sum(mod.9$residuals^2)/(length(x)-2))
```

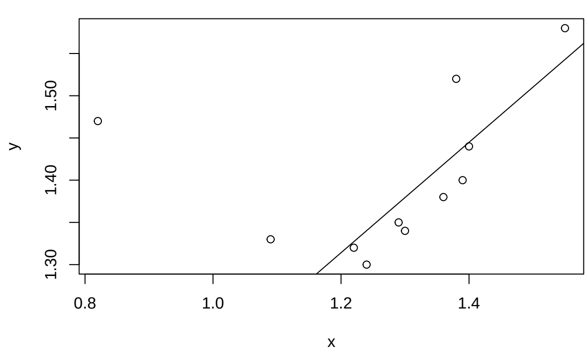
```
## [1] 0.02079419
```

Il n'y a plus bien d'amelioration de $\mathrm{MSE}_w$ en faisant des iterations supplementaires, alors on est aussi bien s'arreter ici.

```r
summary(mod.9)
```

```
##
## Call:
## lm(formula = y ~ x, weights = w.9)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8080 -0.5421 -0.2906  0.6374  3.4293
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5295     0.3869   1.369   0.2043
## x             0.6538     0.2791   2.343   0.0438 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.49 on 9 degrees of freedom
## Multiple R-squared:  0.3788, Adjusted R-squared:  0.3098
## F-statistic: 5.489 on 1 and 9 DF,  p-value: 0.04382
```

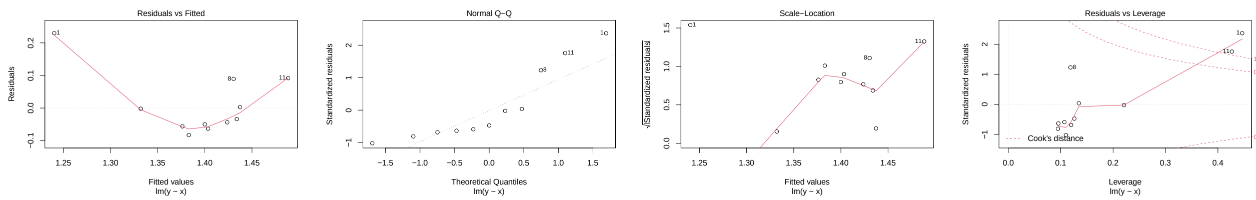
```r
plot(x,y)
abline(mod.9)
```



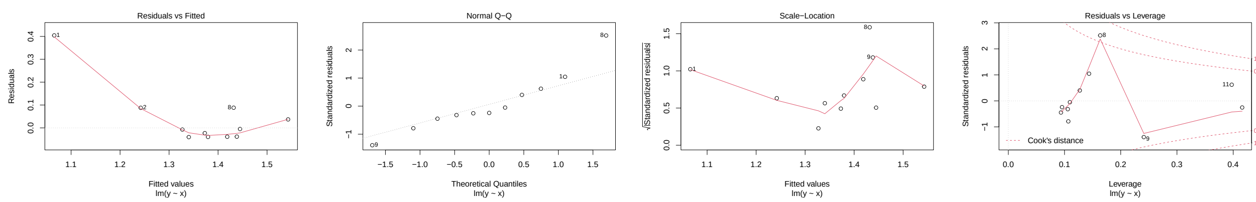
c. Au visuel, on constate que l'approche WLS donne un meilleur ajustement que l'approche avec les poids arbitraires $x^2$, ce qui n'est pas bien surprenant puisque l'on optimise la ponderation lors de l'approche WLS.

Les graphiques diagnostiques penchent aussi de ce cote, quoiqu'il n'y ait pas vraiment assez d'observations pour en etre absolument certain.

