TP2 : Régression Linéaire

DJEBALI Wissam

1 mars 2018

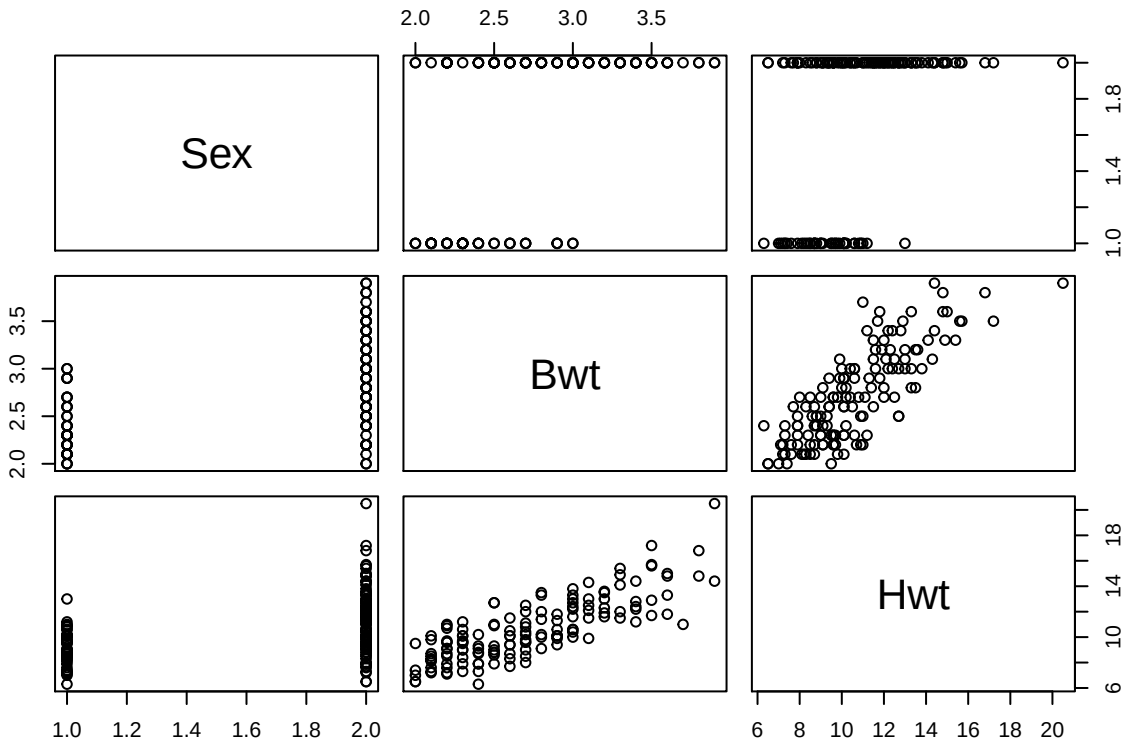
Linear Regression

Packages R : FactoMineR, factoextra, corrplot, stats, MASS

Simple Linear Regression with prevision

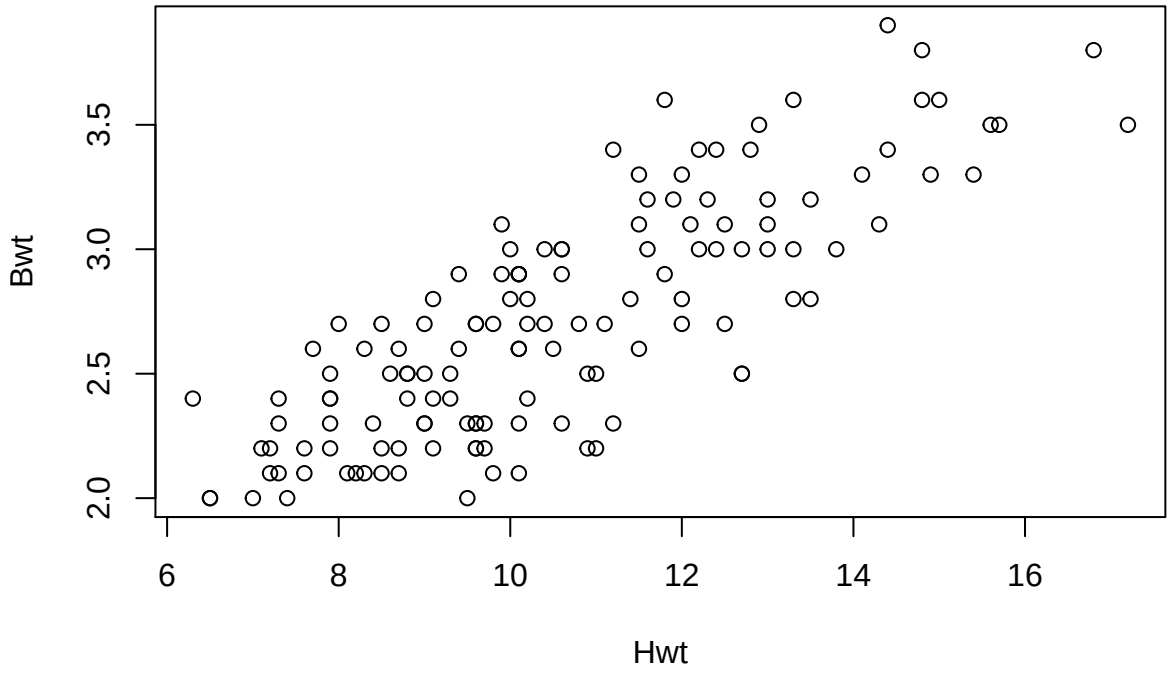
La régression lineaire simple est une généralisation du test de corrélation et du test t

```
data(cats)
# Pour avoir une idee des possible correlation entre les variables
plot(cats)
```

