Projet Groupe 6

David CAKPOSSE & Hippolite SODJINOU

2024-06-24

Contents

Nous allons utiliser la bibliothèque **faraway** pour tous les exercices.

```
library(faraway)
```

```
## Warning: le package 'faraway' a été compilé avec la version R 4.3.3
```

```
## Warning in check_dep_version(): ABI version mismatch:
## lme4 was built with Matrix ABI version 1
## Current Matrix ABI version is 0
## Please re-install lme4 from source or restore original 'Matrix' package
```

Chapitre 3

Problème 3

En Utilisant les donnés teengamb, réalisons un modèle avec gamble comme réponse et l'autre variables comme prédicteurs.

```
data(teengamb)
g=lm(gamble ~ sex + status + income + verbal,data=teengamb)
summary(g)
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

a)

D'après l'analyse des p-value, seules les variables "sex" et "income" sont statiquement significatives puisque leurs p_value sont toutes deux inférieures à 0.05.

b) Interpretation du coefficient "sex"

La variable "sex" étant une variable qualitative (male, femelle), on prend comme codage 1 pour "mâle" et 0 pour "femelle". De ce fait, en considérant toutes les autres comme constantes (sauf la variable "sex"), on remarque que le montant moyen de pari (gamble) chez les hommes diminue de 22.11833 que chez les femmes.

c) Prediction et comparaison des Intervalles de Confiances:

- Pour un homme moyen avec les données moyennes

```
data_h=data.frame(teengamb[-5])
new_data_h=data.frame(sex=1,status=mean(data_h[,2]),income=mean(data_h[,3]),verbal=mean(data_h[,4]))
predict(g,new_data_h,level=0.95,interval="conf")
```

```
##        fit       lwr     upr
## 1 6.124186 -5.795024 18.0434
```

-Pour un homme maximal avec les données maximales

```
new_data_hm=data.frame(sex=1,status=max(data_h[,2]),income=max(data_h[,3]),verbal=max(data_h[,4]))
predict(g,new_data_hm,level=0.95,interval="conf")
```

```
##        fit      lwr      upr
## 1 49.18961 12.87959 85.49963
```

On constate que l'intervalle de confiance à 95% pour un homme aves des données maximales est plus large que celui d'un homme avec des données moyennes (i.e un homme moyen).

d) Faisons un autre modèle avec income comme prédicteurs et gamble comme reponse:

```
g2<-lm(gamble ~ income, data = teengamb)
anova(g2,g)
```

```
## Analysis of Variance Table
##
## Model 1: gamble ~ income
## Model 2: gamble ~ sex + status + income + verbal
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1     45 28009
## 2     42 21624  3    6384.8 4.1338 0.01177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En examinant les résultats du test, on constate que la p_value obtenue (0,01177) est inférieure à 0,05. Cela signifie que nous rejetons le modèle "g2". Par conséquent, au moins une des trois variables "sex", "status" ou "verbal" fournit des informations supplémentaires pour prédire le montant moyen des paris ("gamble") chez les hommes et les femmes.

Problème 4

Utilisons les données sat:

```
data(sat)
sat
```

```
##                expend ratio salary takers verbal math total
## Alabama         4.405  17.2 31.144      8    491  538  1029
## Alaska          8.963  17.6 47.951     47    445  489   934
## Arizona         4.778  19.3 32.175     27    448  496   944
## Arkansas        4.459  17.1 28.934      6    482  523  1005
## California      4.992  24.0 41.078     45    417  485   902
## Colorado        5.443  18.4 34.571     29    462  518   980
## Connecticut     8.817  14.4 50.045     81    431  477   908
## Delaware        7.030  16.6 39.076     68    429  468   897
## Florida         5.718  19.1 32.588     48    420  469   889
## Georgia         5.193  16.3 32.291     65    406  448   854
## Hawaii          6.078  17.9 38.518     57    407  482   889
## Idaho           4.210  19.1 29.783     15    468  511   979
## Illinois        6.136  17.3 39.431     13    488  560  1048
```

```
## Indiana         5.826  17.5 36.785     58    415  467   882
## Iowa            5.483  15.8 31.511      5    516  583  1099
## Kansas          5.817  15.1 34.652      9    503  557  1060
## Kentucky        5.217  17.0 32.257     11    477  522   999
## Louisiana       4.761  16.8 26.461      9    486  535  1021
## Maine           6.428  13.8 31.972     68    427  469   896
## Maryland        7.245  17.0 40.661     64    430  479   909
## Massachusetts   7.287  14.8 40.795     80    430  477   907
## Michigan        6.994  20.1 41.895     11    484  549  1033
## Minnesota       6.000  17.5 35.948      9    506  579  1085
## Mississippi     4.080  17.5 26.818      4    496  540  1036
## Missouri        5.383  15.5 31.189      9    495  550  1045
## Montana         5.692  16.3 28.785     21    473  536  1009
## Nebraska        5.935  14.5 30.922      9    494  556  1050
## Nevada          5.160  18.7 34.836     30    434  483   917
## New Hampshire   5.859  15.6 34.720     70    444  491   935
## New Jersey      9.774  13.8 46.087     70    420  478   898
## New Mexico      4.586  17.2 28.493     11    485  530  1015
## New York        9.623  15.2 47.612     74    419  473   892
## North Carolina  5.077  16.2 30.793     60    411  454   865
## North Dakota    4.775  15.3 26.327      5    515  592  1107
## Ohio            6.162  16.6 36.802     23    460  515   975
## Oklahoma        4.845  15.5 28.172      9    491  536  1027
## Oregon          6.436  19.9 38.555     51    448  499   947
## Pennsylvania    7.109  17.1 44.510     70    419  461   880
## Rhode Island    7.469  14.7 40.729     70    425  463   888
## South Carolina  4.797  16.4 30.279     58    401  443   844
## South Dakota    4.775  14.4 25.994      5    505  563  1068
## Tennessee       4.388  18.6 32.477     12    497  543  1040
## Texas           5.222  15.7 31.223     47    419  474   893
## Utah            3.656  24.3 29.082      4    513  563  1076
## Vermont         6.750  13.8 35.406     68    429  472   901
## Virginia        5.327  14.6 33.987     65    428  468   896
## Washington      5.906  20.2 36.151     48    443  494   937
## West Virginia   6.107  14.8 31.944     17    448  484   932
## Wisconsin       6.930  15.9 37.746      9    501  572  1073
## Wyoming         6.160  14.9 31.285     10    476  525  1001
```

a) Realisons un modèle avec "total" comme reponse et "expend", "ratio" et "salary" comme prédicteurs.

```r
satg1<-lm(total ~ expend + ratio + salary,data=sat)
summary(satg1)
```

```
##
## Call:
## lm(formula = total ~ expend + ratio + salary, data = sat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.911  -46.740   -7.535   47.966  123.329
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1069.234    110.925   9.639 1.29e-12 ***
## expend        16.469     22.050   0.747   0.4589
## ratio          6.330      6.542   0.968   0.3383
## salary        -8.823      4.697  -1.878   0.0667 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68.65 on 46 degrees of freedom
## Multiple R-squared:  0.2096, Adjusted R-squared:  0.1581
## F-statistic: 4.066 on 3 and 46 DF,  p-value: 0.01209
```

-Testons l'hypothèse $\beta_{Salary} = 0$

Pour tester $\beta_{Salary} = 0$, nous allons faire le test suivant:

$$\begin{cases} H_0 : \beta_{Salary} = 0 \\ H_1 : \beta_{Salary} \neq 0 \end{cases}$$

La p_value de ce test est 0.0667 superieure au seuil(0.05), donc l'hypothèse nulle ne peut pas etre rejeter. Ainsi la variable "salary" n'est pas significative.

