Devoir 3 - Solutions

Patrick Boily

2023-02-25

Préliminaires 1

Nous importons l'ensemble `Autos.xlsx` se retrouvant sur Brightspace, avec prédicteur `VKM.q` (X, distance quotidienne moyenne, en km) et réponse `CC.q` (Y, consommation de carburant quotidienne moyenne, en L).

```
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()
```

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

```
## tibble [996 x 2] (S3: tbl_df/tbl/data.frame)
##  $ VKM.q: num [1:996] 330 264 251 235 230 230 215 208 203 196 ...
##  $ CC.q : num [1:996] 49 33 44 22 38 31 28 19 31 19 ...
```

```
x = Autos$VKM.q
y = Autos$CC.q
```

Q21

Exprimez le modèle de régression linéaire $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ à l'aide de la notation matricielle. Utilisez `R` afin de déterminer directement la solution des moindres carrés (sans passer par `lm()`, ni par les sommes $\sum X_i$, $\sum Y_i$, $\sum X_i^2$, $\sum X_i Y_i$, $\sum Y_i^2$).

Solution: on écrit

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \quad \text{et} \quad \mathbf{X} = \begin{pmatrix} 1 & X_{1,1} \\ \vdots & \vdots \\ 1 & X_{n,1} \end{pmatrix}.$$

Ainsi,

$$\mathbf{b} = \hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} :$$

```
intercept = x^0
X = cbind(intercept, x)
(b = solve(t(X) %*% X) %*% (t(X) %*% y))
```

```
##                 [,1]
## intercept -0.1183883
## x          0.1221413
```

Ainsi, $\hat{Y} = -0.1183883 + 0.1221413X$.

Préliminaires 2

Nous importons l'ensemble `Autos.xlsx` se retrouvant sur Brightspace. Nous ne nous intéressons qu'aux véhicules de type VPAS, avec prédicteurs `VKM.q` (X_1, distance quotidienne moyenne, en km) et `Age` (X_2, age du véhicule, en années), et réponse `CC.q` (Y, consommation de carburant quotidienne moyenne, en L).

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

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

```
## tibble [494 x 3] (S3: tbl_df/tbl/data.frame)
##  $ VKM.q: num [1:494] 208 196 173 169 165 161 154 153 151 147 ...
##  $ Age  : num [1:494] 6 9 7 5 0 20 18 11 4 1 ...
##  $ CC.q : num [1:494] 19 19 14 15 18 14 16 13 14 13 ...
```

```r
x1 = Autos$VKM.q
x2 = Autos$Age
y = Autos$CC.q
```

Q22

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).

Utilisez `R` afin de:

a) déterminer la matrice de conception $\mathbf{X}$ du modèle de RLG;

b) calculer les valeurs ajustées de la réponse $\mathbf{Y}$ si $\boldsymbol{\beta} = (1, 5, 1)$;

c) calculer la somme des carrés des résidus lorsque $\boldsymbol{\beta} = (1, 5, 1)$.

Solution:

a) La matrice de conception est tout simplement

$$\mathbf{X} = [\mathbf{1} \mid X_1 \mid X_2] :$$

```
intercept = x1^0
X = cbind(intercept, x1, x2)
```

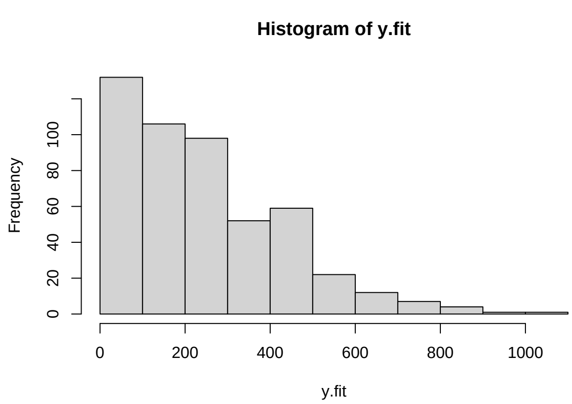
On pourrait imprimer le résultat, mais il est important de constater que cela ne serait pas bien utile…

b) Les valeurs ajustées sont tout simplement $\hat{\mathbf{Y}} = \mathbf{X}\boldsymbol{\beta}$. Dans notre cas, nous obtenons:

```
beta = c(1,5,1)
y.fit = X %*% beta
```

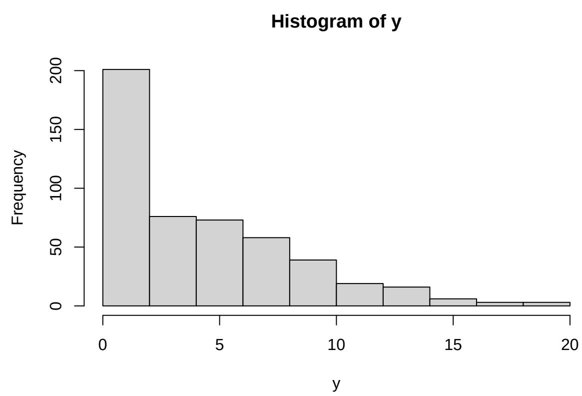
Encore une fois, il serait préférable de ne pas imprimer les résultats… mais on peut toutefois se donner une idée des résultats:

```
hist(y.fit)
```