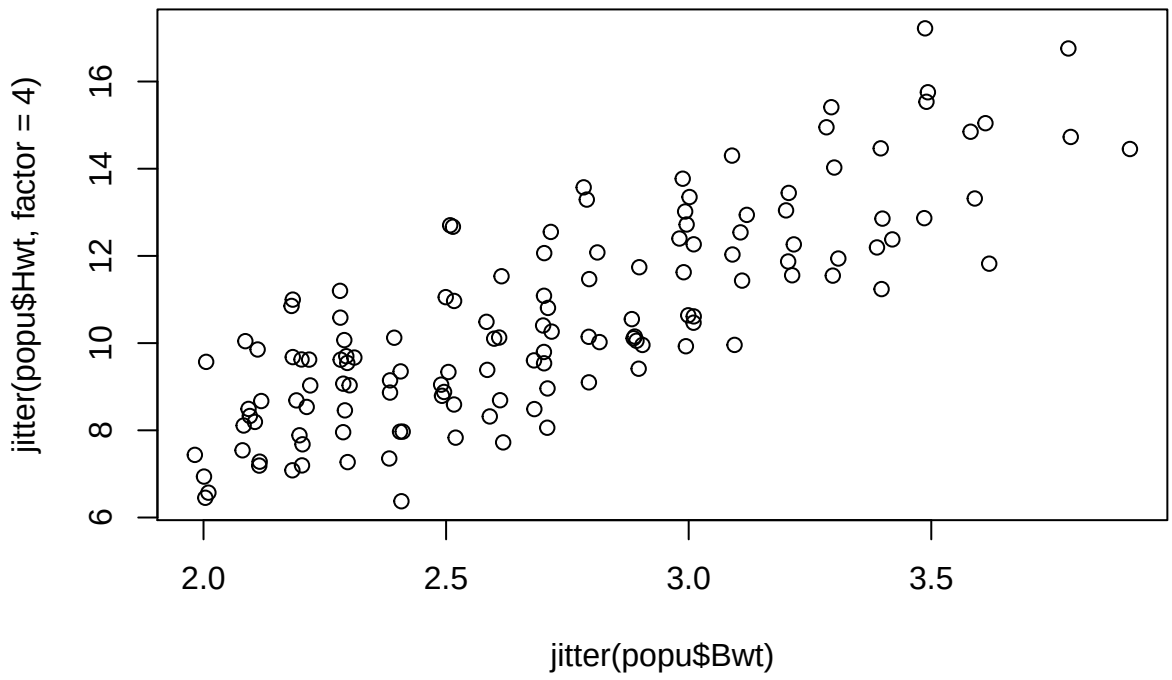



```
# Retrait d'un échantillon de donnees servant de test
# Tirage aleatoire d'indice a retirer
ech =sample(1:nrow(cats),10)
# Extraction des indices
popu = cats[-ech,]
# ou
# Extraire des donnees d'une data
```

```r
# Ici on extrait toutes les lignes ou l'espèces n'est pas 'Iris_setosaa'
# data2<- subset(data, data$species!='Iris-setosa')


# HWt en fonction de Bwt
plot(Bwt~Hwt,data=popu) # ou plot(Bwt,Hwt,data=popu)
```




```r
# meilleur affichage
# augermet le bruit avec factor
plot(jitter(popu$Bwt),jitter(popu$Hwt, factor=4))
```




```
# Corrélation entre Bwt et Hwt
cor(popu$Bwt,popu$Hwt)
```

```
## [1] 0.8131353
```

```
# Régression linéaire
reg<-lm(popu$Hwt~popu$Bwt) # Y~Var1+Var2+Var3
summary(reg)
```

```
##
## Call:
## lm(formula = popu$Hwt ~ popu$Bwt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0002 -0.9555 -0.1183  0.9737  3.4556
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3964     0.6927  -0.572    0.568
## popu$Bwt      4.0402     0.2517  16.050   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.37 on 132 degrees of freedom
## Multiple R-squared:  0.6612, Adjusted R-squared:  0.6586
## F-statistic: 257.6 on 1 and 132 DF,  p-value: < 2.2e-16
```