-Testons l'hypothèse que $\beta_{Salary} = \beta_{ratio} = \beta_{expend} = 0$

Pour tester l'hypothèse $\beta_{Salary} = \beta_{ratio} = \beta_{expend} = 0$,nous allons faire le test suivent:

$$\begin{cases} H_0 : \beta_{Salary} = \beta_{ratio} = \beta_{expend} = 0 \\ H_1 : l'uneaumoinsdesvariables(\text{"}expend\text{"}, \text{"}ratio\text{"}, \text{"}salary\text{"})est \quad significative \end{cases}$$

Nous remarquons que la p_value de ce test est 0,012 ce qui est inférieure au seuil(0,05). Par consequent on rejete l'hypothèse nulle H_0. L'une au moins des variables explicatives est importante pour predire "total".

b) Réalisons un autre modèle en ajoutant "takers":

```
satg2<-lm(total ~ expend + ratio + salary + takers,data=sat)
summary(satg2)
```

```
##
## Call:
## lm(formula = total ~ expend + ratio + salary + takers, data = sat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.531 -20.855  -1.746  15.979  66.571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
## expend         4.4626    10.5465   0.423    0.674
## ratio         -3.6242     3.2154  -1.127    0.266
## salary         1.6379     2.3872   0.686    0.496
## takers        -2.9045     0.2313 -12.559 2.61e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.7 on 45 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.809
## F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

Testons l'hypothèse $\beta_{Salary} = 0$

$$\begin{cases} H_0 : \beta_{Salary} = 0 \\ H_1 : \beta_{Salary} \neq 0 \end{cases}$$

Nous remarquons que la p_value de ce test est supérieure (0.496>0.05) au seuil, donc nous ne pouvons pas rejeter l'hypothèse nulle. La variable "salary" n'est donc pas significative. Cela confirme la conclusion tirée pour la signifiativité de la variable "salary" pour le modèle "satg1".

- Comparons ce modèle au modèle précédent:

```
anova(satg1,satg2)
```

```
## Analysis of Variance Table
##
## Model 1: total ~ expend + ratio + salary
## Model 2: total ~ expend + ratio + salary + takers
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     46 216812
## 2     45  48124  1    168688 157.74 2.607e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La p_value du test est inférieure au seuil, donc on rejette le modèle "satg1".Le modéle avec la variable "takers" au modèle sans cette variable.

Chapitre 4

Problème 4

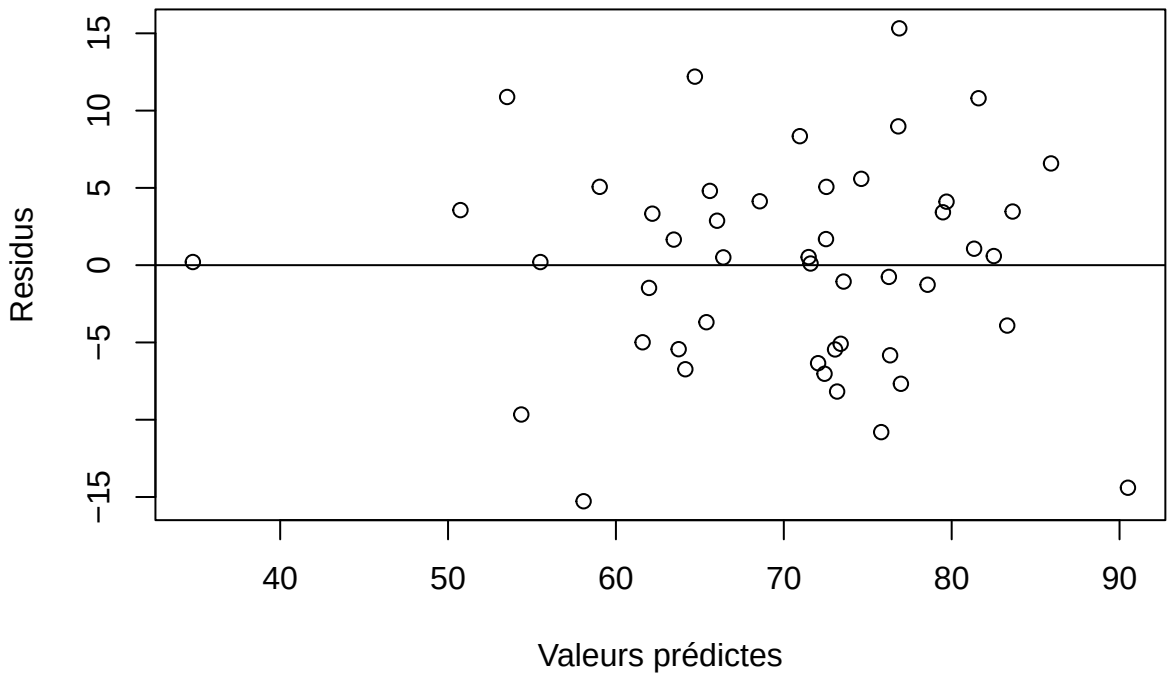
A partir des données “swiss”, réalisons un modèle ayant pour réponse “Fertility” et les autres variables comme prédicteurs.

```
data("swiss")
swissg=lm(Fertility~Agriculture+Examination+Education+Infant.Mortality+Catholic,swiss)

swissgr1 = summary(swissg)
swissgr1
```

```
##
## Call:
## lm(formula = Fertility ~ Agriculture + Examination + Education +
##     Infant.Mortality + Catholic, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
## Examination      -0.25801    0.25388  -1.016  0.31546
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## Catholic          0.10412    0.03526   2.953  0.00519 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.165 on 41 degrees of freedom
## Multiple R-squared:  0.7067, Adjusted R-squared:  0.671
## F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

a) Vérifions l’hypothèse de la variance constante pour les erreurs