En quoi cela se compare-t-il aux réponses réelles?

```
hist(y)
```



Oh boy, ce ne sont pas de bien bonnes valeurs ajustées... pourquoi est-ce le cas, selon vous?

c) La somme des carrés des résidus est donnée par

$$\text{SSE} = \mathbf{Y}^\top (\mathbf{I}_n - \mathbf{H})\,\mathbf{Y} = \mathbf{Y}^\top \left(\mathbf{I}_n - \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\right)\mathbf{Y}.$$

Ainsi:

```
I.n = diag(1, nrow=length(y), ncol=length(y))
H = X %*% solve(t(X) %*% X) %*% t(X)
(SSE = as.numeric(t(y) %*% (I.n-H) %*% y))
```

```
## [1] 2120.459
```

... mais ceci n'est pas vraiment compatible avec les valeurs de `y.fit` et `y` observées en b) (pourquoi?)

C'est que la formule $\text{SSE} = \mathbf{Y}^\top (\mathbf{I}_n - \mathbf{H})\,\mathbf{Y}$ est valide **en supposant que l'ajustement linéaire utilisé est celui donné par les moindres carrés**, ce qui n'est pas nécessairement le cas ici (nous n'avons pas encore calculé l'estimateur en question).

Il faut plutôt calculer

```
sum((y.fit-y)^2)
```

```
## [1] 46018592
```

Voilà qui est plus raisonnable!

Q23

Déterminez directement (à l'aide de manipulations matricielles dans `R`) l'estimateur des moindres carrés **b** du problème RLG. Exprimez la fonction de régression estimée de la réponse Y. Calculez la somme des carrés des résidus dans le cas $\boldsymbol{\beta} = \mathbf{b}$. Cette valeur est-elle compatible avec le résultat obtenu à la partie c) de la question précédente?

Solution: nous avons

```
intercept = x1^0
X = cbind(intercept, x1, x2)
(b = solve(t(X) %*% X) %*% (t(X) %*% y))
```

```
##                   [,1]
## intercept -0.014050253
## x1         0.095157626
## x2         0.007384133
```

Nous avons déjà calculé la somme des carrés de résidus à la question précédente:

```
I.n = diag(1, nrow=length(y), ncol=length(y))
H = X %*% solve(t(X) %*% X) %*% t(X)
(SSE = as.numeric(t(y) %*% (I.n-H) %*% y))
```

```
## [1] 2120.459
```

La somme de carrés des résidus avec $\boldsymbol{\beta}_{\text{OLS}} = (-0.014050253, 0.095157626, 0.007384133)$ est nettement inférieure à celle obtenue lorsque nous utilisons $\boldsymbol{\beta} = (1, 5, 1)$.

