Devoir 2. Statistical Learning

Louis Henri Franc

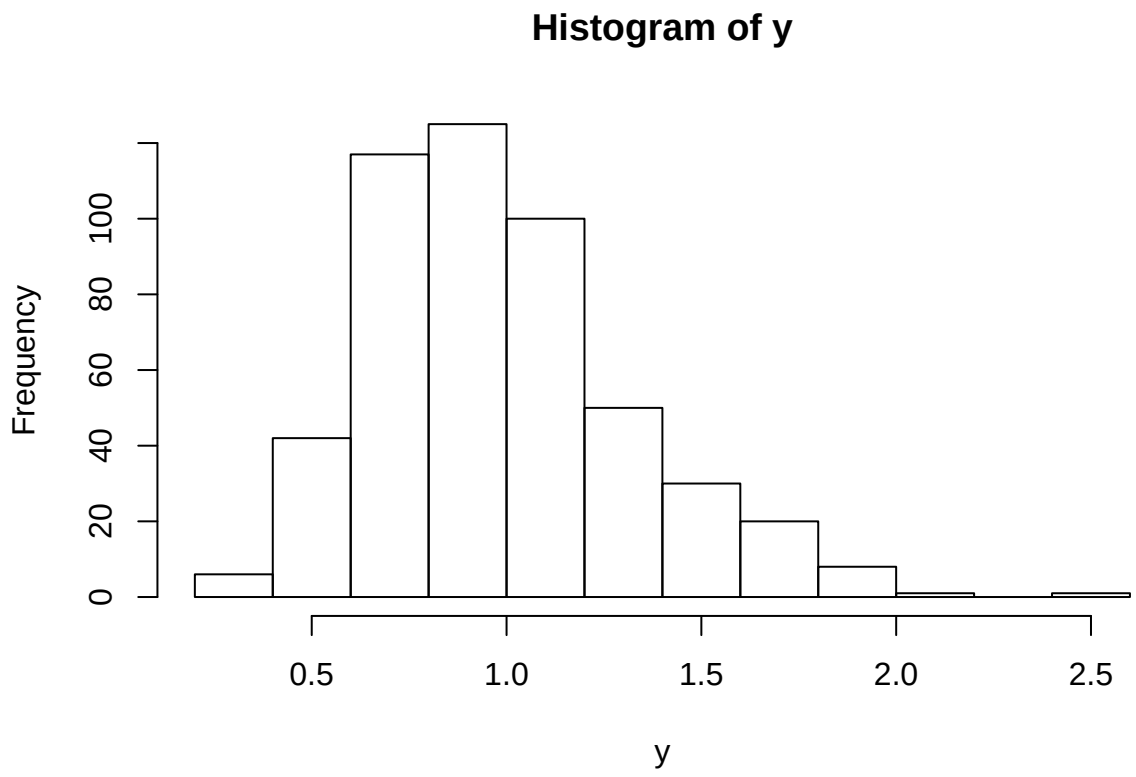
16 septembre 2016

Exercice 1

Question a

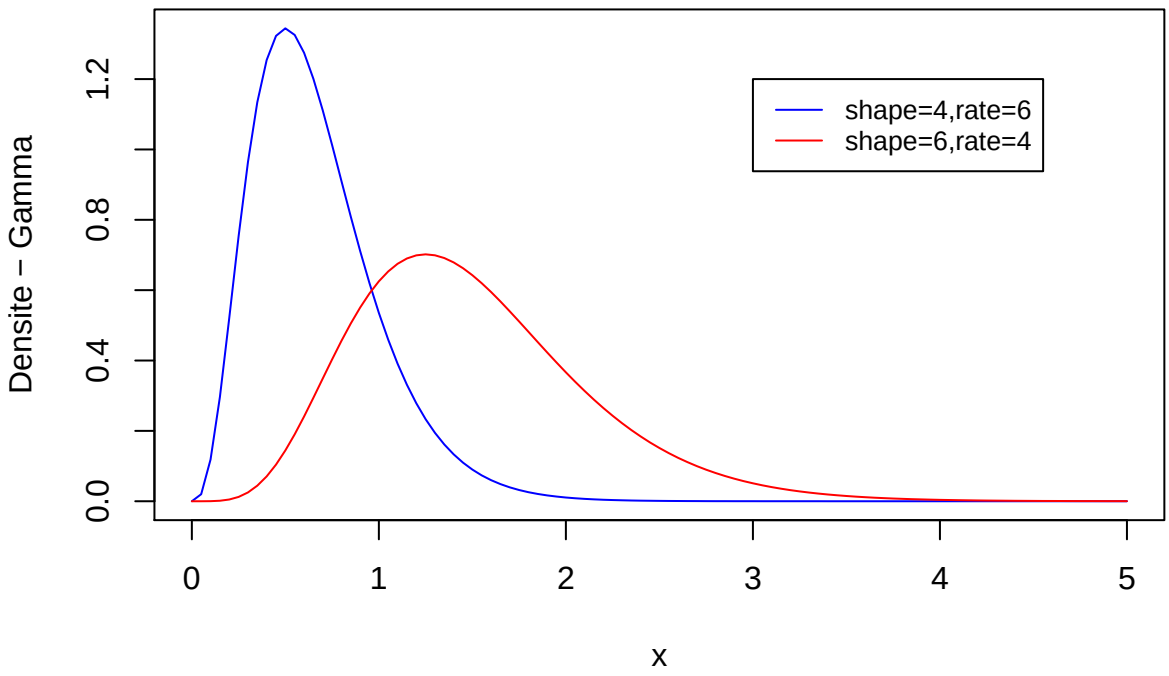
Pour la loi Gamma, nous allons prendre comme paramètre (10,5)

```
y = rgamma(500,10,10)
hist(y)
```



Question b

```
f0 = function(x){ dgamma(x,shape = 4,rate = 6) }
f1 = function(x){ dgamma(x,shape = 6,rate = 4) }
plot(f0,0,5,,y = seq(0,1,0.2),col='blue',ylab="Densite - Gamma")
plot(f1,0,5,y = seq(0,1,0.2),col='red',add=TRUE)
legend(3,1.2,legend=c("shape=4,rate=6","shape=6,rate=4"),col=c("blue","red"), lty=1, cex=0.8)
```



Exercice 2

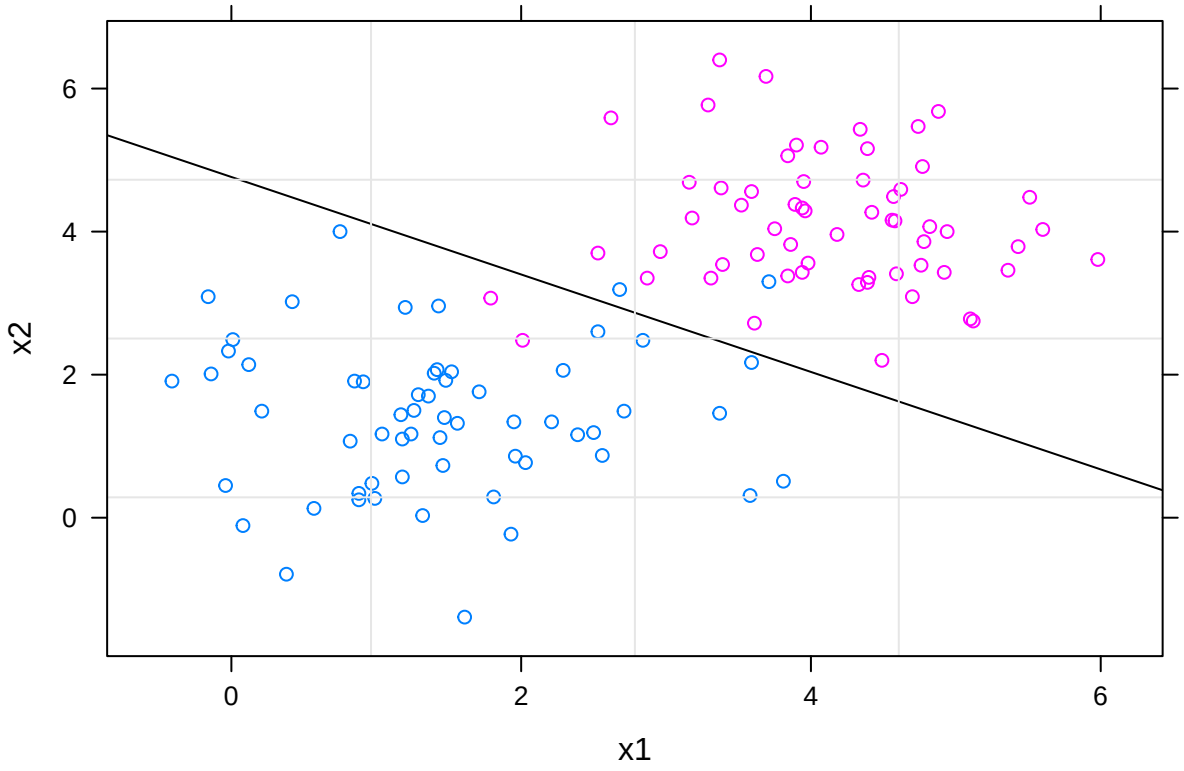
Question a

```
don<-read.csv2("Capteurs.csv")


# One Hot encoding

don$y= as.factor(don$y)
don$y = (don$y == "C")
train = don[0:99,]
test = don[100:120,]
mdl <- glm( y ~ x1 + x2 , data = train , family=binomial)

prediction = predict(mdl,newdata = as.data.frame(test[1:2]),type='response')
results <- ifelse(prediction > 0.5,1,0)
print(paste("Accuracy of the model is", 1 - mean(results != test$y)))
```

```
## [1] "Accuracy of the model is 1"
```

```
slope <- coef(mdl)[2]/(-coef(mdl)[3])
intercept <- coef(mdl)[1]/(-coef(mdl)[3])
xyplot( x2 ~ x1 , data = don, groups = y,
```