```
plot(residuals(swissg)~fitted(swissg),xlab="Valeurs prédictes",ylab = "Residus")
abline(h=0)
```



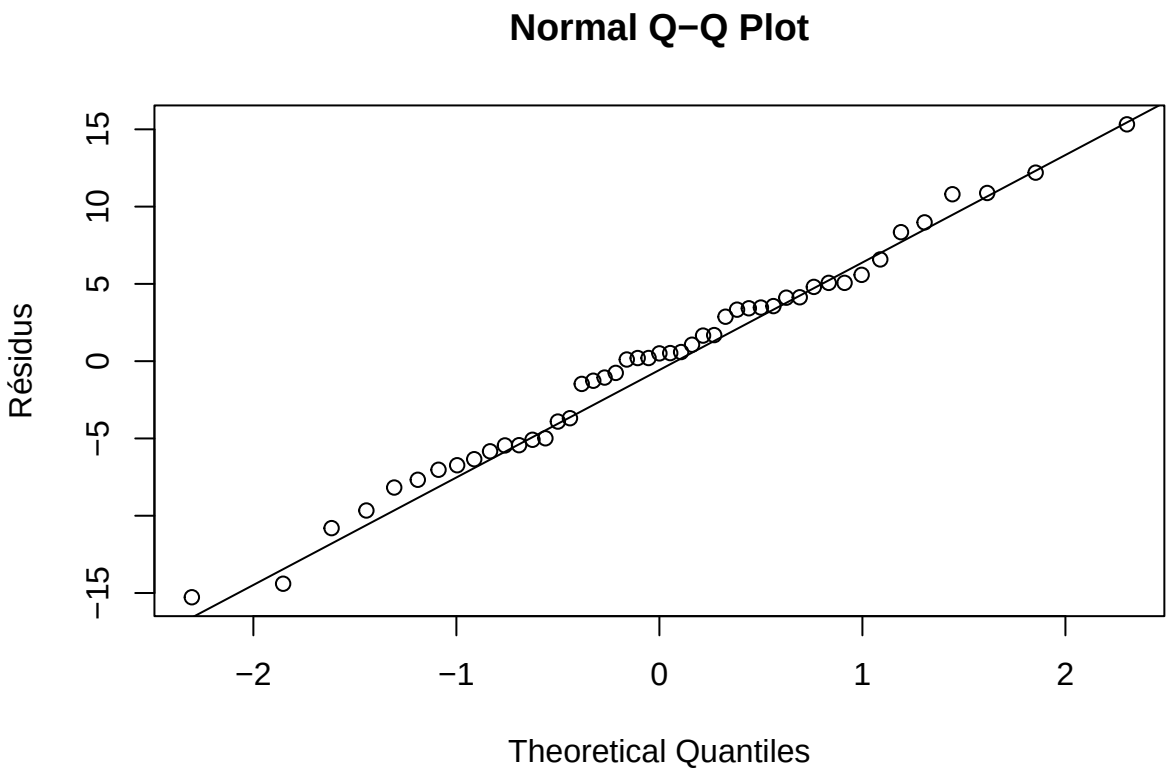
Nous observons graphiquement un nuage de points non répartis uniformement. Ainsi l'hypothèse de variance constante pour les erreurs n'est pas vérifiée.

b) Vérifions l'hypothèse de normalité:

Nous pouvons la verifiée graphiquement (Q-Q plot) ou par le test de Shapiro-Wilk

Nous avons: **- Graphiquement:**

```
qqnorm(residuals(swissg),ylab = "Résidus")
qqline(residuals(swissg))
```



En examinant le graphique, nous remarquons que les erreurs suivent une distribution normale. Donc l'hypothèse de normalité des errreurs est vérifiée. \

- Shapiro-Wilk's test:

```
shapiro.test(residuals(swissg))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(swissg)
## W = 0.98892, p-value = 0.9318
```

La p_value du test est 0.9318 qui est supérieure au seuil (0.05), donc l'hypothèse nulle ne peut être rejetée. C'est-à-dire que les erreurs sont normales.

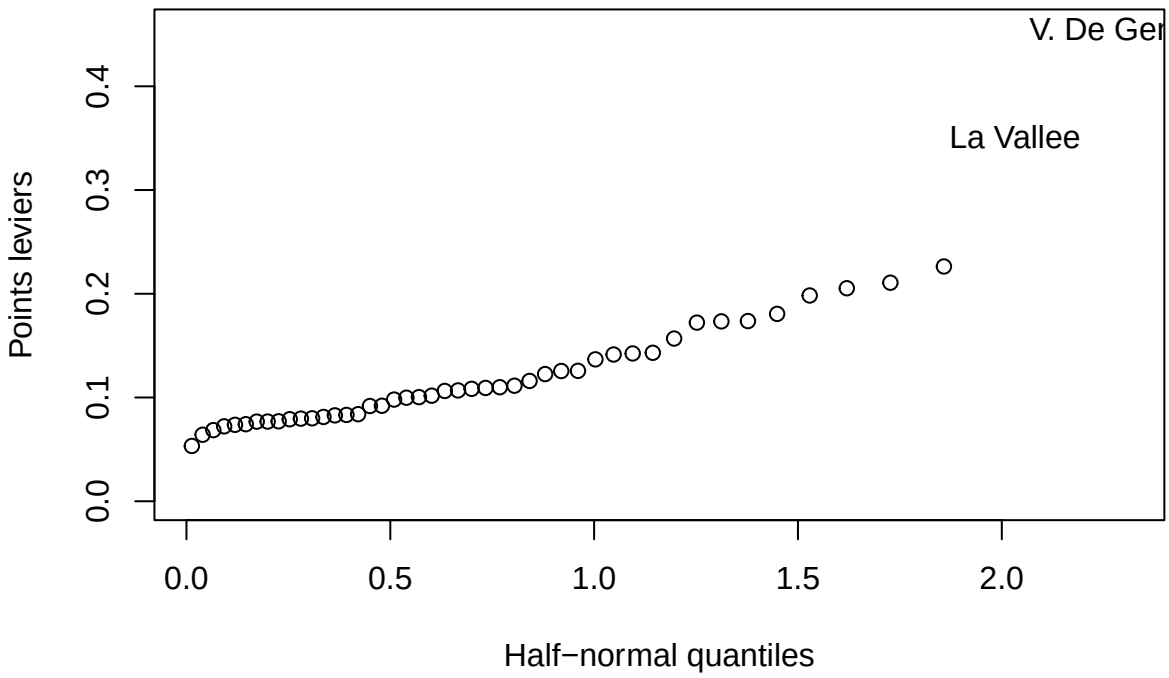
c) Recherchons les points leviers important:

```
swissginf=influence(swissg)
sum(swissginf$hat)
```

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

Nous observons qu'il y a six points leviers. Pour identifier les plus importants, nous procédons comme suit:

```
noms=row.names(swiss)
halfnorm(swissginf$hat, labs=noms , ylab="Points leviers")
```



En examinant le graphique, nous remarquons que les deux observations qui sont au dessus du seuil (2p/n = 2*6/47 = 0.2553191) sont : “La Vallee” et “V. De Geneve”.

d) Vérifions les valeurs abérrantes

```
v_abber=rstudent(swissg)
v_abber[which.max(abs(v_abber))]
```

```
##   Sierre
## 2.445227
```

Nous constatons que l’individu “Sierre” est susceptible d’être une valeur aberrante. Vérifions si c’est réellement le cas.

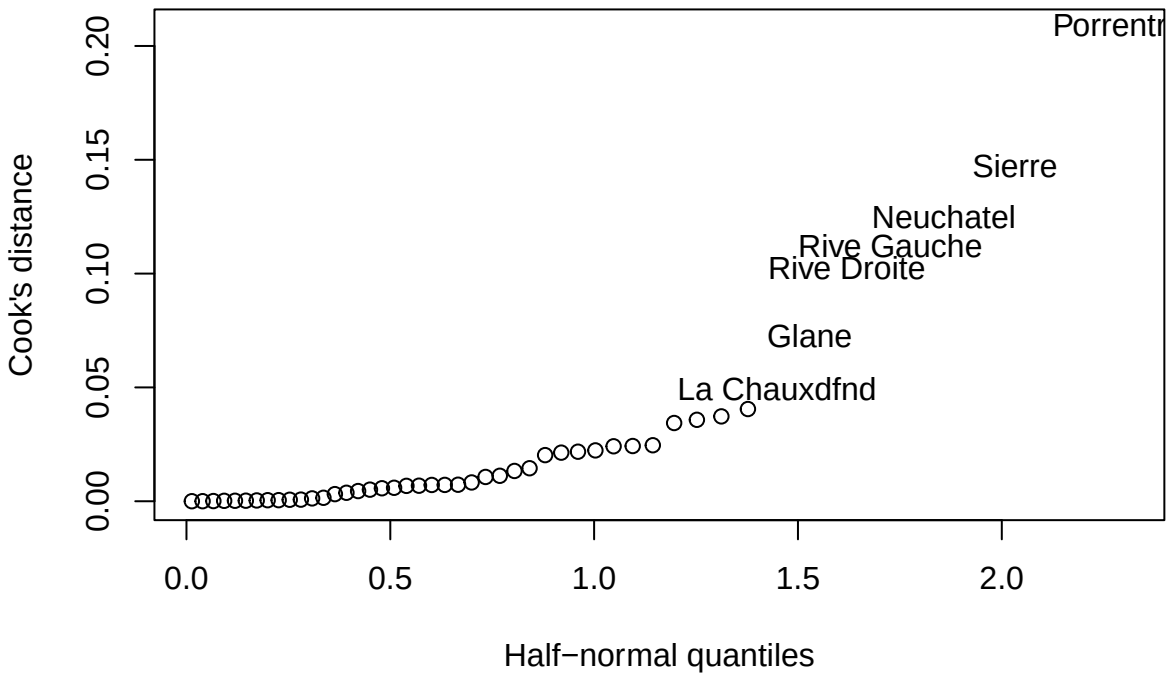
```
qt(.05/(47*2), 40)
```