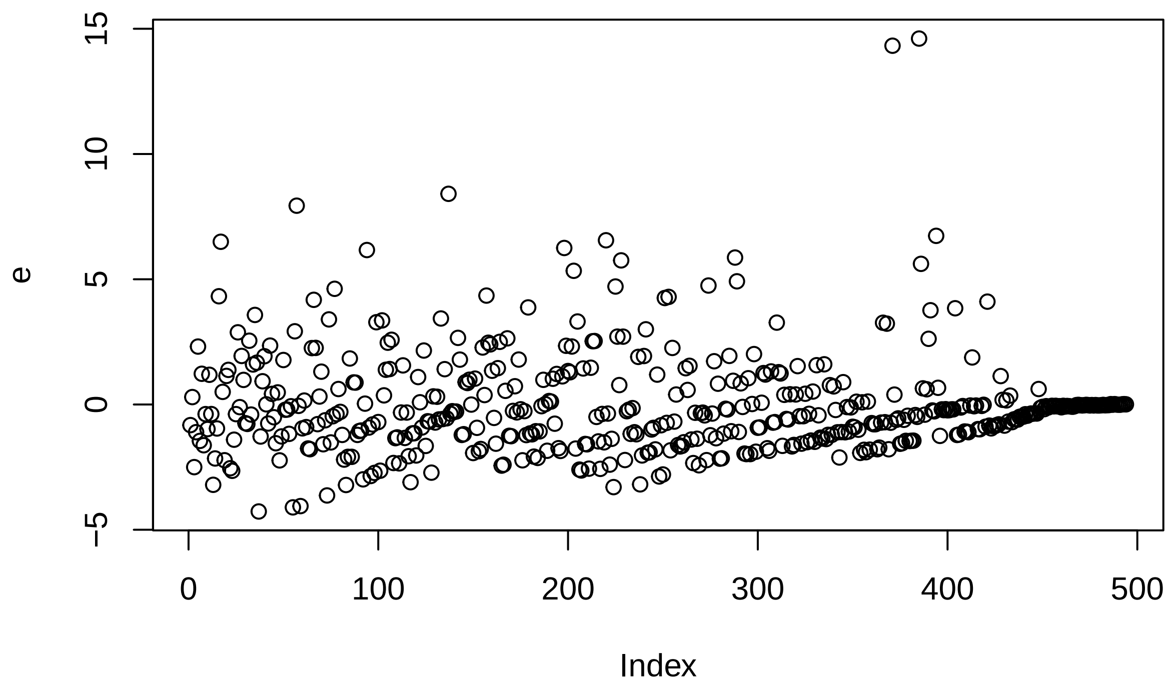
Q24

En ne vous servant que de manipulations matricielles dans `R`, déterminez le vecteur des résidus du problème RLG, ainsi que SST, SSE, et SSR. Vérifiez que SST = SSR + SSE. Quelle est l'erreur quadratique moyenne MSE du modèle RLG?

Solution: le vecteur des résidus est

$$\mathbf{e} = (\mathbf{I}_n - \mathbf{H})\mathbf{Y} :$$

```
e=(I.n-H) %*% y
plot(e)
```



We have

$$\text{SST} = \mathbf{Y}^\top \left(\mathbf{I}_n - \frac{1}{n}\mathbf{J}_n\right)\mathbf{Y}, \quad \text{SSE} = \mathbf{Y}^\top (\mathbf{I}_n - \mathbf{H})\mathbf{Y}, \quad \text{SSR} = \mathbf{Y}^\top \left(\mathbf{H} - \frac{1}{n}\mathbf{J}_n\right)\mathbf{Y} :$$

```
I.n = diag(1, nrow=length(y), ncol=length(y))
H = X %*% solve(t(X) %*% X) %*% t(X)
J.n = matrix(1, nrow = length(y), ncol = length(y))
(SST = as.numeric(t(y) %*% (I.n-J.n/length(y)) %*% y))
```

```
## [1] 8632.024
```

```
(SSE = as.numeric(t(y) %*% (I.n-H) %*% y))
```

```
## [1] 2120.459
```

```
(SSR = as.numeric(t(y) %*% (H-J.n/length(y)) %*% y))
```

```
## [1] 6511.566
```

Nous voyons que $\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE}$:

```
SST-SSE-SSR
```

```
## [1] 9.913492e-11
```

Finalement, puisque $p = 3$, l'erreur quadratique moyenne est:

```
p=3
(MSE=SSE/(length(y)-p))
```

```
## [1] 4.318653
```

Q25

En supposant que le modèle RLG soit valide, testez si la régression est significative à l'aide du test F global – utilisez `R` comme vous l'entendrez, mais utilisez-le!

Solution: nous avons trouvé les estimateurs un peu plus tôt, mais nous allons recommencer en utilisant la fonction `lm()`.

```
mod <- lm(y ~ x1 + x2)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2704 -1.2115 -0.3180  0.6609 14.6080
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.014050   0.207183  -0.068    0.946
## x1           0.095158   0.002452  38.815   <2e-16 ***
## x2           0.007384   0.018365   0.402    0.688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.078 on 491 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7533
## F-statistic: 753.9 on 2 and 491 DF,  p-value: < 2.2e-16
```

Le test F global oppose $H_0 : \beta_0 = \beta_1 = \beta_2 = 0$ à $H_1 : \beta_k \neq 0$ pour au moins un $k \in \{0, 1, 2\}$. Si H_0 est valide, la statistique F^* suit une loi de Fisher avec $p - 1 = 2$ et $n - p = 493$ degrés de liberté.

Mais:

```
(F.star = summary(mod)$fstatistic[1])
```

```
##    value
## 753.8885
```

```
df1 = summary(mod)$fstatistic[2]
df2 = summary(mod)$fstatistic[3]
```

À un niveau de confiance donné par $\alpha = 0.05$, on rejette H_0 si $F^* > F(0.95; 2, 491)$. Puisque

```
F.star > qf(0.95,df1,df2)
```

```
## value
##  TRUE
```

on rejette H_0 en faveur de H_1.

Q26

Déterminez la matrice de variance-covariance estimée $s^2\{\mathbf{b}\}$ pour les estimateurs des moindres carrés $\mathbf{b}$. 'A un niveau de confiance de 95%, testez pour

a) $H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$;

b) $H_0 : \beta_2 = 0$ vs. $H_1 : \beta_2 < 0$.