(Intercept) 1.038601* = **a** = *coefficient directeur de la droite*

cats$Hwt 0.159450 = **b** = *ordonnée à l'origine*

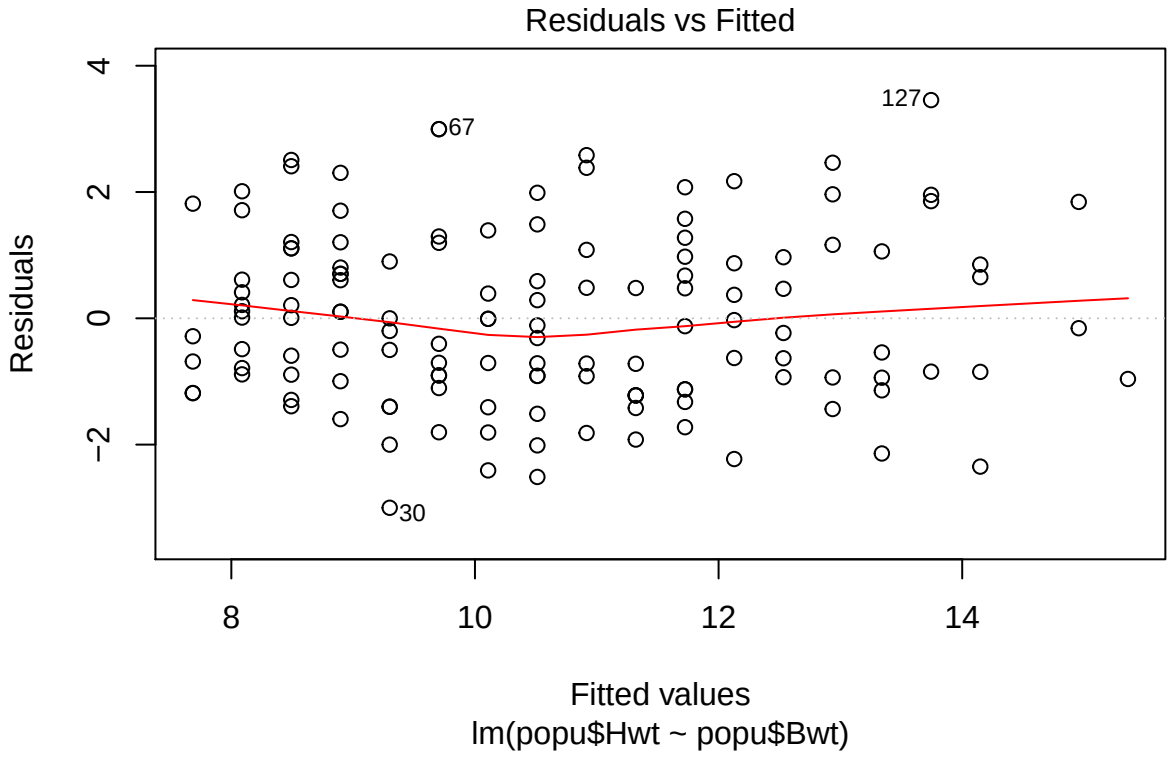
Std. Error : *De petites valeurs sont un gage de stabilité du modèle donc du pouvoir prédictif* :(Ici valeur pour b de 0.72 et pour a de 0.26 très stable)