```
## [1] -3.529468
```

Nous constatons que 2.445227 est inférieur à 3.529468, ce qui indique que “Sierre” n’est pas une valeur aberrante. Par conséquent, il n’y a pas de valeurs aberrantes.

e) Vérifions les points d’influences

```
cook=cooks.distance(swissg)
halfnorm(cook, 7, labs=noms, ylab="Cook's distance")
```



Du point de vue de la distance de Cook, nous pouvons identifier sept individus ayant des valeurs élevées. Cependant, la plus importante est "Porrentruy". Voyons ce qui se passe lorsqu'on la retire du modèle (obtenant ainsi le modèle swissg2).

```
swissg2 = lm(Fertility~.,data= swiss,subset = (cook<max(cook)))
swissgr = summary(swissg2)

Model_avec_Sierre=c(swissgr1$r.squared,swissgr1$sigma,swissgr1$coef[,1])
Model_sans_Sierre=c(swissgr$r.squared,swissgr$sigma,swissgr$coef[,1])

comp_values=c("R^2","Sigma","B_0","B_1","B_2","B_3","B_4","B_5")
data.frame(comp_values,Model_avec_Sierre,Model_sans_Sierre)
```

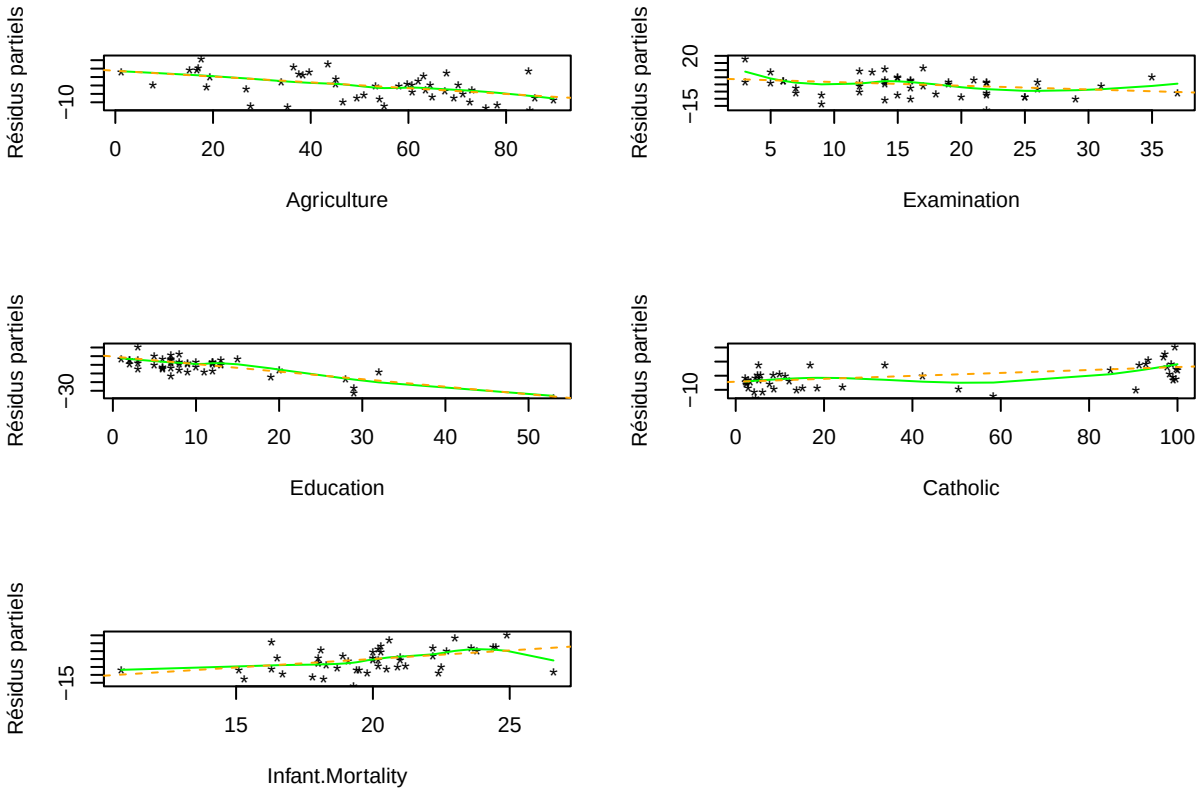
```
##   comp_values Model_avec_Sierre Model_sans_Sierre
## 1         R^2         0.7067350         0.7414649
## 2       Sigma         7.1653688         6.7940749
## 3         B_0        66.9151817        65.4555408
## 4         B_1        -0.1721140        -0.2103426
## 5         B_2        -0.2580082        -0.3227756
## 6         B_3        -0.8709401        -0.8950603
## 7         B_4         1.0770481         0.1126858
## 8         B_5         0.1041153         1.3156652
```

Nous constatons que le R-squared du modèle sans l'individu "Sierre" est supérieur à celui du modèle avec l'individu "Sierre". Cela signifie que la variation de la valeur prédite ("Fertility") est mieux expliquée par les prédicteurs du modèle sans l'individu "Sierre" que par ceux du modèle avec cet individu. De plus, le sigma du modèle sans l'individu "Sierre" est inférieur à celui du modèle avec cet individu, ce qui est favorable.Il

serait donc approprié de retirer cette observation (“Sierre”) des données, car elle constitue une observation influente.

f) Vérifions la structure de la relation entre les prédicteurs et la réponse

```
rpartiels =resid(swissg,type="partial")
par(mfrow=c(3,2))
for(i in 1:5)
{
prov=loess(rpartiels[,names(swiss)[i+1]]~swiss[,i+1])
ordre=order(swiss[,i+1])
plot(swiss[,i+1],rpartiels[,names(swiss)[i+1]],pch="*",xlab = names(swiss)[i+1],ylab = "Résidus partiels
matlines(swiss[,i+1][ordre],predict(prov)[ordre],col = "green")
abline(lsfit(swiss[,i+1],rpartiels[,names(swiss)[i+1]]),col="orange",lty=2)
}
```



Les graphiques des résidus partiels pour les variables telles que “Agriculture”, “Examination”, “Education”, “Catholic” et “Infant.Mortality” indiquent qu’aucune transformation n’est requise, car les résidus partiels sont uniformément répartis le long de la ligne de régression ajustée (représentée en pointillés). La courbe rouge illustre un résumé lissé des données.

Problème 5

A partir des données "divusa", réalisons le modèle ayant comme réponse "divorce" et les autres variables exceptées "year" comme prédicteurs:

```
data("divusa")
divusag1=lm(divorce~unemployed+femlab+marriage+birth+military,data=divusa)
divusag=summary(divusag1)
divusag
```