Solution: la matrice de variance-covariance estimée de $\mathbf{b}$ est

$$\mathrm{s}^2\{\mathbf{b}\} = \mathrm{MSE} \cdot (\mathbf{X}^\top \mathbf{X})^{-1}.$$

Nous avons calculé le vecteur des estimateurs $\mathbf{b}$ et l'erreur quadratique moyenne MSE plus tôt, d'où:

```
(s.2.b = MSE*solve(t(X) %*% X))
```

```
##               intercept            x1            x2
## intercept  0.0429246912 -2.981087e-04 -2.641332e-03
## x1        -0.0002981087  6.010208e-06  1.724343e-06
## x2        -0.0026413316  1.724343e-06  3.372690e-04
```

a) C'est un test bilatéral. À un niveau de confiance donné par $\alpha = 0.05$, on rejette $H_0 : \beta_1 = 0$ si

$$|t^*| = \left| \frac{b_1 - 0}{\mathrm{s}\{b_1\}} \right| > t(0.975; n - p = 491).$$

Puisque

```
t.star = (b[2]-0)/sqrt(s.2.b[2,2])
abs(t.star) > qt(0.975,df2)
```

```
## [1] TRUE
```

on rejette H_0 en faveur de H_1.

b) C'est un test unilatéral à gauche. On rejette $H_0 : \beta_2 = 0$ à un niveau $\alpha = 0.05$, si

$$t^* = \frac{b_2 - 0}{\mathrm{s}\{b_2\}} < -t(0.95; 491).$$

Puisque

```
t.star = (b[3]-0)/sqrt(s.2.b[3,3])
t.star < -qt(0.975,df2)
```

```
## [1] FALSE
```

on ne peux pas rejetter H_0 (ce qui n'est pas la même chose que d'accepter H_0).

Q27

Nous cherchons à prédire la réponse moyenne $E\{Y^*\}$ lorsque $X^* = (20, 5)$. Donnez la valeur ajustée $\hat{Y}^*$ dans ce cas, ainsi qu'un intervalle de confiance à environ 95% de la quantité recherchée.

Solution: le terme constant est sous-entendu:

```
X.star = matrix(c(1,20,5),nrow=1,ncol=3)
```

Nous aurons besoin de **b** et $\mathrm{s}^2\{\mathbf{b}\}$, que nous avons déjà calculé; la valeur ajustée

$$Y^* = \mathbf{X}^* \boldsymbol{b}.$$

```
(y.star = sum(X.star %*% b))
```

```
## [1] 1.926023
```

L'erreur-type est

$$\mathrm{s}\{Y^*\} = \sqrt{\mathbf{X}^* \mathrm{s}^2\{\mathbf{b}\}(\mathbf{X}^*)^\top};$$

```
(se.y.star = as.numeric(sqrt(X.star %*% s.2.b %*% t(X.star))))
```

```
## [1] 0.1255695
```

L'intervalle de confiance de $E\{Y^*\}$ à environ 95% est

$$Y^* \pm t(0.975; n-p=491) \cdot \mathrm{s}\{Y^*\} :$$

```
c(y.star-qt(0.975,df2)*se.y.star,y.star+qt(0.975,df2)*se.y.star)
```

```
## [1] 1.679303 2.172743
```

Q28

Nous cherchons à prédire de nouvelles réponses Y^* lorsque $\mathbf{X}^* = (1, 20, 5)$. Donnez un intervalle de prédiction de Y^* à environ 95%.

Solution: on se sert des calculs des questions précédentes. L'erreur-type est maintenant

$$\mathrm{s}\{\mathrm{pred}^*\} = \sqrt{\mathrm{MSE}}\sqrt{1 + \mathbf{X}^*(\mathbf{X}^\top \mathbf{X})^{-1}(\mathbf{X}^*)^\top};$$

```
(se.pred = as.numeric(sqrt(MSE*(1+X.star %*% solve(t(X) %*% X) %*% t(X.star)))))
```

```
## [1] 2.081927
```

L'intervalle de confiance de Y^*_{pred} à environ 95% est

$$Y^* \pm t(0.975; 491) \cdot \mathrm{s}\{\mathrm{pred}^*\}:$$

```
c(y.star-qt(0.975,df2)*se.pred,y.star+qt(0.975,df2)*se.pred)
```

```
## [1] -2.164563  6.016608
```

L'intervalle de prédiction contient alors l'intervalle de confiance (et des valeurs négatives...).