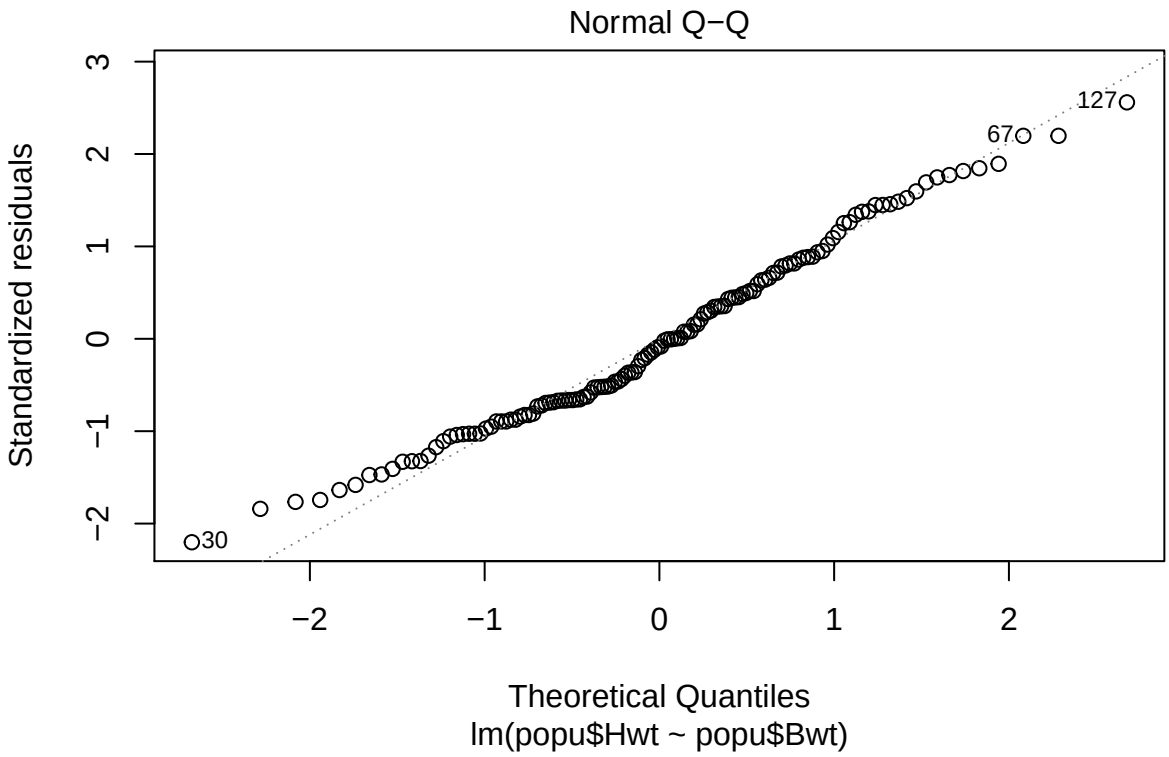
Residual standard error : *Écart-type résiduel doit être faible pour bon pouvoir prédictif* (Ici 1.458 ce qui est faible)

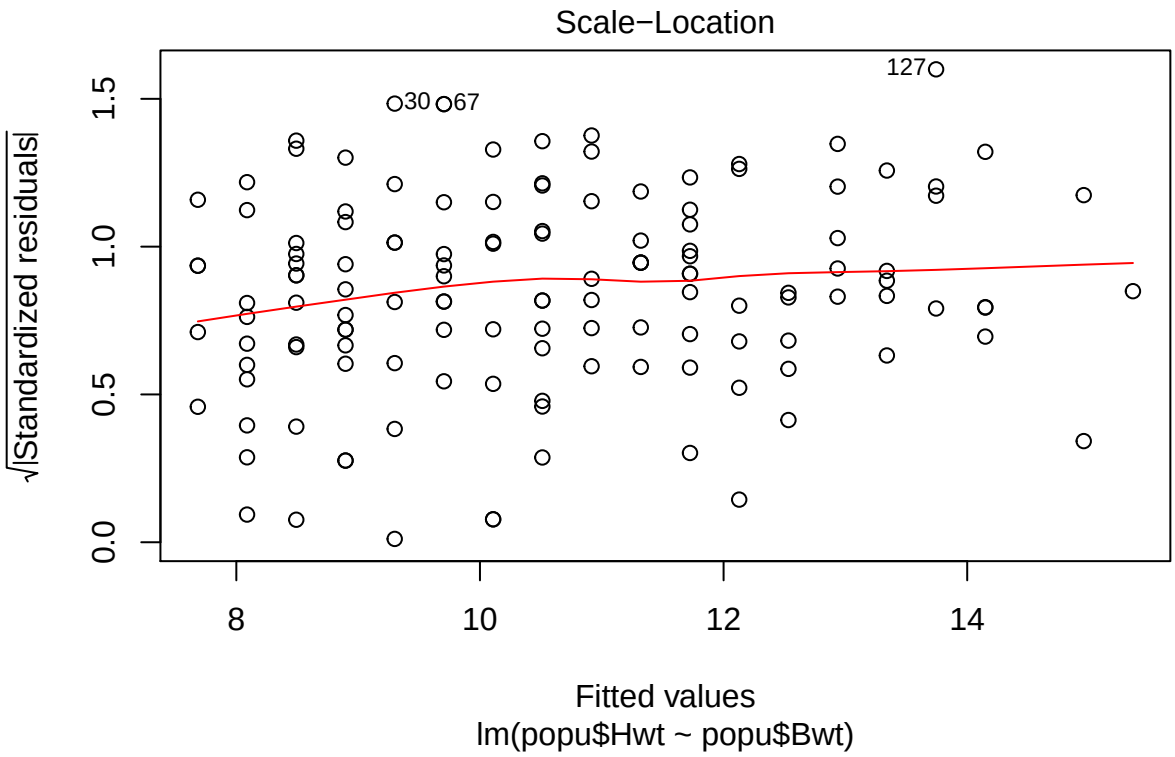
Multiple R-squared *% de la variance de Hwt expliquée par Bwt, doit être proche de 1 pour avoir un bon pouvoir explicatif* :(Ici 64%)