```r
plot(mod.2)
```




```r
plot(mod.9)
```



## Q43

Considérons l'ensemble de données `Autos.xlsx` se retrouvant sur Brightspace. Les prédicteurs sont `VKM.q` ($X_1$, distance quotidienne moyenne, en km), `Age` ($X_2$, age du véhicule en années), et `Rural` ($X_3 = 0$ pour un véhicule urbain, $X_3 = 1$ pour un véhicule rural); la réponse est toujours `CC.q` ($Y$, consommation de carburant quotidienne moyenne, en L). Utilisez l'approche du meilleur sous-ensemble avec le critère $C_p$ de Mallow afin de sélectionner le meilleur modèle.

**Solution:** on commence par aller chercher l'ensemble de donnees en question.

```
Autos <- readxl::read_excel("Autos.xlsx") |> select(VKM.q,Age,Rural,CC.q)
str(Autos)
```

```
## tibble [996 x 4] (S3: tbl_df/tbl/data.frame)
##  $ VKM.q: num [1:996] 330 264 251 235 230 230 215 208 203 196 ...
##  $ Age  : num [1:996] 0 1 10 1 3 5 9 6 3 9 ...
##  $ Rural: num [1:996] 0 0 0 1 1 1 0 0 0 0 ...
##  $ CC.q : num [1:996] 49 33 44 22 38 31 28 19 31 19 ...
```

```
x1 = Autos$VKM.q
x2 = Autos$Age
x3 = Autos$Rural
y = Autos$CC.q
n = nrow(Autos)
```

Il n'y a pas beaucoup de modeles possible (nous en profitons pour calculer des quantites qui seront utiles lors des prochaines questions...):

- pour $p = 0$, nous avons: `y ~ .`

```
p=0
mod.0 = lm(y ~ 1, data=Autos)
SSE.0 = sum(mod.0$residuals^2)
C.0 = 1/n*SSE.0 + 2*(p+2)/n*summary(mod.0)$sigma^2
R.a.2.0 = summary(mod.0)$adj.r.squared
```

- pour $p = 1$, nous avons:
  - $(1, 0, 0)$: `y ~ x1`

```
p=1
mod.1.0.0 = lm(y ~ x1, data=Autos)
SSE.1.0.0 = sum(mod.1.0.0$residuals^2)
C.1.0.0 = 1/n*SSE.1.0.0 + 2*(p+2)/n*summary(mod.1.0.0)$sigma^2
R.a.2.1.0.0 = summary(mod.1.0.0)$adj.r.squared
```

- $(0, 1, 0)$: `y ~ x2`

```
p=1
mod.0.1.0 = lm(y ~ x2, data=Autos)
SSE.0.1.0 = sum(mod.0.1.0$residuals^2)
C.0.1.0 = 1/n*SSE.0.1.0 + 2*(p+2)/n*summary(mod.0.1.0)$sigma^2
R.a.2.0.1.0 = summary(mod.0.1.0)$adj.r.squared
```

- $(0,0,1)$: `y ~ x3`

```
p=1
mod.0.0.1 = lm(y ~ x3, data=Autos)
SSE.0.0.1 = sum(mod.0.0.1$residuals^2)
C.0.0.1 = 1/n*SSE.0.0.1 + 2*(p+2)/n*summary(mod.0.0.1)$sigma^2
R.a.2.0.0.1 = summary(mod.0.0.1)$adj.r.squared
```

- pour $p = 2$, nous avons:
  - $(1,1,0)$: `y ~ x1 + x2`

```
p=2
mod.1.1.0 = lm(y ~ x1 + x2, data=Autos)
SSE.1.1.0 = sum(mod.1.1.0$residuals^2)
C.1.1.0 = 1/n*SSE.1.1.0 + 2*(p+2)/n*summary(mod.1.1.0)$sigma^2
R.a.2.1.1.0 = summary(mod.1.1.0)$adj.r.squared
```

- $(1,0,1)$: `y ~ x1 + x3`

```
p=2
mod.1.0.1 = lm(y ~ x1 + x3, data=Autos)
SSE.1.0.1 = sum(mod.1.0.1$residuals^2)
C.1.0.1 = 1/n*SSE.1.0.1 + 2*(p+2)/n*summary(mod.1.0.1)$sigma^2
R.a.2.1.0.1 = summary(mod.1.0.1)$adj.r.squared
```

- $(0,1,1)$: `y ~ x2 + x3`