Q29

a) Donnez des intervalles de confiance simultanés des paramètres β_0, β_1, et β_2 à environ 95%.

b) Donnez des intervalles de confiance simultanés de $E\{Y_\ell^*\}$ à l'aide de la procédure WH pour

$$\mathbf{X}_1^* = (1, 50, 10), \mathbf{X}_2^* = (1, 20, 5), \mathbf{X}_3^* = (1, 200, 8).$$

Solution:

a) Le facteur de Bonferroni est $t\left(1 - \frac{0.05/3}{2}; 491\right)$. Les intervalles de confiance simultanés sont ainsi:

$$\mathrm{IC_B}(\beta_k; 0.95) \equiv b_k \pm 2.2402 \cdot \mathrm{s}\{b_k\}.$$

```
c(b[1]-qt(1-(0.05/3)/2,491)*sqrt(s.2.b[1,1]),b[1]+qt(1-(0.05/3)/2,491)*sqrt(s.2.b[1,1]))
```

```
## [1] -0.5117470  0.4836465
```

```
c(b[2]-qt(1-(0.05/3)/2,491)*sqrt(s.2.b[2,2]),b[2]+qt(1-(0.05/3)/2,491)*sqrt(s.2.b[2,2]))
```

```
## [1] 0.08926843 0.10104682
```

```
c(b[3]-qt(1-(0.05/3)/2,491)*sqrt(s.2.b[3,3]),b[3]+qt(1-(0.05/3)/2,491)*sqrt(s.2.b[3,3]))
```

```
## [1] -0.03673221  0.05150047
```

b) Les intervalles de confiance recherché prennent la forme

$$\hat{Y}_\ell^* \pm \sqrt{pF(1-\alpha; p, n-p)} \cdot \mathrm{s}\{\hat{Y}_\ell^*\} = \mathbf{X}_\ell^* \mathbf{b} \pm \sqrt{3F(0.95; 3, 491)} \cdot \sqrt{\mathrm{MSE}} \sqrt{\mathbf{X}_\ell^* (\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{X}_\ell^*)^\top}.$$

Ainsi, les intervalles de confiance simultanés de la valeur moyenne $E\{Y_\ell^*\}$ sont:

```
WH = sqrt(3*qf(0.95,3,491))
X1.star = matrix(c(1,50,10),nrow=1,ncol=3)
c(X1.star %*% b - WH*sqrt(MSE)*sqrt(X1.star %*% solve(t(X) %*% X) %*% t(X1.star)),
  X1.star %*% b + WH*sqrt(MSE)*sqrt(X1.star %*% solve(t(X) %*% X) %*% t(X1.star)))
```

```
## [1] 4.526634 5.108711
```

```
X2.star = matrix(c(1,20,5),nrow=1,ncol=3)
c(X2.star %*% b - WH*sqrt(MSE)*sqrt(X2.star %*% solve(t(X) %*% X) %*% t(X2.star)),
  X2.star %*% b + WH*sqrt(MSE)*sqrt(X2.star %*% solve(t(X) %*% X) %*% t(X2.star)))
```

```
## [1] 1.573774 2.278271
```

```
X3.star = matrix(c(1,200,8),nrow=1,ncol=3)
c(X3.star %*% b - WH*sqrt(MSE)*sqrt(X3.star %*% solve(t(X) %*% X) %*% t(X3.star)),
  X3.star %*% b + WH*sqrt(MSE)*sqrt(X3.star %*% solve(t(X) %*% X) %*% t(X3.star)))
```