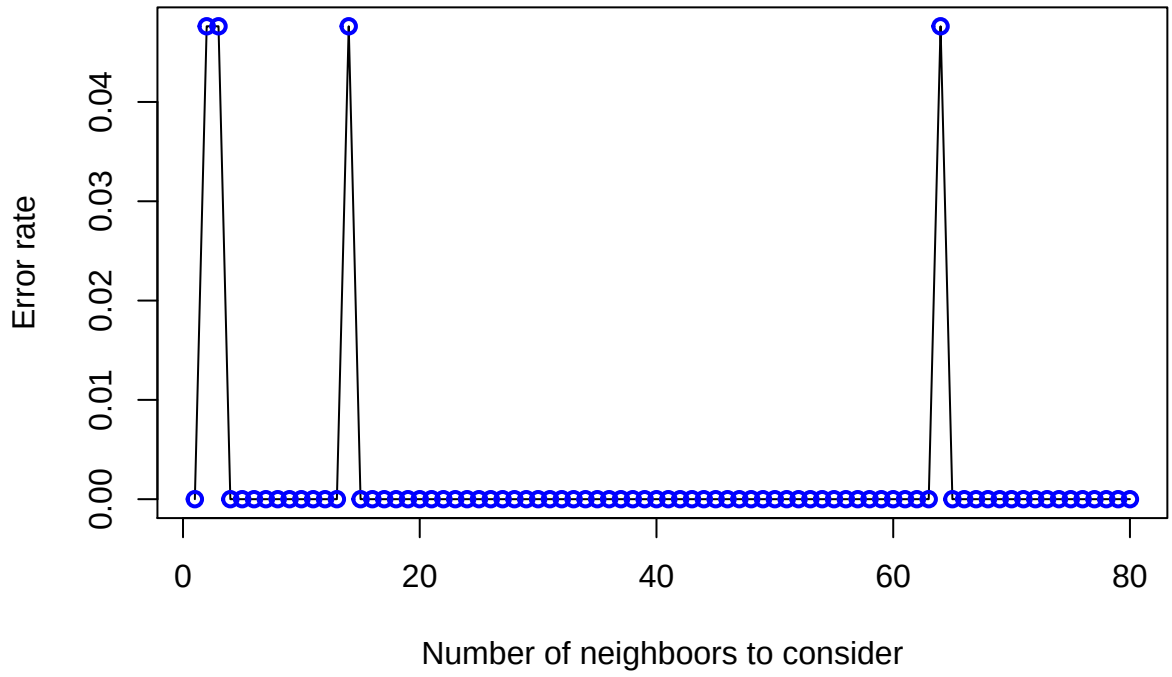
```
  panel=function(...){
      panel.xyplot(...)
      panel.abline(intercept , slope)
      panel.grid(...)
      })
```



Question b

```
# Standardising the data
don_scale = scale(don[,c(1,2)])
# Splitting the data
train = don_scale[0:99,]
test = don_scale[100:120,]

train_X_std = train[,c(1,2)]
test_X_std  = test[,c(1,2)]

train_Y = don[0:99,3]
test_Y  = don[100:120,3]

# Accuracy values
accuracy = vector(length=80)

for(clusterK in 1:80) {
```

```
  knn = knn(train=train_X_std,test= test_X_std,cl= train_Y, k = clusterK)
  tau = mean(knn != test_Y)
  accuracy[clusterK] = tau
}

plot(accuracy, xlab = "Number of neighboors to consider",
     ylab= "Error rate",type="l")
points(1:80,accuracy,lwd=2,col="blue")
```




```
print(which.min(accuracy))
```

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

Il aurait été intéressant d'utiliser la méthode de cross validation afin que le choix du k ne dépende pas d'un seul test. On obtient de nombreux k optimaux. Prendre k = 1 minimise le temps de calcul, mais sera sans doute très instable aux bruits des données. Prendre comme valeur 4 ou 5 semble être plus raisonnable. Dans la suite de l'exercice nous allons prendre 5 comme valeurs

Question c

```
# Prediction from Linear Regression
ifelse(predict(mdl,newdata = data.frame(x1 = 3.2,x2 = 3.5)) > 0.5,"C","D")
```

```
##   1
## "C"
```

```
# Prediction using KNN and optimal number of neighboors = 5
train_X_std = don_scale[,c(1,2)]
train_Y = don[,3]

test = c(3.2,3.5)
ifelse(knn(train=train_X_std,test= test,cl= train_Y, k = 5,prob = TRUE) == 1,"D","C")
```

```
## [1] "C"
```

Les deux méthodes indiquent que l'état de l'équipement est probablement correct. Si l'on souhaite prédire uniquement une valeur, la méthode des plus proches voisins est moins couteuse en calcul. Cependant si l'on estime que l'on peut séparer nos données par une droite, et que les résidus de notre modèle sont faibles, alors la regression linéaire donnera une valeur correcte à un intervalle de confiance élevée.