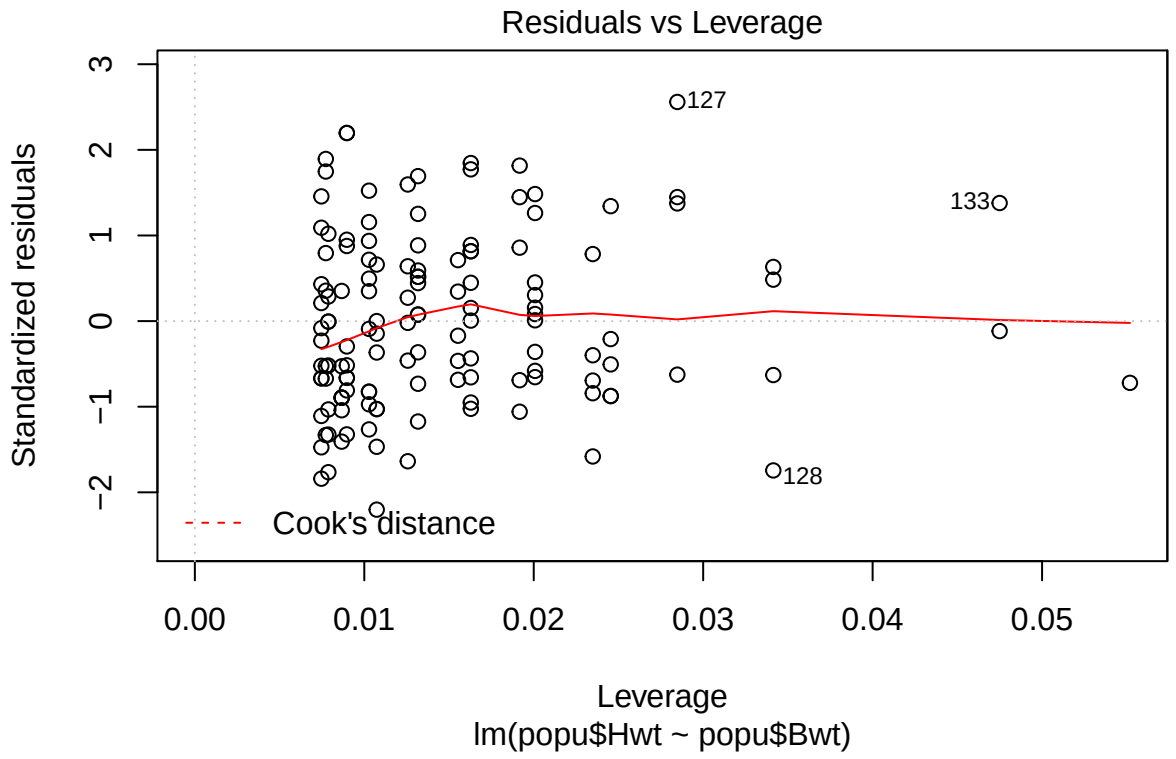
p-value <2e-16 *probabilité que l'ordonnée à l'origine soit proche de 0*

```
plot(reg)
```




Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
127
67
30
−2
−1
0
1
2
Theoretical Quantiles
lm(popu$Hwt ~ popu$Bwt)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
30
67
127
8
10
12
14
Fitted values
lm(popu$Hwt ~ popu$Bwt)




```
# les résidus
reg$residuals # ou residuals(reg)
```

```
##             1             2             3             4             5
## -0.6840840992 -0.2840840992  1.8159159008 -0.8881058523 -0.7881058523
##             6             7             8             9            10
## -0.4881058523  0.0118941477  0.1118941477  0.2118941477  0.4118941477
##            11            12            13            14            15
##  0.6118941477  1.7118941477 -1.3921276054  0.2078723946  0.6078723946
##            16            17            18            19            20
##  1.2078723946  2.4078723946  2.5078723946 -1.5961493585 -0.9961493585
##            21            22            23            24            25
## -0.4961493585  0.1038506415  0.1038506415  0.6038506415  0.7038506415
##            26            27            28            29            30
##  0.8038506415  1.2038506415  1.7038506415  2.3038506415 -3.0001711116
##            31            32            33            34            35
## -0.5001711116  0.8998288884 -0.7041928647  1.1958071353 -1.4082146178
##            36            37            38            39            40
## -0.0082146178 -0.0082146178 -2.0122363709 -0.3122363709  0.2877636291
##            41            42            43            44            45
## -1.4202798771 -1.2202798771 -1.2202798771 -1.1243016302  1.2756983698
##            46            47            48            49            50
## -1.1840840992 -1.1840840992  2.0118941477 -1.2921276054 -0.8921276054
##            51            52            53            54            55
## -0.5921276054  0.0078723946  1.1078723946  1.1078723946  0.7038506415
##            56            57            58            59            60
```