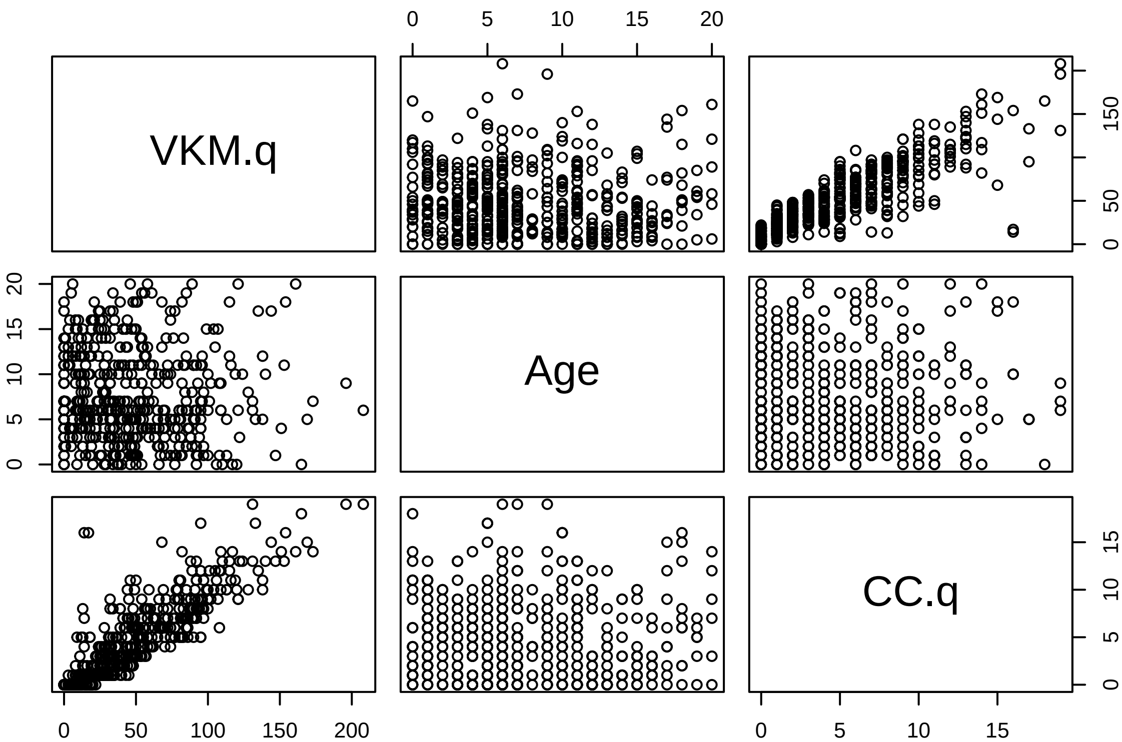
```
## [1] 17.99397 20.15912
```

Q30

Selon vous, est-ce que le modèle de régression linéaire multiple est préférable aux deux modèles de régression linéaire simple pour le même sous-ensemble de `Autos.xlsx` (en utilisant X_1 ou X_2, mais pas les 2)? Soutenez votre réponse.

Solution: on commence par tracer les nuages de points pour chacune des 3 situations.

```
pairs(Autos)
```



On considère trois modèles.

```
mod.1.1 <- lm(y ~ x1 + x2)
mod.1.0 <- lm(y ~ x1)
mod.0.1 <- lm(y ~ x2)
```

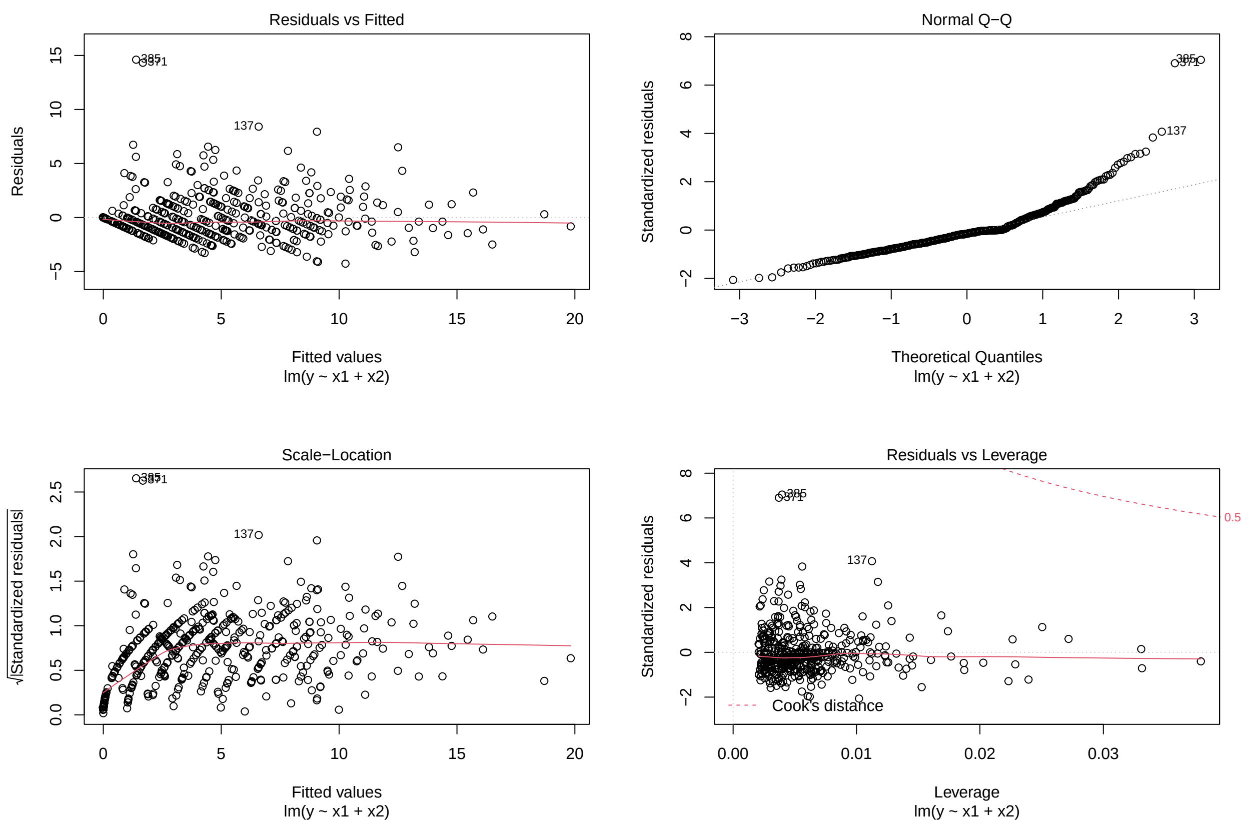
Consultons leur sommaires:

```
summary(mod.1.1)
```