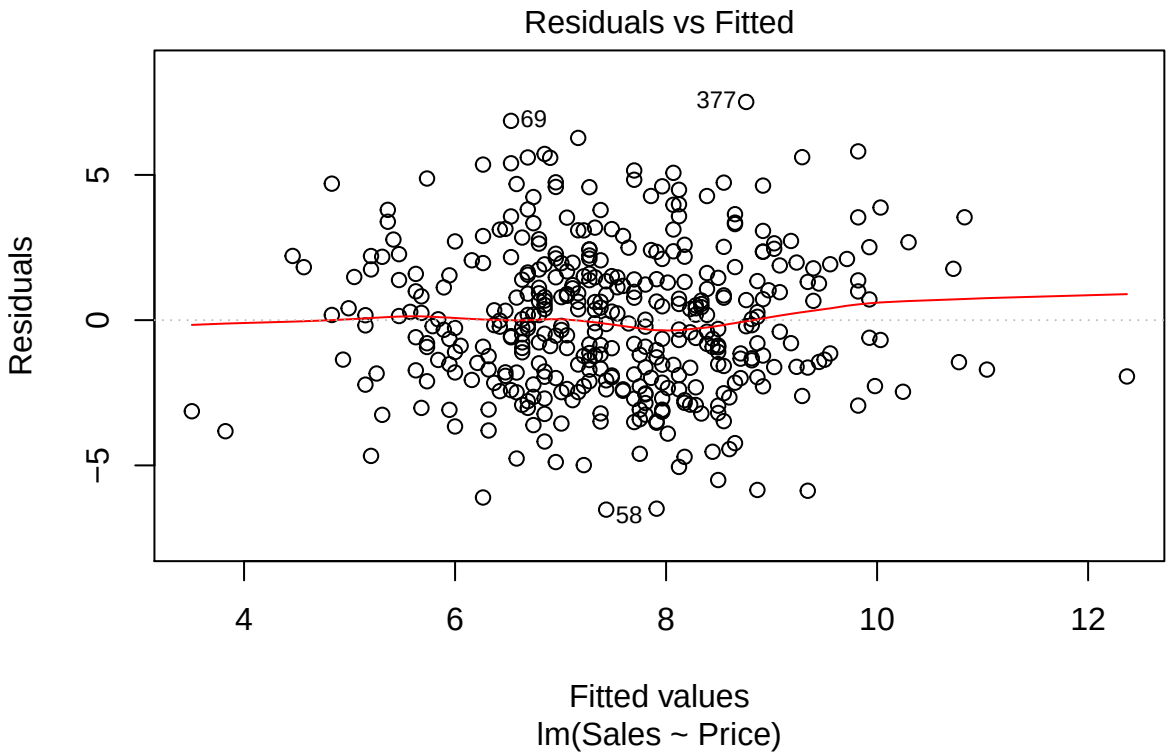
Exercice 3

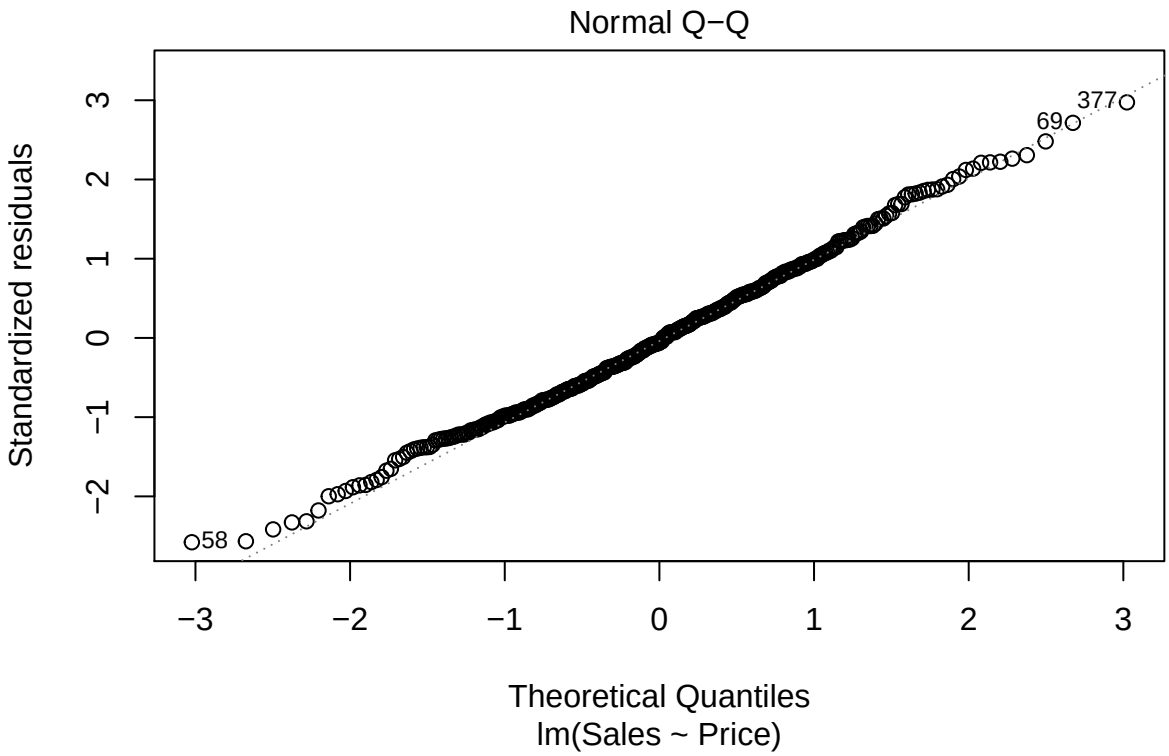
Question a

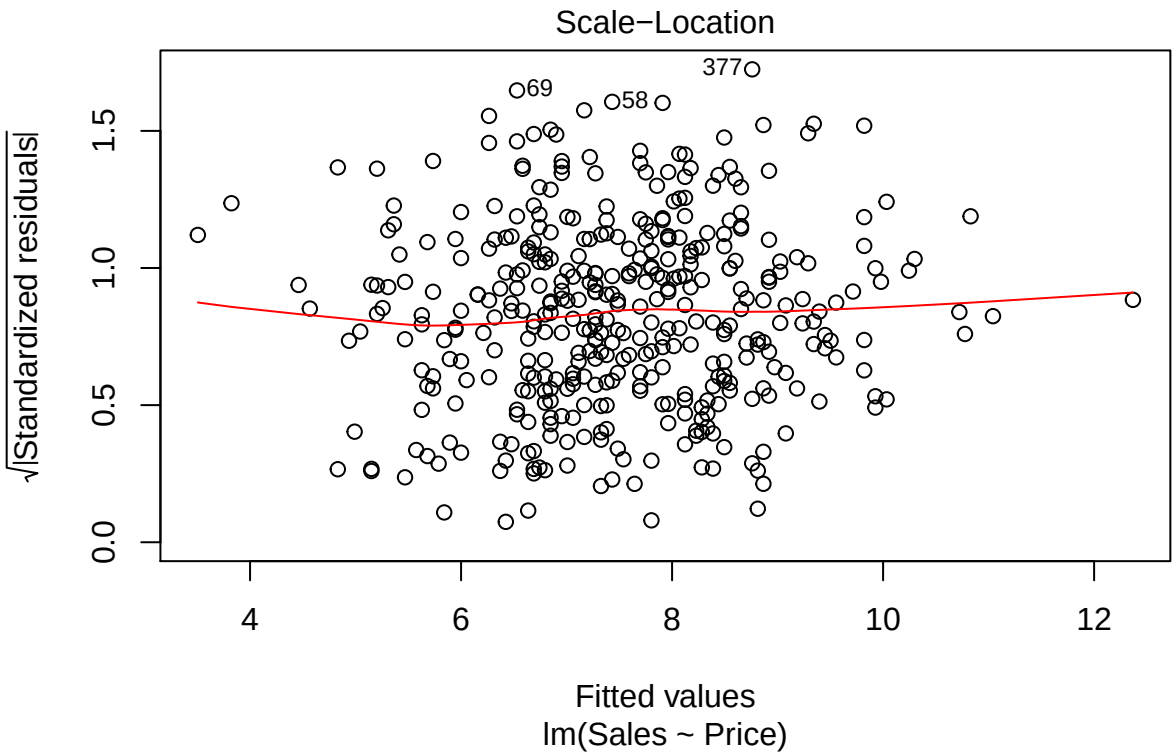
```
fitmodel = lm(Sales~Price,data=Carseats)
summary(fitmodel)
```

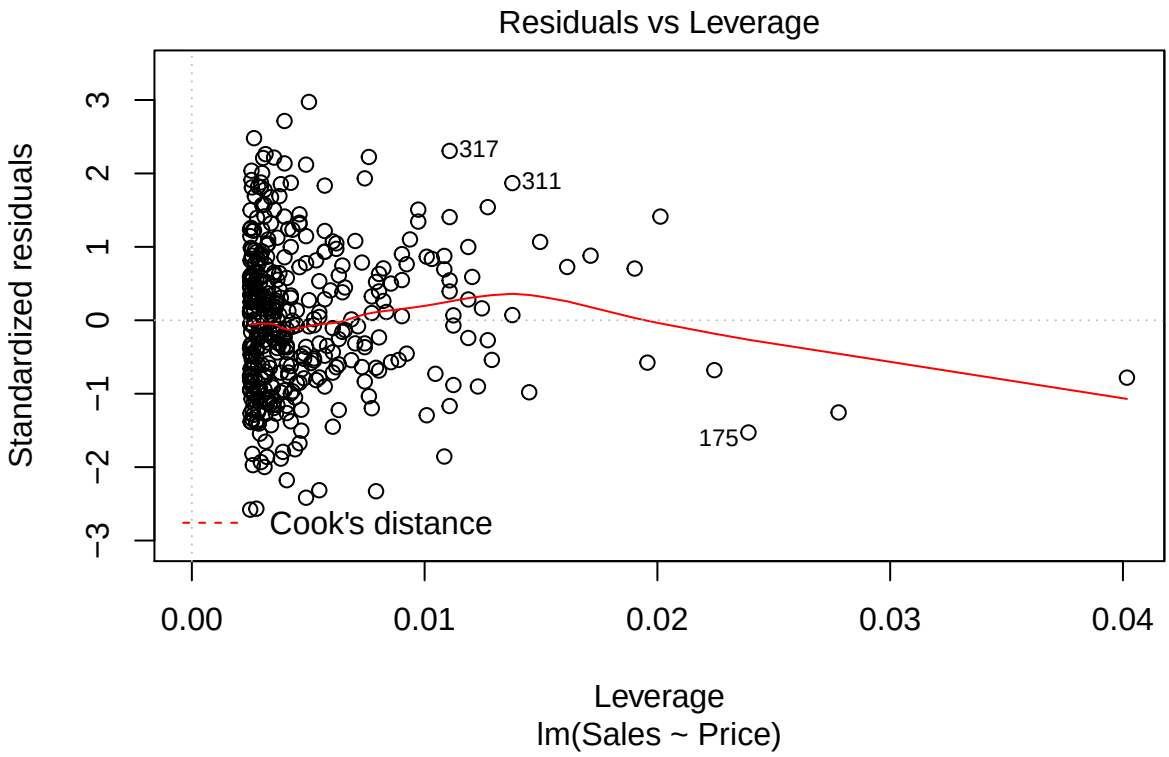
```
##
## Call:
## lm(formula = Sales ~ Price, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5224 -1.8442 -0.1459  1.6503  7.5108
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.641915   0.632812  21.558   <2e-16 ***
## Price       -0.053073   0.005354  -9.912   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.532 on 398 degrees of freedom
## Multiple R-squared:  0.198,  Adjusted R-squared:  0.196
## F-statistic: 98.25 on 1 and 398 DF,  p-value: < 2.2e-16
```

```
plot(fitmodel)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ Price)
377
69
58
5
0
−5
4
6
8
10
12


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ Price)
58
69
377
−3
−2
−1
0
1
2
3


Scale−Location
377
69
58
√|Standardized residuals|
0.0
0.5
1.0
1.5
4
6
8
10
12
Fitted values
lm(Sales ~ Price)



Analyse des graphes diagnostiques des résidus

- Le coefficient entre X1 et Y est négatif -0.05. On remarque cependant qu'une grande variation des X1 n'affectent pas à la même échelle une variation des Y. L'erreur type est petite comparé aux valeurs des Y, donc la régression approche bien les valeurs de vente. Enfin comme la p_value pour le price est très petite, on peut rejeter l'hypothèse nulle et considérer qu'il y a un lien linéaire entre X et Y.
- Le modèle linéaire n'explique pas bien la variation des Y. Lorsque l'on regarde la statistique R^2, on remarque que celle ci est plus proche de 0 que de 1. Il y a donc une grande partie de la variabilité des valeurs de Y qui n'est pas expliqué par ce modèle linéaire.