```
##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8916 -0.0496  0.8650  3.8300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48784    3.39378   0.733   0.4659
## unemployed  -0.11125    0.05592  -1.989   0.0505 .
## femlab       0.38365    0.03059  12.543  < 2e-16 ***
## marriage     0.11867    0.02441   4.861 6.77e-06 ***
## birth       -0.12996    0.01560  -8.333 4.03e-12 ***
## military    -0.02673    0.01425  -1.876   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 71 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9152
## F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

Vérifions l'hypothèse d'auto-corrélation des erreurs:

Pour cela, nous allons procéder au test de Durbin-Watson afin d'évaluer la corrélation des erreurs :

```
library(lmtest)
```

```
## Warning: le package 'lmtest' a été compilé avec la version R 4.3.3
```

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

```
## Warning: le package 'zoo' a été compilé avec la version R 4.3.3
```

```
##
## Attachement du package : 'zoo'
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(divorce~unemployed+femlab+marriage+birth+military,data=divusa)
```

```
##
```

```
##  Durbin-Watson test
##
## data:  divorce ~ unemployed + femlab + marriage + birth + military
## DW = 0.29988, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Nous observons que la p-value de ce test est inférieure à 0.05 (2.2e-16 $<$ 0.05), donc nous rejetons l'hypothèse nulle. Cela signifie que les erreurs sont autocorrélées.

Chapitre 5

Problème 3

a) A partir des données "divusa", réalisons le modèle ayant comme réponse "divorce" et les "unemployed", "femlab", "marriage", "birth" et "military" comme prédicteurs:

Remarquons que c'est le même modèle que celui du problème 5 du chapitre 4. Donc, pour éviter de refaire le même modèle, utilisons directement celui qui a déjà été réalisé.

Calculons les nombres de condition ("condions numbers")

```
x =model.matrix(divusag1)[,-1]
val_propres= eigen(t(x) %*% x)
val_propres$values
```

```
## [1] 1174600.548   21261.741   16133.842    6206.181    1856.894
```

```
sqrt(val_propres$values[1]/val_propres$values)
```

```
## [1]  1.000000  7.432684  8.532498 13.757290 25.150782
```

Nous obtenons des valeurs propres assez élevées, mais le plus grand nombre de condition (condition number) est de 25.150782. Ainsi, aucun des nombres de condition n'est supérieur à 30. Cela suggère qu'il semble n'y avoir qu'une seule combinaison linéaire qui pose un problème de colinéarité.

b) Pour le même modèle, calculons les facteurs d'inflation de la variance (Vif) :

```
vif(x)
```

```
## unemployed     femlab   marriage      birth   military
##   2.252888   3.613276   2.864864   2.585485   1.249596
```

Les valeurs obtenues indiquent une faible colinéarité entre les variables. Cependant, nous ne disposons pas de suffisamment d'informations pour affirmer que la colinéarité rend certains prédicteurs non significatifs, car les facteurs d'inflation de la variance sont faibles.

c)

En retirant a priori les variables non significatives, on pourrait potentiellement réduire le risque de colinéarité. Cependant, cela entraînerait probablement une diminution du pourcentage de variation de la réponse expliquée par les prédicteurs.

(Investigation) Selon la sortie de la régression linéaire, les variables non significatives sont "unemployed" et "military", car la p-value du t-test associé à chacune de ces variables est supérieure au seuil de 5%.

Faisons une nouvelle regression sans ces deux variables et voyons voir ce qui change.

```
divusagr=update(divusag1,.~.-(unemployed+military))
div= summary(divusagr)
x_1=model.matrix(divusagr)[,-1]
val_propres1=eigen(t(x_1) %*% x_1)
val_propres1$values
```

```
## [1] 1158413.60   20972.53    6208.46
```

```r
sqrt(val_propres1$values[1]/val_propres1$values)
```

```
## [1]  1.000000  7.432012 13.659660
```

```r
vif(x_1)
```

```
##   femlab marriage    birth
## 1.893390 2.201891 2.008469
```

```r
divusag$r.squared
```

```
## [1] 0.9208209
```

```r
div$r.squared
```

```
## [1] 0.9140885
```

Nous observons que non seulement les facteurs d'inflation ont légèrement diminué dans ce nouveau modèle, mais aussi le pourcentage de variance expliquée a diminué..Cela confirme ce qu'on avait affirmé au début de la question.

Chapitre 6

Problème 1

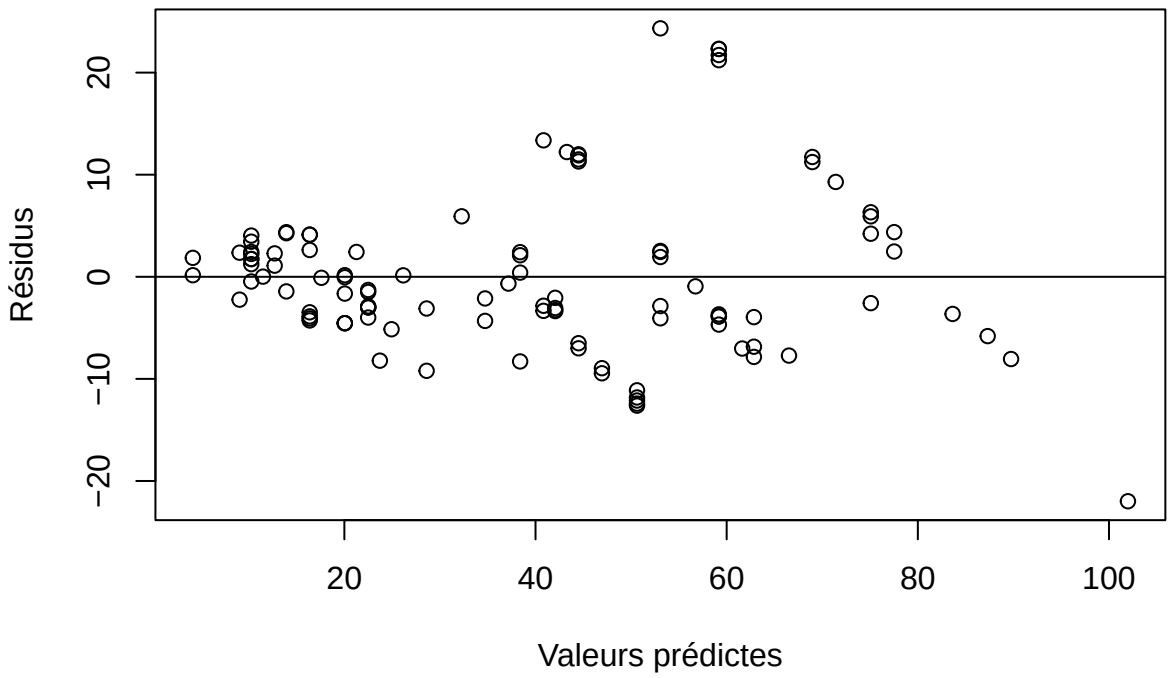
a) Ajustons le modèle de régression Lab Fiel

```
data("pipeline")
pipelg1=lm(Lab~Field,data=pipeline)
summary(pipelg1)
```

```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.985  -4.072  -1.431   2.504  24.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96750    1.57479  -1.249    0.214
## Field        1.22297    0.04107  29.778   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.865 on 105 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8931
## F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

Vérifions l'hypothèse d'homoscédasticité (variance constante)

```
plot(fitted(pipelg1), residuals(pipelg1), xlab="Valeurs prédictes", ylab="Résidus")
abline(h=0)
```