```
p=2
mod.0.1.1 = lm(y ~ x2 + x3, data=Autos)
SSE.0.1.1 = sum(mod.0.1.1$residuals^2)
C.0.1.1 = 1/n*SSE.0.0.1 + 2*(p+2)/n*summary(mod.0.1.1)$sigma^2
R.a.2.0.1.1 = summary(mod.0.1.1)$adj.r.squared
```

- pour $p = 3$, nous avons: `y ~ x1 + x2 + x3`

```
p=3
mod.3 = lm(y ~ x2 + x3, data=Autos)
SSE.3 = sum(mod.3$residuals^2)
C.3 = 1/n*SSE.3 + 2*(p+2)/n*summary(mod.3)$sigma^2
R.a.2.3 = summary(mod.3)$adj.r.squared
```

Voici les resultat dans un tableau:

```
resultats <- data.frame(rbind(c(0,0,0,0,SSE.0,C.0,R.a.2.0),
                 c(1,1,0,0,SSE.1.0.0,C.1.0.0,R.a.2.1.0.0),
                 c(1,0,1,0,SSE.0.1.0,C.0.1.0,R.a.2.0.1.0),
                 c(1,0,0,1,SSE.0.0.1,C.0.0.1,R.a.2.0.0.1),
                 c(2,1,1,0,SSE.1.1.0,C.1.1.0,R.a.2.1.1.0),
                 c(2,1,0,1,SSE.1.0.1,C.1.0.1,R.a.2.1.0.1),
```

```r
                 c(2,0,1,1,SSE.0.1.1,C.0.1.1,R.a.2.0.1.1),
                 c(3,1,1,1,SSE.3,C.3,R.a.2.3)))
colnames(resultats) <- c("k","X1", "X2", "X3", "SSE", "C.p", "R.a.2")
rownames(resultats) <- c()
resultats
```

```
##   k X1 X2 X3      SSE      C.p      R.a.2
## 1 0  0  0  0 36827.72 37.12427 0.00000000
## 2 1  1  0  0 10416.13 10.52109 0.71688154
## 3 1  0  1  0 36061.98 36.42536 0.01980750
## 4 1  0  0  1 36356.13 36.72247 0.01181219
## 5 2  1  1  0 10388.64 10.51440 0.71734427
## 6 2  1  0  1 10324.40 10.44938 0.71909208
## 7 2  0  1  1 35556.00 36.78974 0.03258713
## 8 3  1  1  1 35556.00 36.05830 0.03258713
```

Pour chaque nombre de predicteurs $k$, on choisit $\mathcal{M}_k$ ayant la plus petite valeur de SSE:

```r
resultats.2 = resultats[ resultats$SSE == ave(resultats$SSE, resultats$k, FUN=min), ]
rownames(resultats.2) <- c()
resultats.2
```

```
##   k X1 X2 X3      SSE      C.p      R.a.2
## 1 0  0  0  0 36827.72 37.12427 0.00000000
## 2 1  1  0  0 10416.13 10.52109 0.71688154
## 3 2  1  0  1 10324.40 10.44938 0.71909208
## 4 3  1  1  1 35556.00 36.05830 0.03258713
```

Finalement, on choisit le modele qui a la plus petite valeur de $C_p$.

```r
resultats.3 <- resultats.2[ resultats.2$C.p == min(resultats.2$C.p), ]
rownames(resultats.3) <- c()
resultats.3
```

```
##   k X1 X2 X3     SSE      C.p     R.a.2
## 1 2  1  0  1 10324.4 10.44938 0.7190921
```

Le meilleur modele est ainsi:

```r
mod.1.0.1
```

```
##
## Call:
## lm(formula = y ~ x1 + x3, data = Autos)
##
## Coefficients:
## (Intercept)           x1           x3
##     -0.2033       0.1216       0.9452
```

## Q44

Répétez la question précédente, mais avec le coefficient de détermination ajusté $R_a^2$.

**Solution:** on s'y prend de la meme maniere pour obtenir la table `resultats.2`, mais on utilise $R_a^2$ pour obtenir:

```
resultats.3 <- resultats.2[ resultats.2$R.a.2 == max(resultats.2$R.a.2), ]
rownames(resultats.3) <- c()
resultats.3
```

```
##   k X1 X2 X3     SSE      C.p     R.a.2
## 1 2  1  0  1 10324.4 10.44938 0.7190921
```

C'est le meme modele!