```
# Interval de prédiction
predict(fitmodel,interval="prediction",newdata = data.frame(Price=100))
```

```
##        fit      lwr      upr
## 1 8.334613 3.347215 13.32201
```

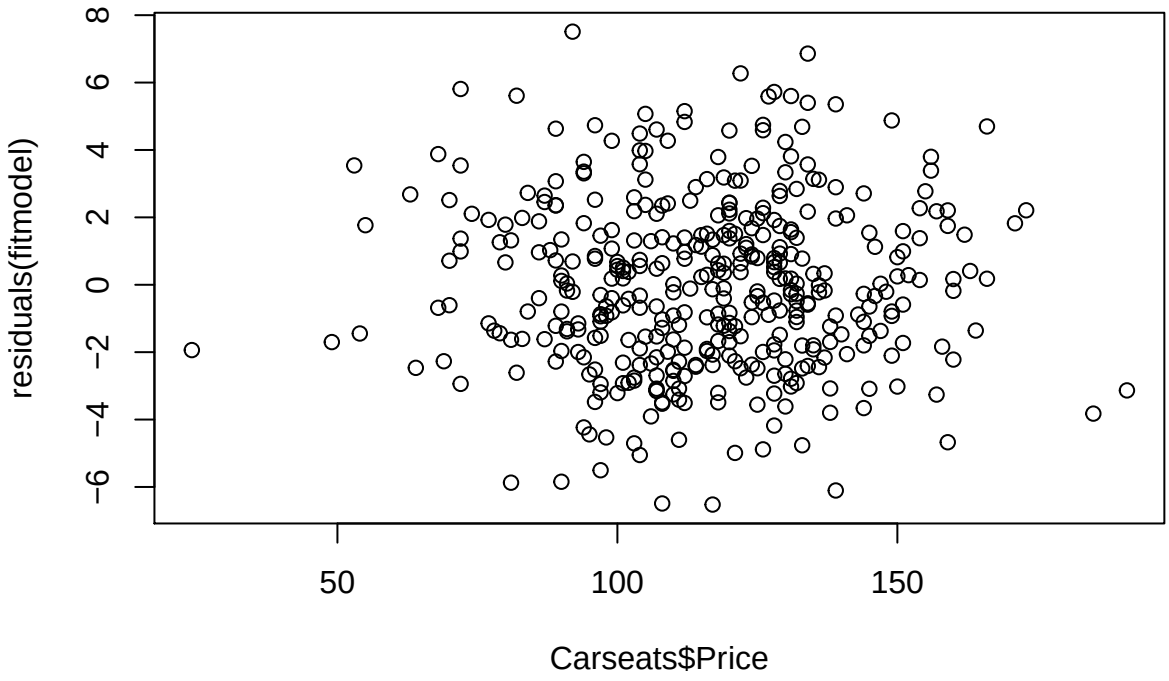
```
# Interval de confiance
predict(fitmodel,interval="confidence",newdata = data.frame(Price=100))
```

```
##        fit      lwr      upr
## 1 8.334613 8.035271 8.633955
```

1. Le premier graphe Residual vs Fitted indique que les résidus ne semblent pas varier suivant la variable dépendante.

2. Le second graphe indique que l'approximation linéaire semble correcte, et que les résidus suivent une loi normale.
3. Le troisième graphe montre que les résidus sont répartis également le long des valeurs prédites. Cependant on aimerait savoir si cela est le cas pour les inputs

```
plot(Carseats$Price,residuals(fitmodel))
```



Les points leviers sont les points 317, 311,et 175. Analysons ces points un par un:

```
price1 = Carseats$Price[-c(317)]
y_1 = Carseats$Sales[-c(317)]
summary(lm(y_1 ~ price1))
```

```
##
## Call:
## lm(formula = y_1 ~ price1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5091 -1.8333 -0.1291  1.6471  7.5529
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.494042   0.632579  21.332   <2e-16 ***
## price1      -0.051923   0.005348  -9.708   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.519 on 397 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.1898
## F-statistic: 94.25 on 1 and 397 DF,  p-value: < 2.2e-16
```

```
price1 = Carseats$Price[-c(311)]
y_1 = Carseats$Sales[-c(311)]
summary(lm(y_1 ~ price1))
```

```
##
## Call:
## lm(formula = y_1 ~ price1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5092 -1.8196 -0.1211  1.6552  7.4973
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.753822   0.633644   21.71   <2e-16 ***
## price1      -0.054142   0.005368  -10.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.524 on 397 degrees of freedom
## Multiple R-squared:  0.204,  Adjusted R-squared:  0.202
## F-statistic: 101.7 on 1 and 397 DF,  p-value: < 2.2e-16
```

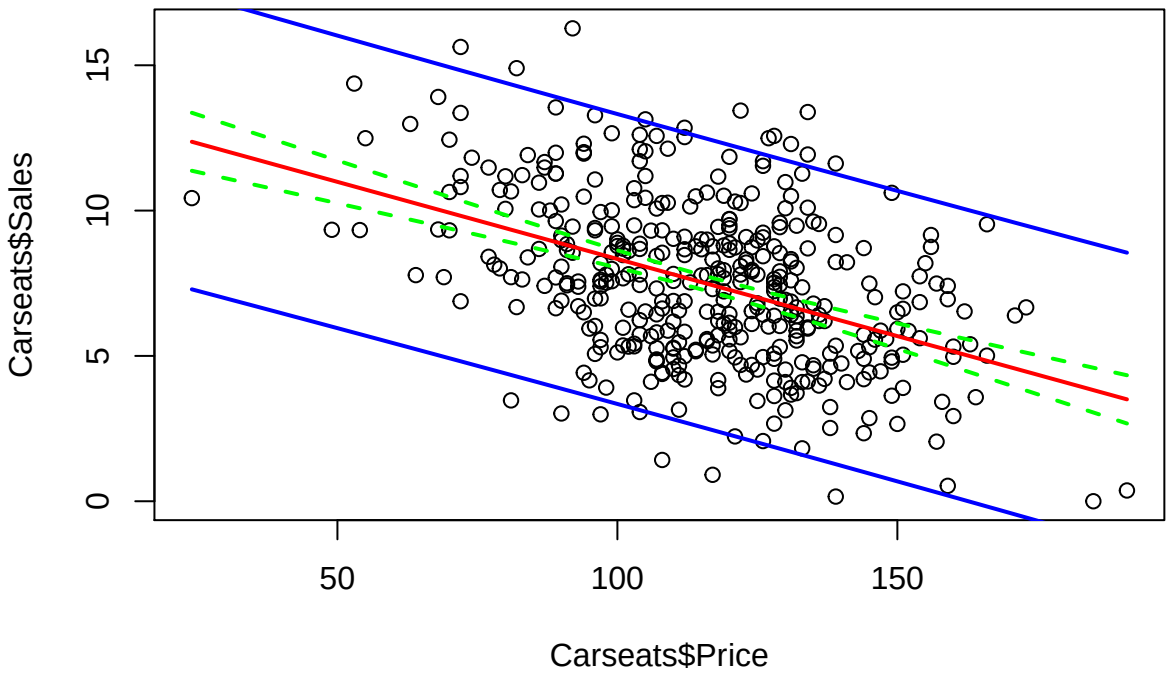
```
price1 = Carseats$Price[-c(175)]
y_1 = Carseats$Sales[-c(175)]
summary(lm(y_1 ~ price1))
```

```
##
## Call:
## lm(formula = y_1 ~ price1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5336 -1.8400 -0.1436  1.6461  7.5298
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.511370   0.637476  21.195   <2e-16 ***
## price1      -0.051861   0.005404  -9.597   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.528 on 397 degrees of freedom
## Multiple R-squared:  0.1883, Adjusted R-squared:  0.1863
## F-statistic: 92.11 on 1 and 397 DF,  p-value: < 2.2e-16
```

On remarque une amélioration négligeable de la performance de la régression en comparant les RSE, mais il ne semble pas que ces points influent énormément sur la qualité de la régression. En conclusion, nos hypothèses sur les résidus dans notre modèle de départ semblent être corrects.

Question b