Nous observons un nuage de points non uniformément répartis, ce qui indique que l'hypothèse de variance constante n'est pas vérifiée.

b) Ajustons le modèle par la méthode WLS (Weighted Least Square)

```
i=order(pipeline$Field)

n_pipel= pipeline[i,]
ff= gl(12,9)[-108]

meanfield= unlist(lapply(split(n_pipel$Field,ff),mean))
varlab=unlist(lapply(split(n_pipel$Lab,ff),var))

pipelg2=lm(log(varlab)~log(meanfield),weights = 1/meanfield)
```

Résumé de la régression:

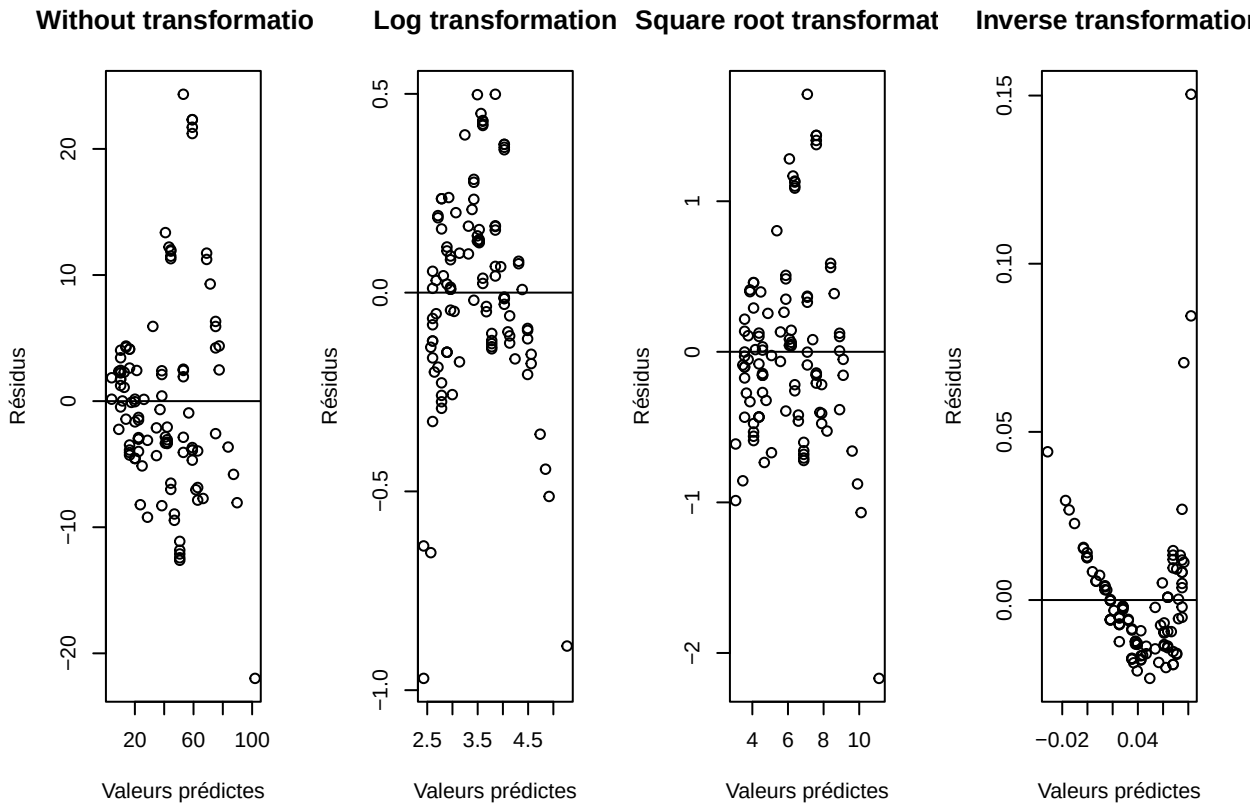
```
summary(pipelg2)
```

```
##
## Call:
## lm(formula = log(varlab) ~ log(meanfield), weights = 1/meanfield)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25034 -0.12237  0.02634  0.13204  0.17924
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.07856    1.02902  -0.076   0.9406
## log(meanfield)  1.03344    0.34426   3.002   0.0133 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.158 on 10 degrees of freedom
## Multiple R-squared:  0.474,  Adjusted R-squared:  0.4214
## F-statistic: 9.011 on 1 and 10 DF,  p-value: 0.0133
```

c) Recherchons des transformations adéquates afin que la relation soit approximativement linéaire avec la variance constante

```
par(mfrow=c(1,4))
plot(fitted(pipelg1), residuals(pipelg1),main = "Without transformation", xlab="Valeurs prédictes", ylab

abline(h=0)

pipelgr1=lm(log(Lab)~Field,data=pipeline)

plot(fitted(pipelgr1), residuals(pipelgr1),main = "Log transformation", xlab="Valeurs prédictes", ylab=
abline(h=0)

pipelgr2= lm(sqrt(Lab)~Field,data=pipeline)

plot(fitted(pipelgr2), residuals(pipelgr2),main = "Square root transformation", xlab="Valeurs prédictes

abline(h=0)

pipelgr3=lm(1/(Lab)~Field,data=pipeline)
plot(fitted(pipelgr3), residuals(pipelgr3),main = "Inverse transformation", xlab="Valeurs prédictes", yl
abline(h=0)
```




```
par(mfrow=c(1,1))
```

On peut remarquer que la transformation en “Square Root” est celle qui améliore le plus le nuage de points du modèle initial, car aucune tendance particulière n’est observée. De plus, les points sont mieux répartis de part et d’autre de l’axe des abscisses, contrairement aux autres transformations.

Chapitre 8

Problème 5

a) Ajustons un modèle linéaire à partir des données "stackloss", avec "stack.loss" comme réponse et les autres variables comme prédicteurs:

```
data("stackloss")
stacklossg1=lm(stack.loss~.,data=stackloss)
summary(stacklossg1)
```

```
##
## Call:
## lm(formula = stack.loss ~ ., data = stackloss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2377 -1.7117 -0.4551  2.3614  5.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
## Water.Temp    1.2953     0.3680   3.520  0.00263 **
## Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

```
p_aberr1=rstudent(stacklossg1)
p_aberr1
```

```
##           1           2           3           4           5           6
##  1.20947467 -0.70513857  1.61790411  2.05179748 -0.53050364 -0.96320379
##           7           8           9          10          11          12
## -0.82594672 -0.47365206 -1.04858585  0.42618802  0.87829204  0.96670672
##          13          14          15          16          17          18
## -0.46873058 -0.01695002  0.80061639  0.29118502 -0.59958579 -0.14868029
##          19          20          21
## -0.19719938  0.44311701 -3.33049332
```

b) Simplifions le modèle.

D'après le résumé, il apparaît que la variable "Acid.Conc." n'est pas significative pour ce modèle avec un seuil de risque de 5% car sa p-value (0.34405) est supérieure au seuil.

Ajustons alors un modèle sans cette variable:

```
stacklossg2=update(stacklossg1,.~.-Acid.Conc.)
stacklossgr2=summary(stacklossg2)
```

Il est tout à fait normal que le pourcentage de variance expliquée diminue légèrement. En revanche, la p-value du test de Fisher est plus significative, ce qui souligne l'importance des variables "Air.Flow" et "Water.Temp" dans la prédiction de "stack.loss".

c) Vérifions pour le modèle (initial), les points abérrants et les points influents

Points abérrants