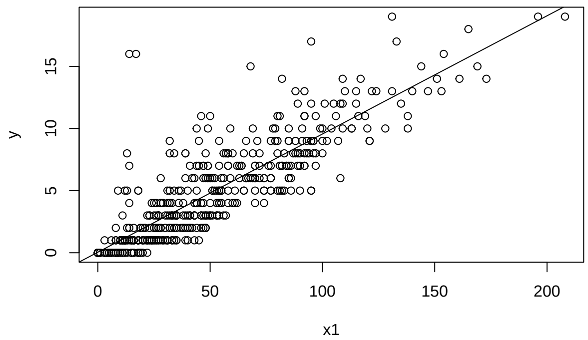
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -4.2704 -1.2115 -0.3180  0.6609 14.6080
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.014050   0.207183  -0.068    0.946
## x1           0.095158   0.002452  38.815   <2e-16 ***
## x2           0.007384   0.018365   0.402    0.688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.078 on 491 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7533
## F-statistic: 753.9 on 2 and 491 DF,  p-value: < 2.2e-16
```

```r
plot(mod.1.1)
```



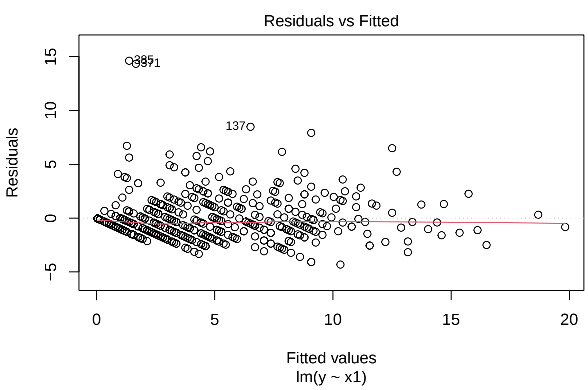

```r
plot(x1,y)
abline(mod.1.0)
```

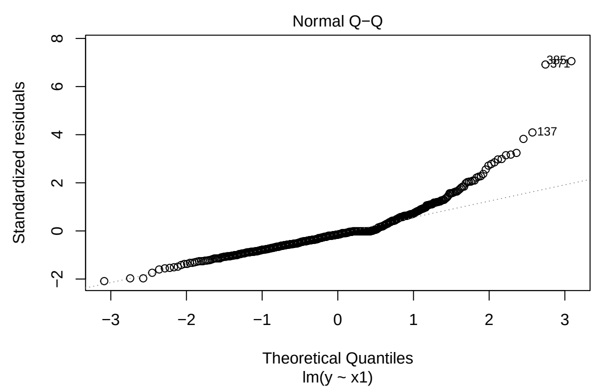


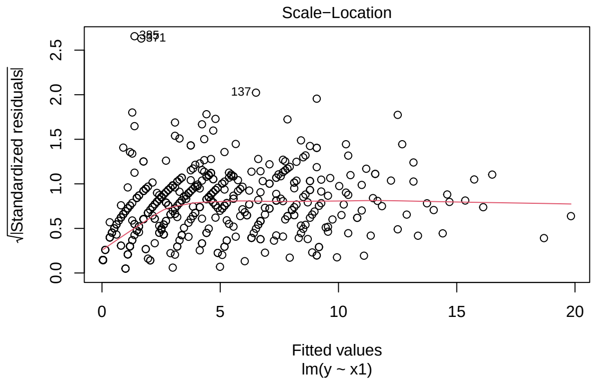

```
summary(mod.1.0)
```

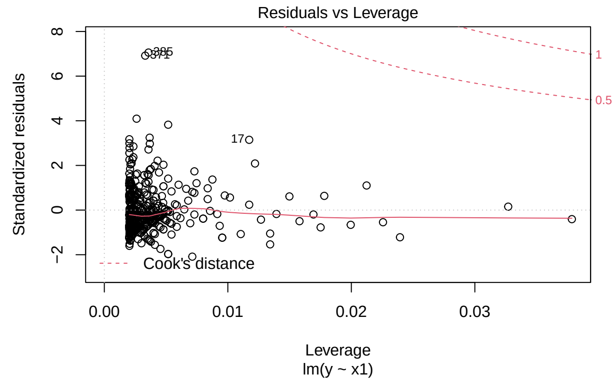
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3167 -1.1852 -0.3217  0.7075 14.6245
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.043779   0.149000   0.294    0.769
## x1          0.095120   0.002448  38.861   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.076 on 492 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7538
## F-statistic:  1510 on 1 and 492 DF,  p-value: < 2.2e-16
```

```
plot(mod.1.0)
```