```
NewPrice = data.frame(Price=c(seq(24,191,(191-24)/100)))
pred.int = predict(fitmodel,interval="prediction",newdata = NewPrice)
conf.int = predict(fitmodel,interval="confidence",newdata = NewPrice)
values = pred.int[,1]
pred.lower = pred.int[,2]
pred.upper = pred.int[,3]
conf.lower = conf.int[,2]
conf.upper = conf.int[,3]

# Nuage de points
plot(Carseats$Price,Carseats$Sales)

# Droite de régression
lines(NewPrice$Price,values,col="red",lwd=2)

# Intervalle de prévision
lines(NewPrice$Price,pred.lower,col="blue",lwd=2,lty=1)
lines(NewPrice$Price,pred.upper,col="blue",lwd=2,lty=1)

# Intervalle de confiance
lines(NewPrice$Price,conf.upper,col="green",lwd=2,lty=2)
lines(NewPrice$Price,conf.lower,col="green",lwd=2,lty=2)
```



Question c

L'erreur standart des résidus est assez petite (si l'on regarde l'ordre de grandeur des données. Par ailleurs la droite de régression approxime bien les données Y. La R squared ne nous informe si le modèle est correct ou si il faut utiliser un ordre polynomial supérieur. Elle indique que la variation des X n'explique pas la variation des Y, donc il semble qu'utiliser un modèle d'ordre supérieur avec les même inputs n'améliorera pas la qualité de l'approximation.

```
summary(lm(Carseats$Sales ~ poly(Carseats$Price, 2)))
```

```
##
## Call:
## lm(formula = Carseats$Sales ~ poly(Carseats$Price, 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4945 -1.8714 -0.1248  1.6428  7.5140
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.4963     0.1268  59.140   <2e-16 ***
## poly(Carseats$Price, 2)1 -25.1004     2.5351  -9.901   <2e-16 ***
## poly(Carseats$Price, 2)2   0.8757     2.5351   0.345     0.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.535 on 397 degrees of freedom
## Multiple R-squared:  0.1982, Adjusted R-squared:  0.1942
## F-statistic: 49.07 on 2 and 397 DF,  p-value: < 2.2e-16
```

On observe aucune amélioration dans l'affinement de la régression. Le modèle linéaire est suffisant si l'on n'utilise que les données de Prix, cependant il semblerait qu'il faille utiliser d'autres variables pour expliquer la variable dépendante Vente.

Question d

La pvalue du Test de Fisher étant inférieure à 0.05, on accepte l'hypothèse nulle selon laquelle le modèle à trois variable incluant la variable quantitative US, n'est pas meilleur que le modèle à une variable indépendante.

Question e

```
fit1 = lm(Sales ~ CompPrice + Income + Advertising + Population + Price + Age + Education,data=Carseats)
summary(fit1)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Population +
##     Price + Age + Education, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0598 -1.3515 -0.1739  1.1331  4.8304
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.7076934  1.1176260   6.896 2.15e-11 ***
## CompPrice    0.0939149  0.0078395  11.980  < 2e-16 ***
## Income       0.0128717  0.0034757   3.703 0.000243 ***
## Advertising  0.1308637  0.0151219   8.654  < 2e-16 ***
## Population  -0.0001239  0.0006877  -0.180 0.857092
## Price       -0.0925226  0.0050521 -18.314  < 2e-16 ***
## Age         -0.0449743  0.0060083  -7.485 4.75e-13 ***
## Education   -0.0399844  0.0371257  -1.077 0.282142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.929 on 392 degrees of freedom
## Multiple R-squared:  0.5417, Adjusted R-squared:  0.5335
## F-statistic: 66.18 on 7 and 392 DF,  p-value: < 2.2e-16
```

Pour la sélection des variables, la selection à l'envers sera utilisé. Puisque Population a la plus grande p-value, celle ci va être éliminée de notre régression.

```
fit1 = lm(Sales ~ CompPrice+ Income +Advertising + Price+ Age+ Education,data=Carseats)
summary(fit1)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
##     Age + Education, data = Carseats)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.055 -1.360 -0.170  1.124  4.829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.651797   1.072393   7.135 4.69e-12 ***
## CompPrice    0.094053   0.007792  12.070  < 2e-16 ***
## Income       0.012895   0.003469   3.717 0.000231 ***
## Advertising  0.130150   0.014576   8.929  < 2e-16 ***
## Price       -0.092552   0.005043 -18.352  < 2e-16 ***
## Age         -0.044919   0.005993  -7.495 4.43e-13 ***
## Education   -0.039309   0.036891  -1.066 0.287278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.927 on 393 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.5346
## F-statistic:  77.4 on 6 and 393 DF,  p-value: < 2.2e-16
```

Puis on élimine la variable Education.