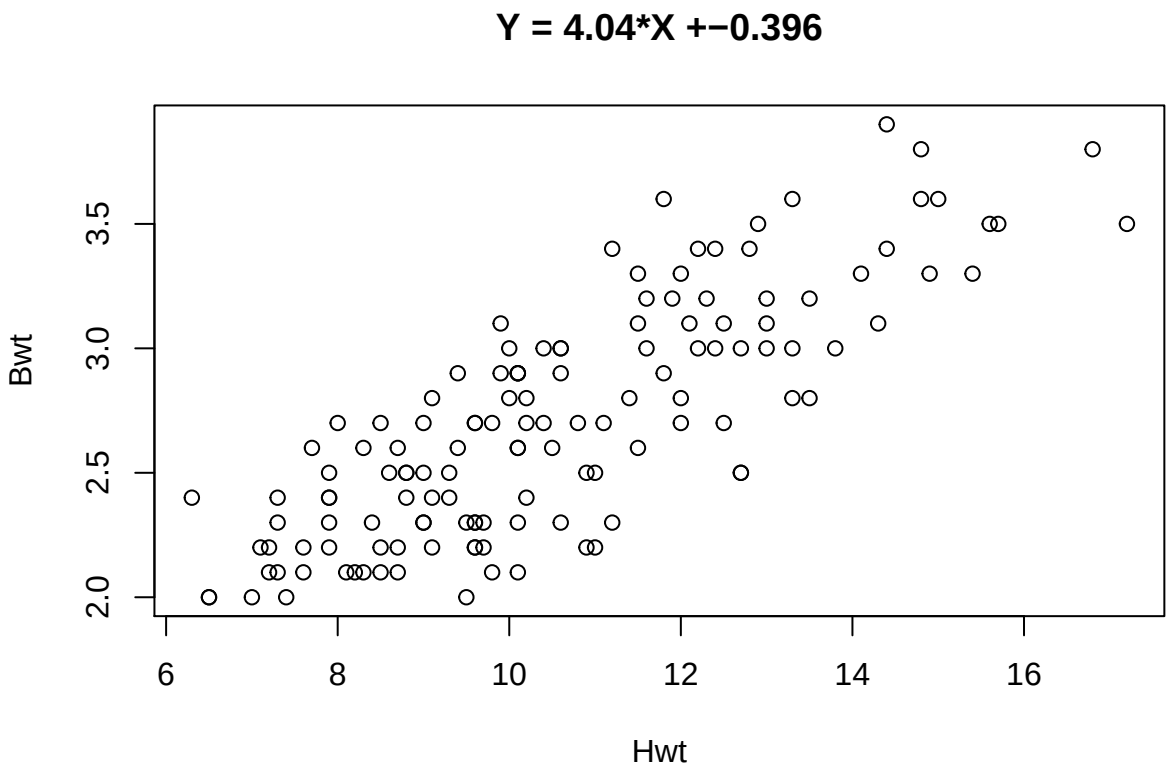
```
## -2.0001711116 -1.4001711116 -1.4001711116 -0.2001711116 -0.0001711116
##            61            62            63            64            65
## -1.8041928647 -1.1041928647 -0.9041928647 -0.9041928647 -0.4041928647
##            66            67            68            69            70
##  1.2958071353  2.9958071353  2.9958071353 -2.4082146178 -1.8082146178
##            71            72            73            74            75
## -0.7082146178  0.3917853822  1.3917853822 -2.5122363709 -1.5122363709
##            76            77            78            79            80
## -0.9122363709 -0.9122363709 -0.7122363709 -0.1122363709  0.5877636291
##            81            82            83            84            85
##  1.4877636291  1.9877636291 -1.8162581240 -0.9162581240 -0.7162581240
##            86            87            88            89            90
##  0.4837418760  1.0837418760  2.3837418760  2.5837418760 -1.9202798771
##            91            92            93            94            95
## -1.2202798771 -0.7202798771  0.4797201229 -1.7243016302 -1.3243016302
##            96            97            98            99           100
## -1.1243016302 -0.1243016302  0.4756983698  0.6756983698  0.9756983698
##           101           102           103           104           105
##  1.5756983698  2.0756983698 -2.2283233833 -0.6283233833 -0.0283233833
##           106           107           108           109           110
##  0.3716766167  0.8716766167  2.1716766167 -0.9323451364 -0.6323451364
##           111           112           113           114           115
## -0.2323451364  0.4676548636  0.9676548636 -1.4363668895 -0.9363668895
##           116           117           118           119           120
##  1.1636331105  1.9636331105  2.4636331105 -2.1403886426 -1.1403886426
##           121           122           123           124           125
## -0.9403886426 -0.5403886426  1.0596113574 -0.8444103957  1.8555896043
##           126           127           128           129           130
##  1.9555896043  3.4555896043 -2.3484321488 -0.8484321488  0.6515678512
##           131           132           133           134
##  0.8515678512 -0.1564756550  1.8435243450 -0.9604974081
```

```
# les valeurs des Y(ici Hwt) ajustées
reg$fitted.values # ou fitted.values(reg)
```

```
##         1         2         3         4         5         6         7
##  7.684084  7.684084  7.684084  8.088106  8.088106  8.088106  8.088106
##         8         9        10        11        12        13        14
##  8.088106  8.088106  8.088106  8.088106  8.088106  8.492128  8.492128
##        15        16        17        18        19        20        21
##  8.492128  8.492128  8.492128  8.492128  8.896149  8.896149  8.896149
##        22        23        24        25        26        27        28
##  8.896149  8.896149  8.896149  8.896149  8.896149  8.896149  8.896149
##        29        30        31        32        33        34        35
##  8.896149  9.300171  9.300171  9.300171  9.704193  9.704193 10.108215
##        36        37        38        39        40        41        42
## 10.108215 10.108215 10.512236 10.512236 10.512236 11.320280 11.320280
##        43        44        45        46        47        48        49
## 11.320280 11.724302 11.724302  7.684084  7.684084  8.088106  8.492128
##        50        51        52        53        54        55        56
##  8.492128  8.492128  8.492128  8.492128  8.492128  8.896149  9.300171
##        57        58        59        60        61        62        63
##  9.300171  9.300171  9.300171  9.300171  9.704193  9.704193  9.704193
##        64        65        66        67        68        69        70
##  9.704193  9.704193  9.704193  9.704193  9.704193 10.108215 10.108215
```

```
##        71        72        73        74        75        76        77
## 10.108215 10.108215 10.108215 10.512236 10.512236 10.512236 10.512236
##        78        79        80        81        82        83        84
## 10.512236 10.512236 10.512236 10.512236 10.512236 10.916258 10.916258
##        85        86        87        88        89        90        91
## 10.916258 10.916258 10.916258 10.916258 10.916258 11.320280 11.320280
##        92        93        94        95        96        97        98
## 11.320280 11.320280 11.724302 11.724302 11.724302 11.724302 11.724302
##        99       100       101       102       103       104       105
## 11.724302 11.724302 11.724302 11.724302 12.128323 12.128323 12.128323
##       106       107       108       109       110       111       112
## 12.128323 12.128323 12.128323 12.532345 12.532345 12.532345 12.532345
##       113       114       115       116       117       118       119
## 12.532345 12.936367 12.936367 12.936367 12.936367 12.936367 13.340389
##       120       121       122       123       124       125       126
## 13.340389 13.340389 13.340389 13.340389 13.744410 13.744410 13.744410
##       127       128       129       130       131       132       133
## 13.744410 14.148432 14.148432 14.148432 14.148432 14.956476 14.956476
##       134
## 15.360497
```