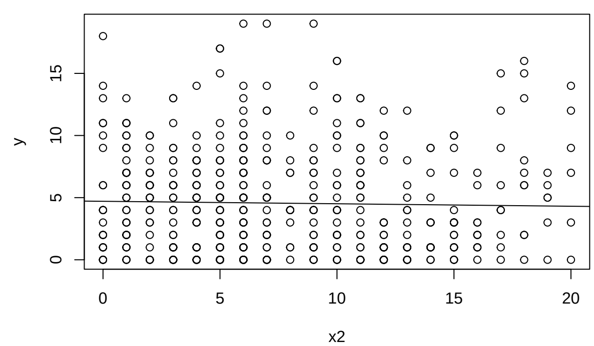








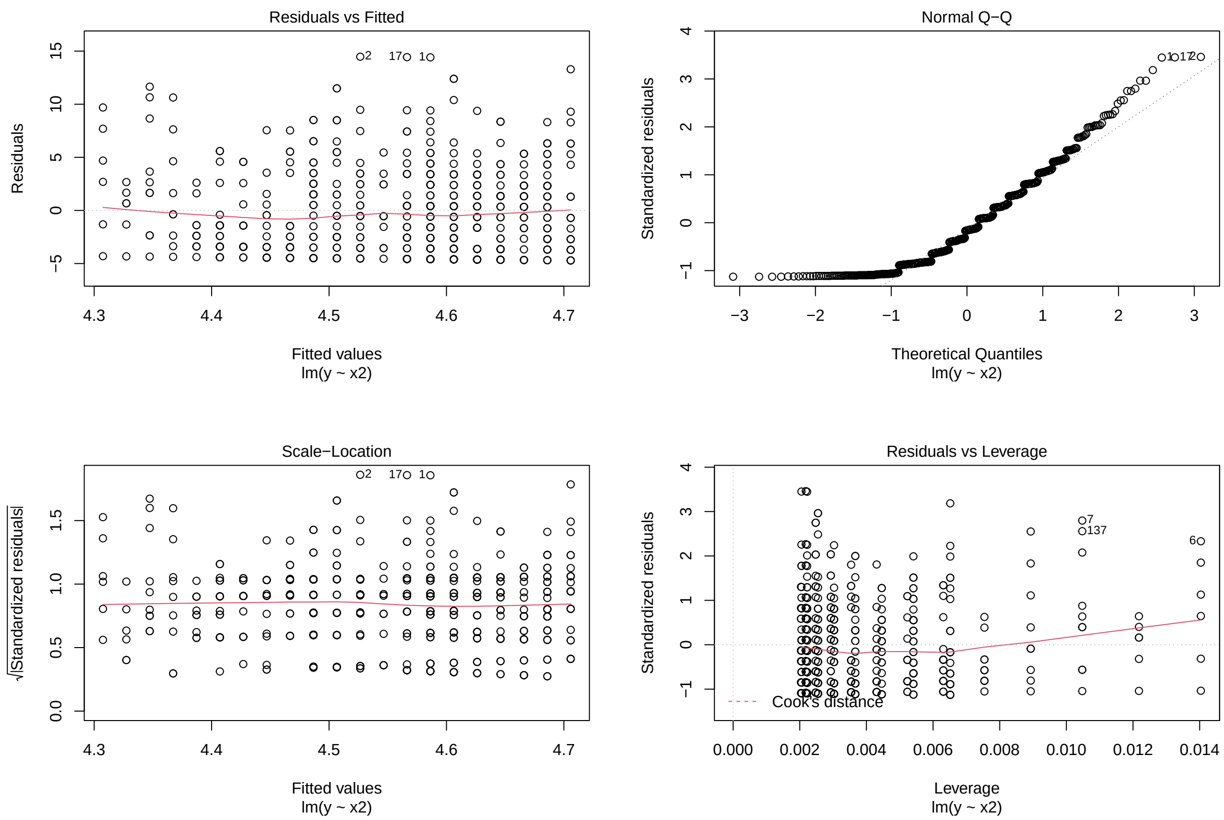

```r
plot(x2,y)
abline(mod.0.1)
```




```r
summary(mod.0.1)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7058 -3.5465 -0.6759  2.4685 14.4734
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.70581    0.33800  13.922   <2e-16 ***
## x2          -0.01992    0.03698  -0.539     0.59
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.187 on 492 degrees of freedom
## Multiple R-squared:  0.0005893,  Adjusted R-squared:  -0.001442
## F-statistic: 0.2901 on 1 and 492 DF,  p-value: 0.5904
```

```
plot(mod.0.1)
```



En terme de toutes les statistiques, il semblerait que le modèle `CC.q ~ VKM.q` soit meilleur que les autres.