```
fit1 = lm(Sales ~ CompPrice+ Income +Advertising + Price+ Age+ Education,data=Carseats)
summary(fit1)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
##     Age + Education, data = Carseats)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.055 -1.360 -0.170  1.124  4.829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.651797   1.072393   7.135 4.69e-12 ***
## CompPrice    0.094053   0.007792  12.070  < 2e-16 ***
## Income       0.012895   0.003469   3.717 0.000231 ***
## Advertising  0.130150   0.014576   8.929  < 2e-16 ***
## Price       -0.092552   0.005043 -18.352  < 2e-16 ***
## Age         -0.044919   0.005993  -7.495 4.43e-13 ***
## Education   -0.039309   0.036891  -1.066 0.287278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.927 on 393 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.5346
## F-statistic:  77.4 on 6 and 393 DF,  p-value: < 2.2e-16
```

Aucune p-value pour la T statistique n'est inférieur 5%,

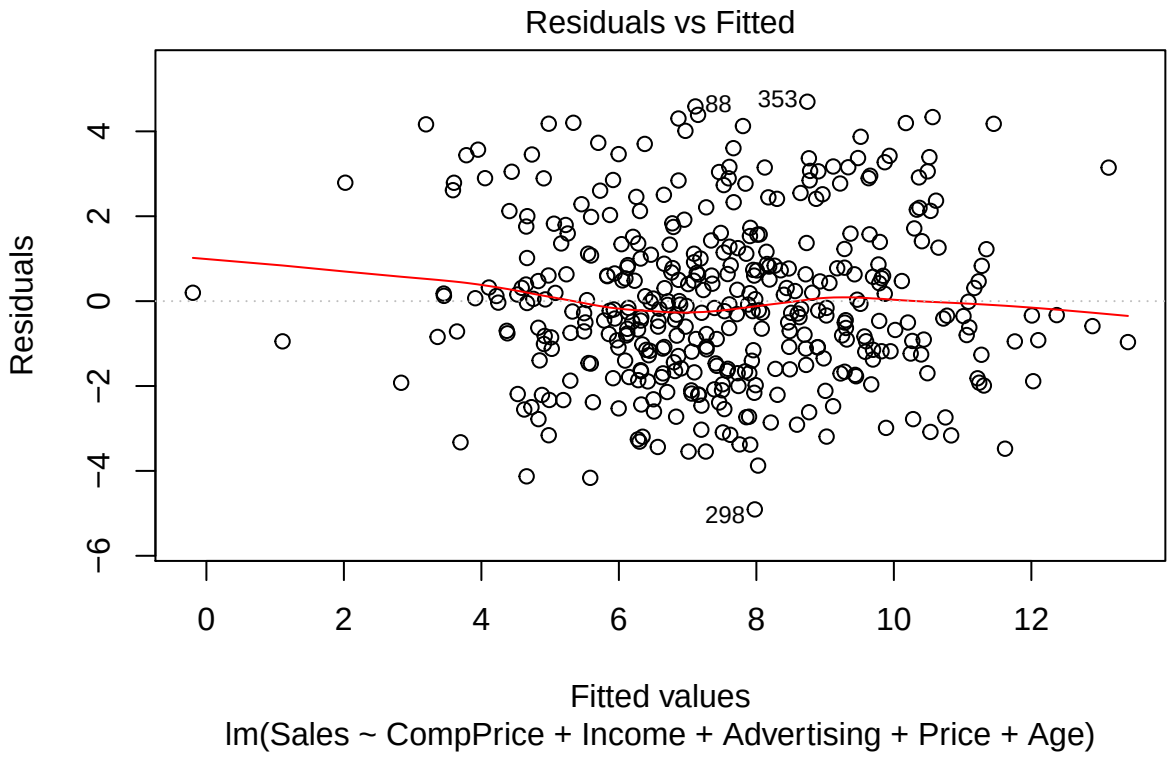
En utilisant la fonction stepAIC, on obtient le même résultat.

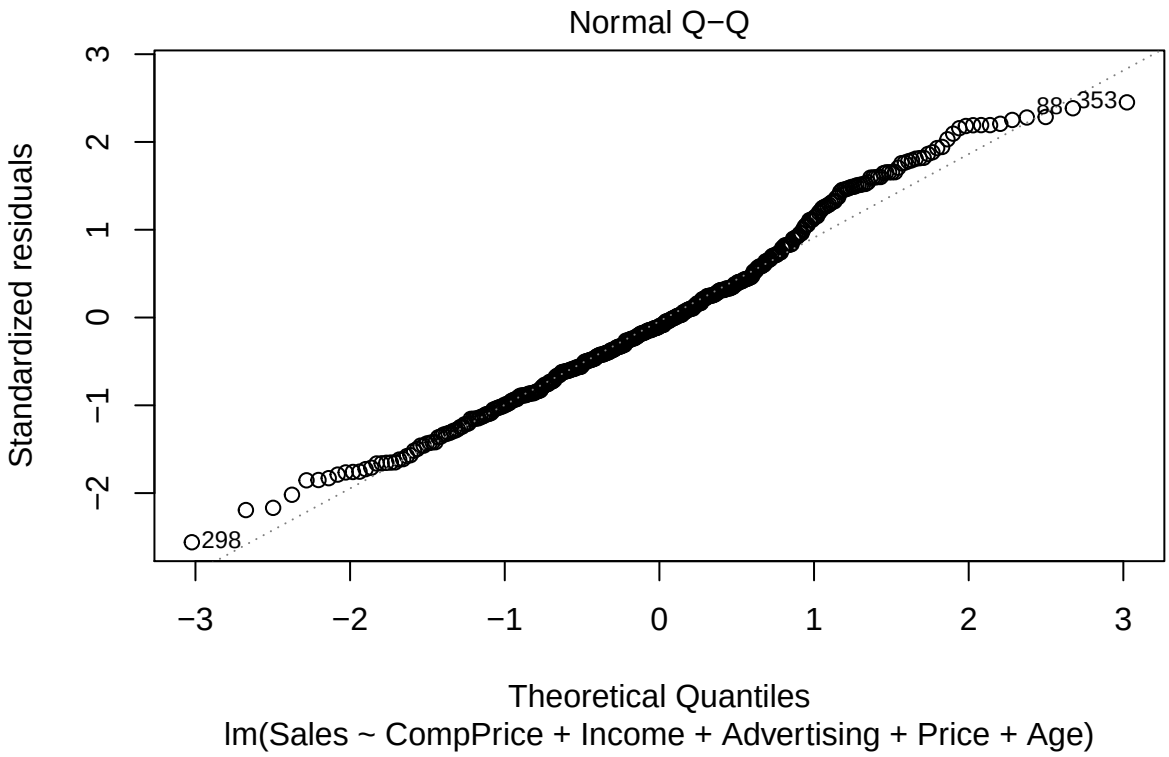
```
fit1 = lm(Sales ~ CompPrice + Income + Advertising + Population + Price + Age + Education,data=Carseats)
step <- stepAIC(fit1, direction="backward")
```

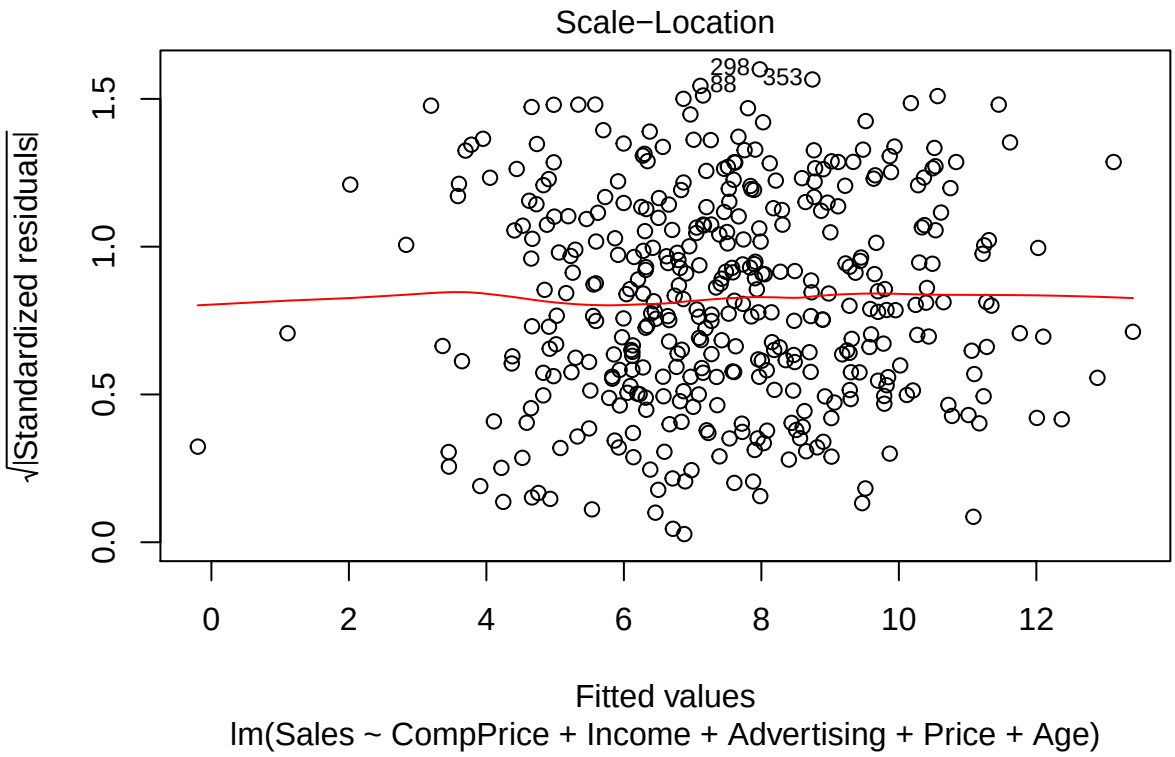
```
## Start:  AIC=533.5
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     Age + Education
##
##               Df Sum of Sq    RSS    AIC
## - Population   1      0.12 1458.7 531.53
## - Education    1      4.32 1462.9 532.68
## <none>                     1458.6 533.50
## - Income       1     51.03 1509.6 545.25
## - Age          1    208.48 1667.0 584.94
## - Advertising  1    278.65 1737.2 601.43
## - CompPrice    1    533.98 1992.5 656.28
## - Price        1   1247.94 2706.5 778.78
##
## Step:  AIC=531.53
## Sales ~ CompPrice + Income + Advertising + Price + Age + Education
##
##               Df Sum of Sq    RSS    AIC
## - Education    1      4.21 1462.9 530.68
## <none>                     1458.7 531.53
## - Income       1     51.29 1510.0 543.35
## - Age          1    208.51 1667.2 582.97
## - Advertising  1    295.91 1754.6 603.41
## - CompPrice    1    540.75 1999.4 655.66
## - Price        1   1250.06 2708.7 777.11
##
## Step:  AIC=530.68
## Sales ~ CompPrice + Income + Advertising + Price + Age
##
##               Df Sum of Sq    RSS    AIC
## <none>                     1462.9 530.68
## - Income       1     53.02 1515.9 542.92
## - Age          1    209.00 1671.9 582.10
## - Advertising  1    298.27 1761.2 602.91
## - CompPrice    1    539.21 2002.1 654.20
## - Price        1   1249.81 2712.7 775.70
```

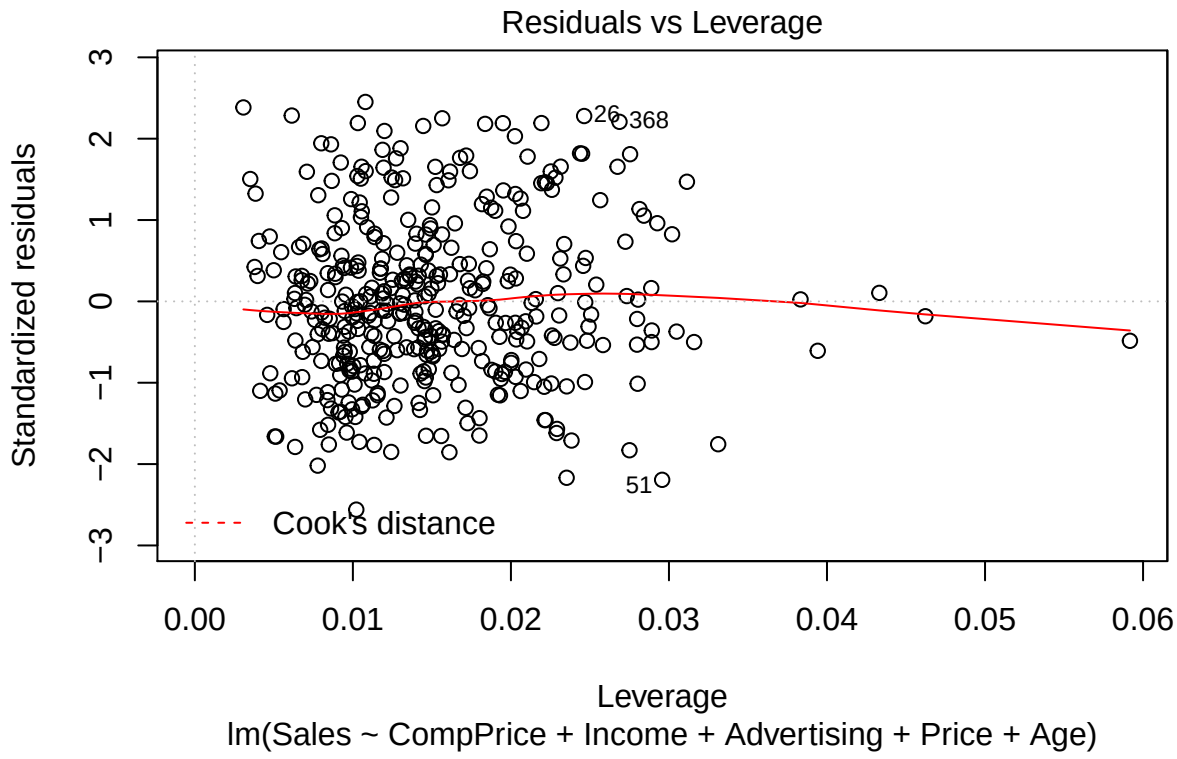
Nous allons nous réduire au modèle trouvé réduit à 5 variables.