```
p_aberr2=rstudent(stacklossg2)
p_aberr2
```

```
##           1           2           3           4           5           6
##  1.37707925 -0.46568693  1.64941219  2.02664163 -0.54418189 -0.97790028
##           7           8           9          10          11          12
## -1.11227162 -0.76532659 -1.12436660  0.68918371  0.68918371  0.82677955
##          13          14          15          16          17          18
## -0.28485886 -0.37014929  0.47631886  0.15460185  0.06056767  0.06056767
##          19          20          21
## -0.03447648  0.58346743 -3.47073200
```

```
p_aberr2[which.max(abs(p_aberr2))]
```

```
##        21
## -3.470732
```

Le quantile du modèle ajusté:

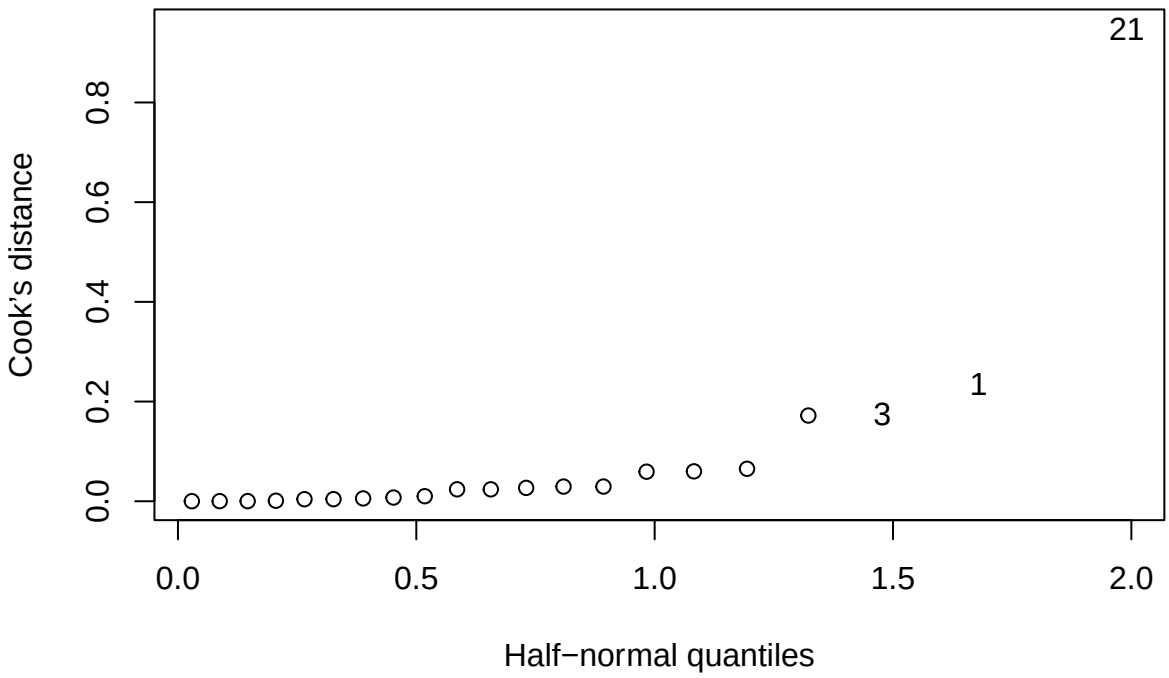
```
qt(0.05/(2*21),17)
```

```
## [1] -3.565438
```

On constate que 3.470732 est inférieure à 3.565438, donc on conclut que l'observation "21" n'est pas une valeur abérrante. Par conséquent il n'y a pas de valeurs abérrantes.

- *Points influents*

```
states=row.names(stackloss)
cook1 =cooks.distance(stacklossg2)
halfnorm(cook1, 3, labs=states, ylab="Cook's distance")
```



Le graphique de la "Cook's distance" montre que l'observation 21 semble être influente, car elle est très éloignée des autres observations.

Pour déterminer s'il est préférable de la supprimer du modèle ou non, effectuons une comparaison avec le modèle dans lequel cette observation a été supprimée:

```
stacklossg3=lm(stack.loss~Air.Flow+Water.Temp,subset=(cook1<max(cook1)),data=stackloss)
stacklossgr3=summary(stacklossg3)

Model_avec_21=c(stacklossgr2$r.squared,stacklossgr2$sigma,stacklossgr2$coef[,1])

Model_sans_21=c(stacklossgr3$r.squared,stacklossgr3$sigma,stacklossgr3$coef[,1])

c_values =c("R^2","Sigma","B_0","B_1","B_2")

data.frame(c_values,Model_avec_21,Model_sans_21)
```

```
##   c_values Model_avec_21 Model_sans_21
## 1      R^2     0.9087609     0.9464265
## 2    Sigma     3.2386154     2.5494859
## 3      B_0   -50.3588401   -51.0759778
## 4      B_1     0.6711544     0.8630041
## 5      B_2     1.2953514     0.8032569
```

Nous observons que les valeurs estimées des coefficients ont changé. Cela montre que les estimations sont sensibles à la présence de l'observation 21. Nous préconisons donc de retirer cette observation des données, car il s'agit d'une observation influente.

d) Effectuons les mêmes vérifications pour le modèle initial ("stacklossmdl1")

- *Points abérrants*

```
p_aberr1 =rstudent(stacklossg1)
p_aberr1[which.max(abs(p_aberr1))]
```

```
##        21
## -3.330493
```

Le quantile du modèle initial:

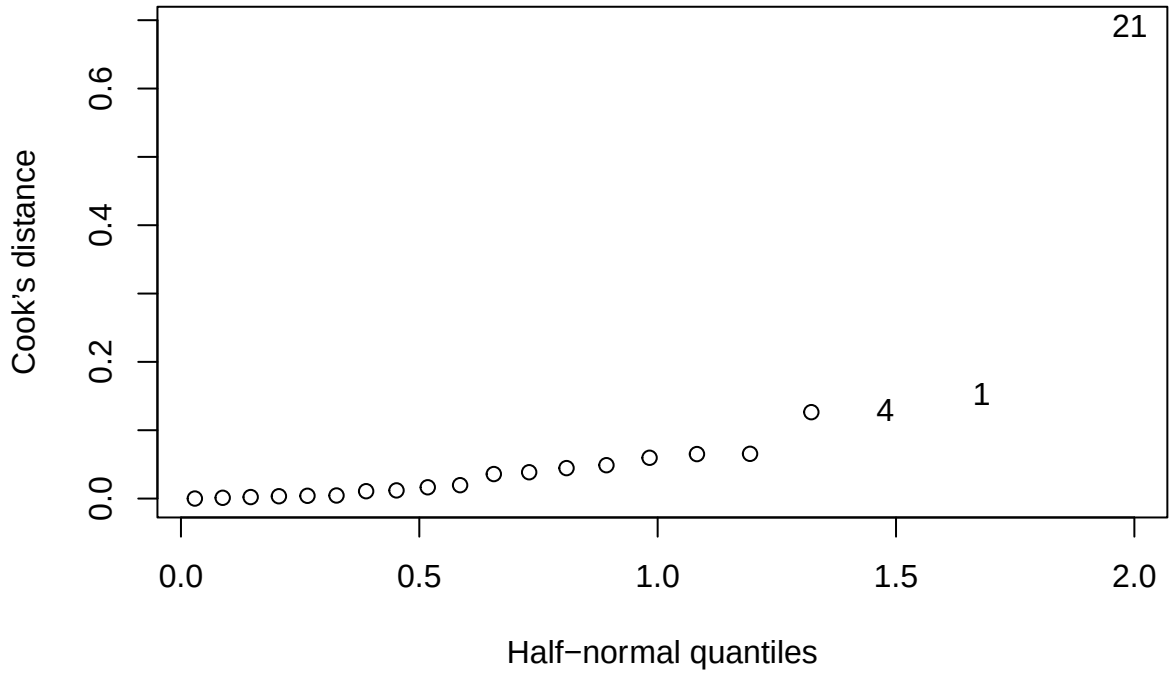
```
qt(0.05/(21*2), 16)
```