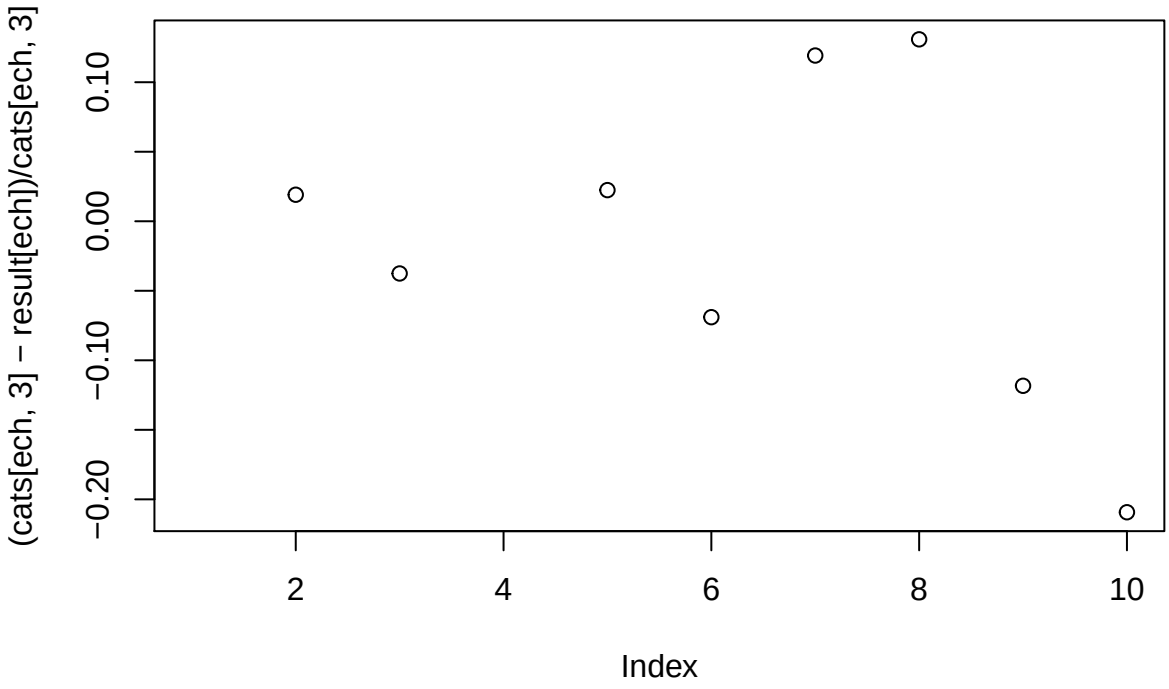
```r
# Traçage de la droite de régression linéaire
coeff=coefficients(reg)
# Equation de la droite de regression :
eq = paste0("Y = ", round(coeff[2],3), "*X +", round(coeff[1],3))
# Graphe
plot(Bwt~Hwt,data=popu,main=eq)
abline(reg,col='blue')
```




```
# calcul des prédictions
result<-predict(reg,cats[ech,])
```

```
## Warning: 'newdata' had 10 rows but variables found have 134 rows
```

```
# affichage de l'erreur pour chaque estimation
plot((cats[ech,3]-result[ech])/cats[ech,3],col='black')
```