```
plot(step)
```

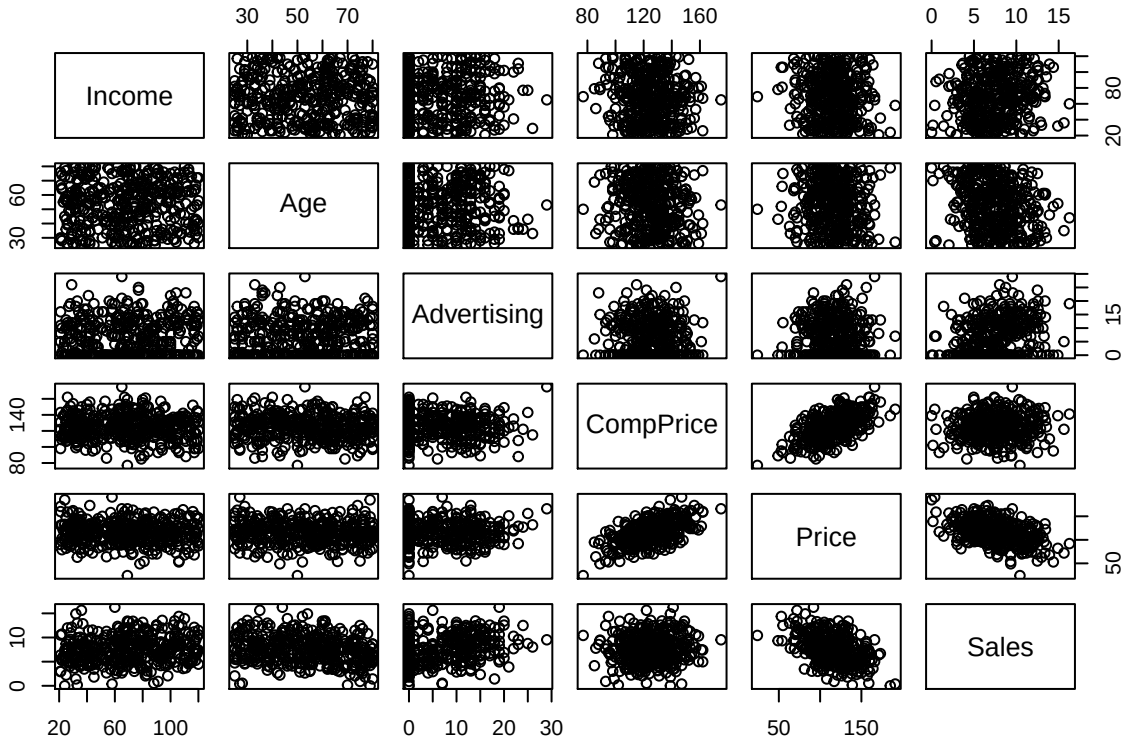

Residuals vs Fitted
88
353
298
Residuals
4
2
0
−2
−4
−6
0
2
4
6
8
10
12
Fitted values
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)
298
88
353
−3
−2
−1
0
1
2
3


Scale−Location
298
88
353
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
2
4
6
8
10
12
Fitted values
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)



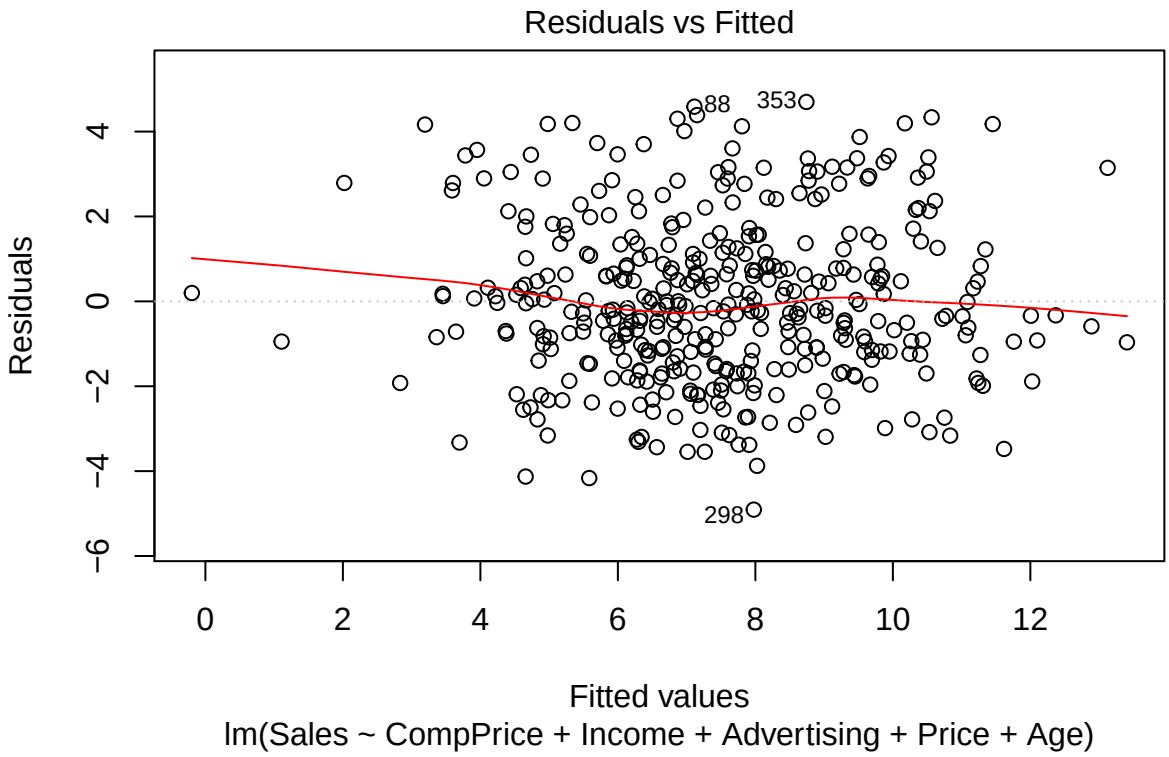