```
summary(mod.1.0.1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x3, data = Autos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3181 -1.8091 -0.4848  0.6769 23.6501
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.20328    0.15810  -1.286  0.19881
## x1           0.12160    0.00243  50.037  < 2e-16 ***
## x3           0.94516    0.31822   2.970  0.00305 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.224 on 993 degrees of freedom
## Multiple R-squared:  0.7197, Adjusted R-squared:  0.7191
## F-statistic:  1275 on 2 and 993 DF,  p-value: < 2.2e-16
```

### Q45

Répétez la question précédente, mais avec la méthode de sélection par étapes rétrograde et avec le critère $C_p$ de Mallow.

**Solution:** on commence avec le modele complete $\mathcal{M}_3$. On choisit ensuite le modele a 2 variables qui a la plus petite SSE: c'est le modele $\mathcal{M}_2 \equiv (1, 0, 1)$.

Ensuite, on calcule la SSE pour les modeles qui ont un predicteur en moins a partir de $\mathcal{M}_2$, c'est-a-dire $(1, 0, 0)$ et $(0, 0, 1)$ et on obtient $\mathcal{M}_1 \equiv (1, 0, 0)$.

Finalement, on considere le modele nul $\mathcal{M}_0$.

Le tableau des modeles est ainsi:

```
resultats.2 = resultats[c(1,2,6,8),]
rownames(resultats.2) <- c()
resultats.2
```

```
##   k X1 X2 X3      SSE      C.p      R.a.2
## 1 0  0  0  0 36827.72 37.12427 0.00000000
## 2 1  1  0  0 10416.13 10.52109 0.71688154
## 3 2  1  0  1 10324.40 10.44938 0.71909208
## 4 3  1  1  1 35556.00 36.05830 0.03258713
```

Vu que c'est le meme tableau qu'aux questions precedentes, c'est encore une fois le modele $(1, 0, 1)$ qui l'emporte.

## Q46

Répétez la question précédente, mais avec la méthode de sélection par étapes rétrograde et avec le coefficient de détermination ajusté $R_a^2$.

**Solution:** voir la reponse de la question precedente.

## Q47

Répétez la question précédente, mais avec la méthode de sélection par étapes par l'avant et avec le critère $C_p$ de Mallow.

**Solution:** on obtient le meme tableau qu'aux questions precedentes, alors cela sera le modele $(1, 0, 1)$ qui est retenu.

## Q48

Répétez la question précédente, mais avec la méthode de sélection par étapes par l'avant et avec le coefficient de détermination ajusté $R_a^2$.

**Solution:** on obtient le meme tableau qu'aux questions precedentes, alors cela sera le modele $(1, 0, 1)$ qui est retenu. Il est important de noter que cela ne sera pas toujours le cas, en general, mais que c'est bien ce que nous obenons ici.

## Q49

Considérons l'ensemble de données `Autos.xlsx` se retrouvant sur Brightspace. Nous ne nous intéressons qu'aux véhicules de type VPAS. Les prédicteurs sont `VKM.q` ($X_1$, distance quotidienne moyenne, en km) et `Age` ($X_2$, age du véhicule en années); la réponse est toujours `CC.q` ($Y$, consommation de carburant quotidienne moyenne, en L). Déterminez les valeurs aberrantes en $X$ de l'ensemble de données.

**Solution:** on va chercher la matrice $\mathbf{H}$.