```
## [1] -3.603616
```

Nous constatons que 3.330493 est inférieure à 3.603616, ainsi l'observation "21" n'est pas une valeur abérrante. Par conséquent il n'y a pas de valeurs abérrantes.

- *Points influents*

```
cook2=cooks.distance(stacklossg1)
halfnorm(cook2, 3, labs=states, ylab="Cook's distance")
```



Nous constatons maintenant que les observations 1, 4 et 21 sont les plus élevées mais l'observation 21 reste maximale.

En supprimant cette observation, on obtient le modèle suivant:

```
stacklossg4=lm(stack.loss~Air.Flow+Water.Temp+Acid.Conc.,subset=(cook2<max(cook2)),data=stackloss)
stacklossgr4=summary(stacklossg4)
stacklossgr4
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
##     data = stackloss, subset = (cook2 < max(cook2)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0449 -2.0578  0.1025  1.0709  6.3017
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -43.7040     9.4916  -4.605 0.000293 ***
## Air.Flow      0.8891     0.1188   7.481 1.31e-06 ***
## Water.Temp    0.8166     0.3250   2.512 0.023088 *
## Acid.Conc.   -0.1071     0.1245  -0.860 0.402338
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.569 on 16 degrees of freedom
## Multiple R-squared:  0.9488, Adjusted R-squared:  0.9392
## F-statistic: 98.82 on 3 and 16 DF,  p-value: 1.541e-10
```

e) Répétons les processus de sélection des variables.

Backward selections:

D'après le résumé du modèle "stacklossg4", on constate que la variable "Acid.Conc." n'est pas significative (pour un seuil de risque de 5%). En éliminant cette variable non significative, nous retrouvons le modèle précédemment simplifié, c'est-à-dire le modèle "stacklossg3":

```
stacklossgr3
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow + Water.Temp, data = stackloss,
##     subset = (cook1 < max(cook1)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9052 -2.2893  0.5151  1.0123  6.2916
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -51.0760     4.0502 -12.611 4.69e-10 ***
## Air.Flow      0.8630     0.1140   7.568 7.70e-07 ***
## Water.Temp    0.8033     0.3222   2.493   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.549 on 17 degrees of freedom
```

```
## Multiple R-squared:  0.9464, Adjusted R-squared:  0.9401
## F-statistic: 150.2 on 2 and 17 DF,  p-value: 1.571e-11
```

D'après ce résumé, on constate que toutes les variables présentes sont significatives. Cette méthode aboutit donc au modèle contenant les variables "Air.Flow" et "Water.Temp".

An Information Criterion (AIC):

```
step(stacklossg4)
```

```
## Start:  AIC=41.28
## stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.
##
##              Df Sum of Sq    RSS    AIC
## - Acid.Conc.  1      4.89 110.50 40.185
## <none>                    105.61 41.281
## - Water.Temp  1     41.67 147.28 45.932
## - Air.Flow    1    369.42 475.04 69.353
##
## Step:  AIC=40.19
## stack.loss ~ Air.Flow + Water.Temp
##
##              Df Sum of Sq    RSS    AIC
## <none>                    110.50 40.185
## - Water.Temp  1     40.41 150.90 44.418
## - Air.Flow    1    372.32 482.82 67.678
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow + Water.Temp, data = stackloss,
##     subset = (cook2 < max(cook2)))
##
## Coefficients:
## (Intercept)     Air.Flow   Water.Temp
##    -51.0760       0.8630       0.8033
```

Cette méthode s'arrête également sur le modèle contenant les variables "Air.Flow" et "Water.Temp", puisque l'AIC de ce modèle est inférieur à celui du modèle contenant toutes les variables explicatives.

R2 adjusted:

```
stacklossgr4$adj.r.squared
```

```
## [1] 0.9391942
```

```
stacklossgr3$adj.r.squared
```

```
## [1] 0.9401238
```

Nous constatons que le R^2 ajusté du modèle sans la variable "Acid.Conc." est supérieur à celui du modèle contenant la variable "Acid.Conc.". Cela nous permet de choisir le modèle sans la variable "Acid.Conc.".

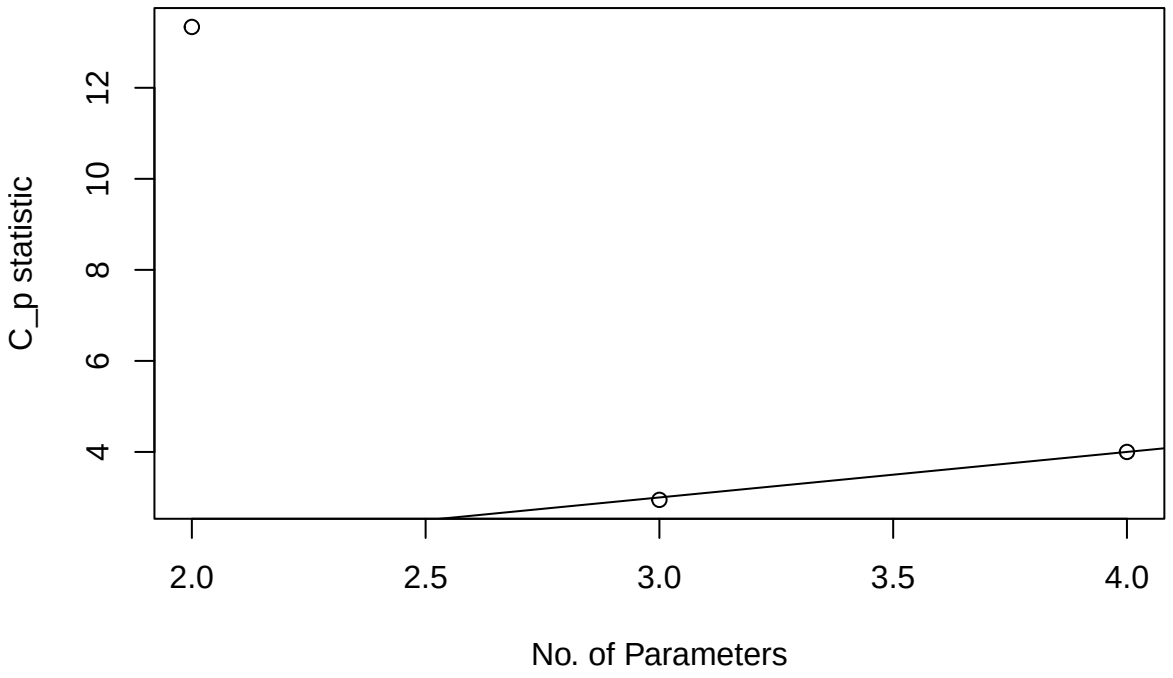
Mallow Cp:

```
library(leaps)
```

```
## Warning: le package 'leaps' a été compilé avec la version R 4.3.3
```

```
h=regsubsets(stack.loss~.,data=stackloss)

hr=summary(h)
```

```r
plot(2:4, hr$cp, xlab="No. of Parameters", ylab="C_p statistic")
abline(0,1)
```



À la lumière de ce graphique, les modèles comportant au moins deux paramètres semblent être les plus susceptibles d'être retenus.