```
pairs(Carseats[,c("Income","Age","Advertising","CompPrice","Price","Sales")])
```

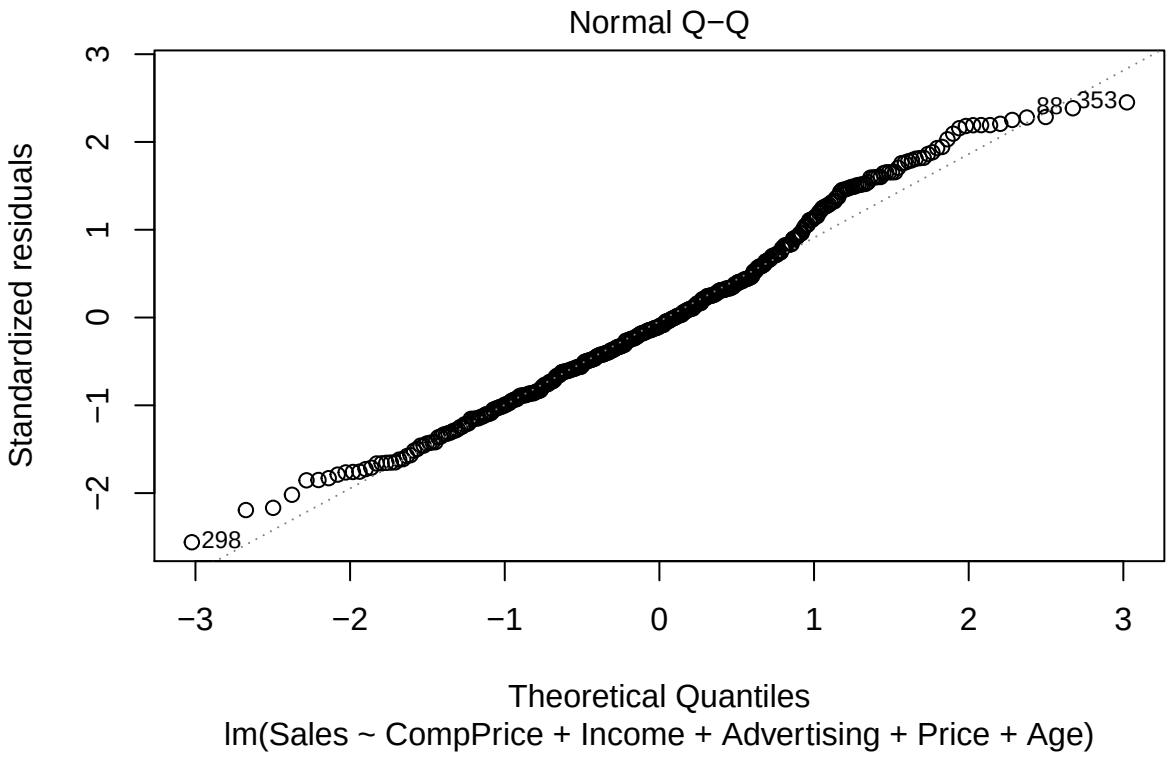


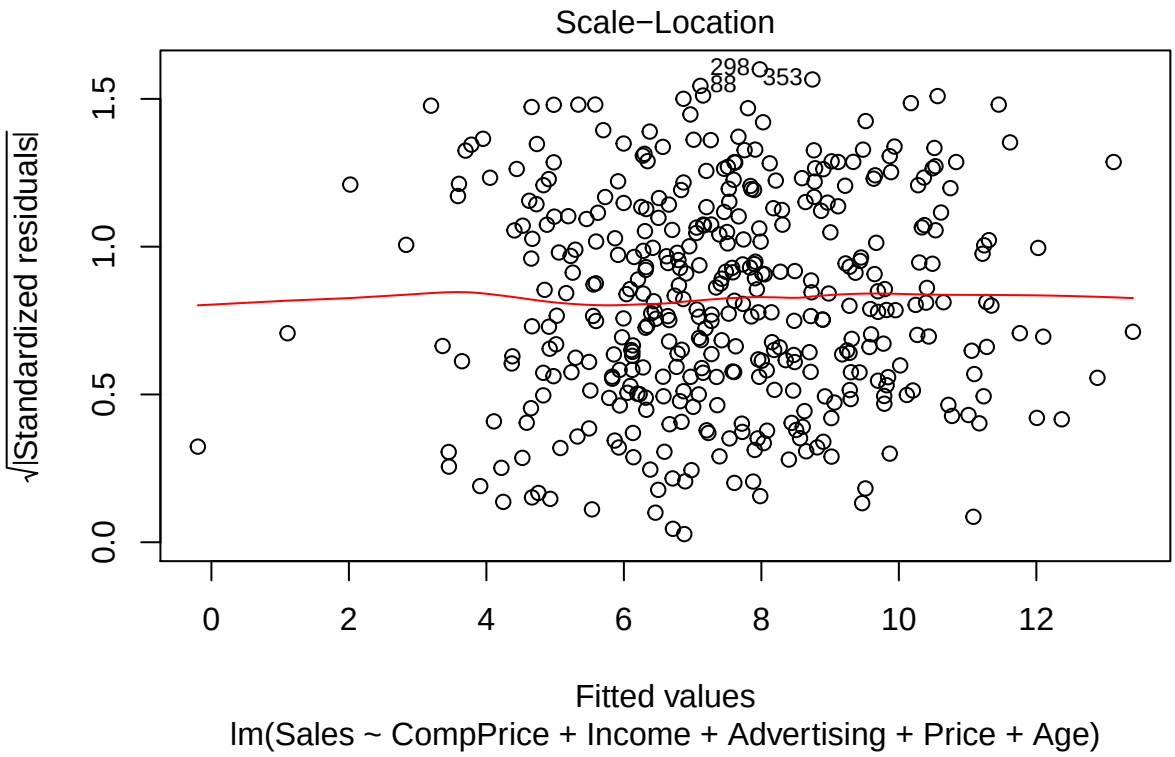
D'après la matrice de corrélation, il semblerait y avoir une corrélation entre la variable CompPrice et Price. Cependant cette corrélation n'est que de 0.5. De plus lorsque l'on retire ces variables du modèle linéaire, la qualité de la régression est impactée (diminution du R squared, augmentation des RSE). Il faut donc garder CompPrice et Price dans notre modèle.

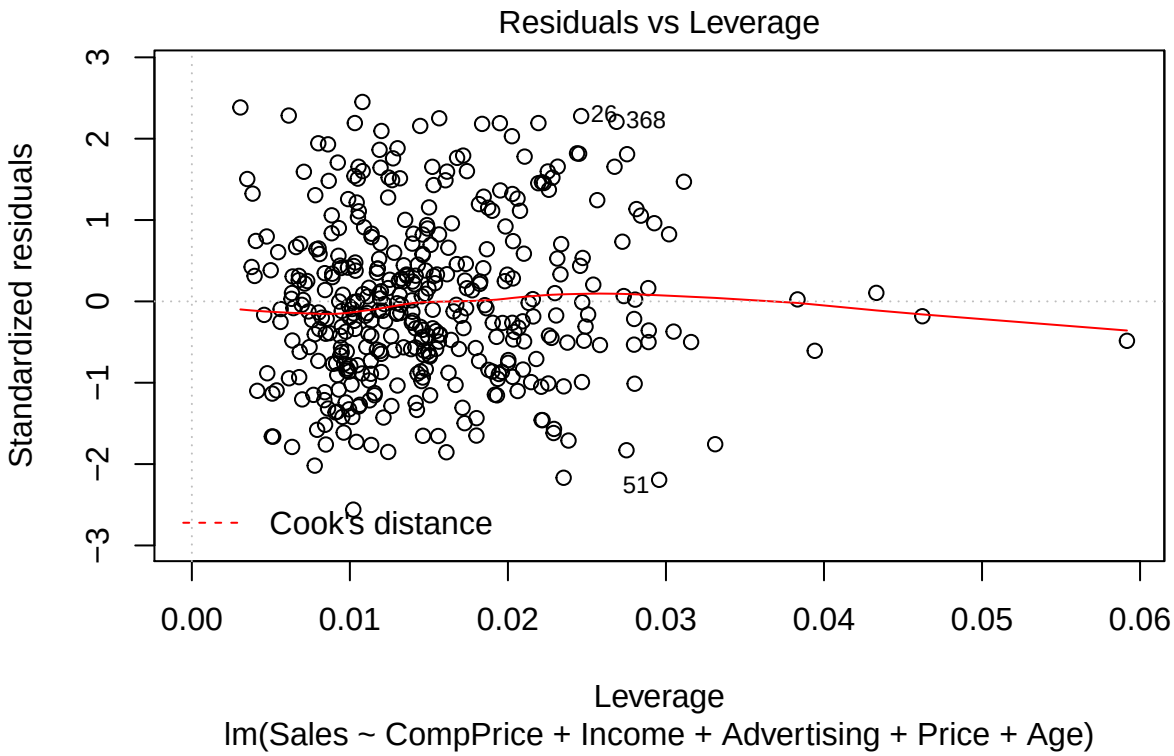
Dans la suite, nous allons décider de supprimer la variable Income car sa t_value est petite. Nous allons garder le modèle à quatre variable explicative.

```
fit1 = lm(Sales ~ CompPrice + Income + Advertising + Price + Age,data=Carseats)
plot(fit1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)
88
353
298
−6
−4
−2
0
2
4
0
2
4
6
8
10
12


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)
298
88
353
−3
−2
−1
0
1
2
3


Scale−Location
298
88
353
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
2
4
6
8
10
12
Fitted values
lm(Sales ~ CompPrice + Income + Advertising + Price + Age)



On remarque que le graphe Normal QQ est légerement courbé aux extrémités. Cela indique que les données ont plus de valeurs extrêmes par rapport à notre modèle gaussien initial.
Les points influents observés aux dernier graphe ne semblent pas être très extrêmes car la somme des résidus ne varient pas lorsque ces points sont retirés.

Pour conclure, on remarque que les modèles définies n'expliquent pas très bien les variations des ventes. Une grande partie de la variance n'est pas expliqué par le modèle (R squared ne dépasse jamais 0.6).