```
Autos <- readxl::read_excel("Autos.xlsx") |>
  filter(Type == "VPAS") |> select(VKM.q,Age,CC.q)
x1 = Autos$VKM.q
x2 = Autos$Age
y = Autos$CC.q

fit <- lm(y ~ x1 + x2, data=Autos)
X = model.matrix(fit)
H = X %*% solve(t(X) %*% X) %*% t(X)
```

Les leviers se retrouvent sur la diagonale de $\mathbf{H}$.

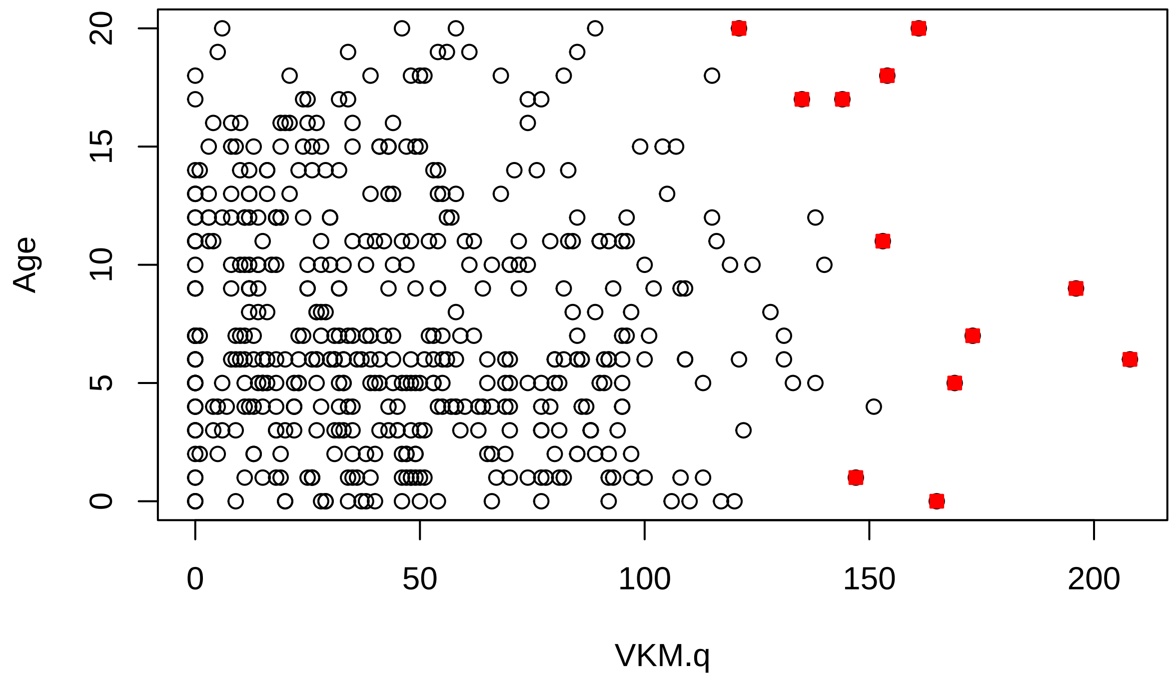
```
leviers = diag(H)
```

Puisque $n$ est relativement eleve, les valeurs aberrantes en $X$ sont les observations pour lesquelles $h_{ii} > 3\overline{h}$.

```
3*mean(leviers)
```

```
## [1] 0.01821862
```

```
indices.X = which(leviers > 3*mean(leviers))
plot(x1,x2, xlab = "VKM.q", ylab = "Age")
points(x1[indices.X],x2[indices.X], col="red", pch=22, bg="red")
```




```r
unname(indices.X)
```

```
##  [1]  1  2  3  4  5  6  7  8 10 11 15 23
```

Les valeurs aberrantes en $X$ sont traces plus bas:

```r
Autos[indices.X,c("VKM.q","Age")]
```

```
## # A tibble: 12 x 2
##    VKM.q   Age
##    <dbl> <dbl>
##  1   208     6
##  2   196     9
##  3   173     7
##  4   169     5
##  5   165     0
##  6   161    20
##  7   154    18
##  8   153    11
##  9   147     1
## 10   144    17
## 11   135    17
## 12   121    20
```

## Q50

Construisez le modèle d'ajustement linéaire par les moindres carrés $\hat{Y} = b_0 + b_1 X_1 + b_2 X_2$. Identifiez les valeurs aberrantes en $Y$ de l'ensemble de données.

**Solution:** une facon de s'y prendre est de calculer les residus studentises internes,

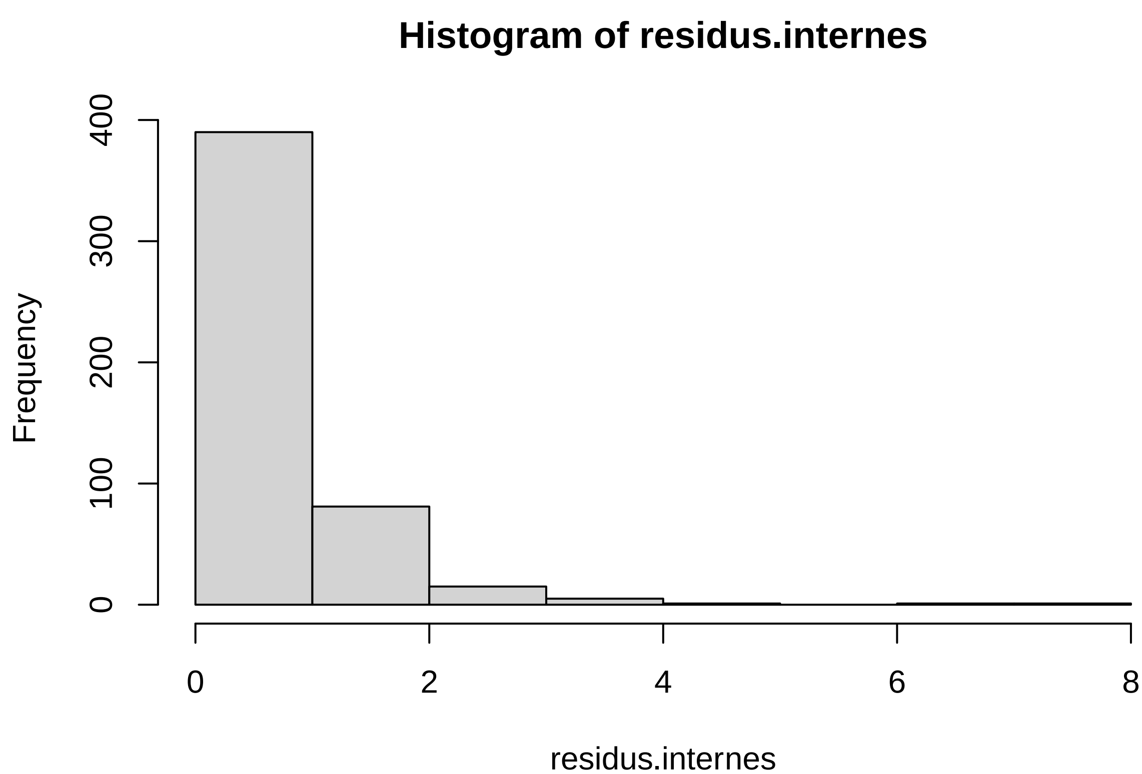
$$|r_i| = \left|\frac{e_i}{s\{e_i\}}\right| = \left|\frac{e_i}{\sqrt{\text{MSE}}\sqrt{1-h_{ii}}}\right|.$$

Nous avons deja calcule le modele a la question precedente (`fit`).

```
e = fit$residuals
MSE = summary(fit)$sigma^2
residus.internes = abs(e/sqrt(MSE*(1-leviers)))
```

L'histogramme des residus studentises internes est trace ci-bas.

```
hist(residus.internes)
```




Il y en a quelques uns qui sont plus grand que 3.

Allons les chercher:

```
indices.Y = which(residus.internes >= 3)
Autos[indices.Y,]
```

```
## # A tibble: 8 x 3
##   VKM.q   Age  CC.q
##   <dbl> <dbl> <dbl>
## 1   131     7    19
## 2    95     5    17
## 3    68    18    15
## 4    50     1    11
## 5    46    11    11
## 6    17    10    16
## 7    14    10    16
## 8    13     6     8
```