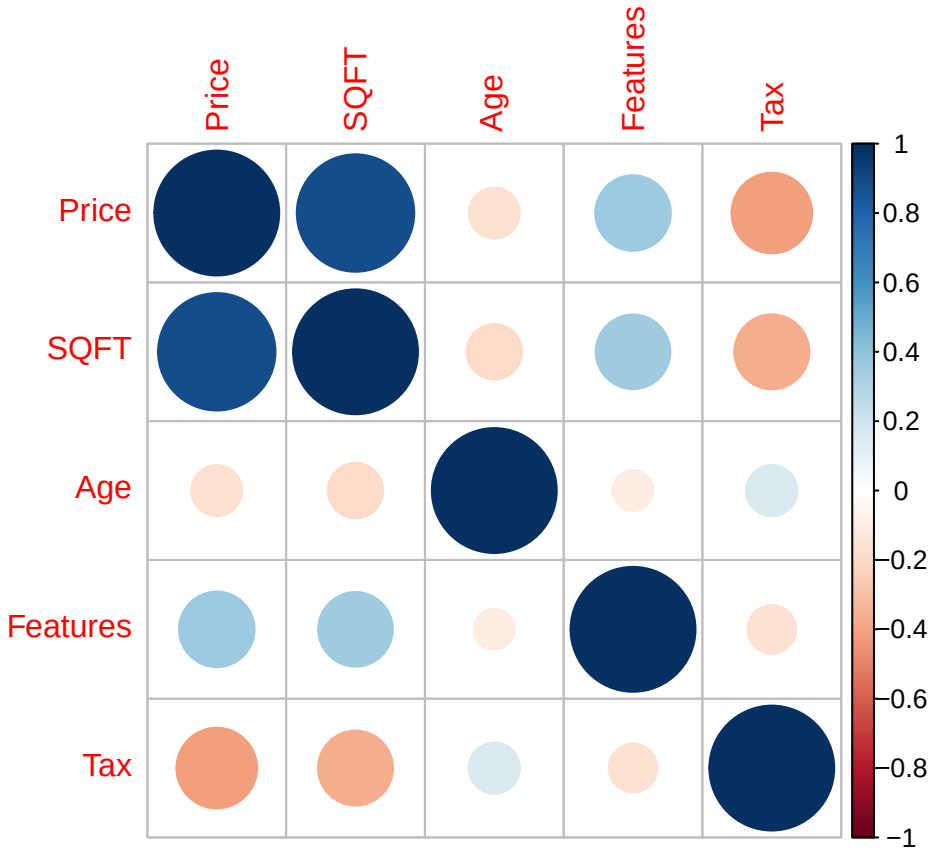
Multiple Linear Regression with prevision and AIC(Akaike Information Criterion)

```
houses <- read.csv("C:/Users/DJEBALI/Documents/M2_ISIFAR/Data_Mining/houses.txt", sep="")
houses$NE<-as.factor(houses$NE)
houses$Corner<-as.factor(houses$Corner)

# Données test
sampl=sample(1:nrow(houses),10)

houses_sampl=houses[sampl,]

# Description des variables quantitatives
summary(houses)
```

```
##      Price            SQFT           Age           Features     NE
##  Min.   : 580   Min.   : 970   Min.   : 2.00   Min.   :1.000   0:25
##  1st Qu.: 905   1st Qu.:1404   1st Qu.: 7.25   1st Qu.:3.000   1:41
##  Median :1050   Median :1690   Median :20.00   Median :4.000
##  Mean   :1169   Mean   :1751   Mean   :17.44   Mean   :3.985
##  3rd Qu.:1265   3rd Qu.:1926   3rd Qu.:26.75   3rd Qu.:4.000
##  Max.   :2150   Max.   :2931   Max.   :31.00   Max.   :8.000
##  Corner      Tax
##  0:51   Min.   : 2.00
##  1:15   1st Qu.:23.50
##         Median :56.50
```

```
##          Mean   :52.14
##          3rd Qu.:77.75
##          Max.   :96.00
```

```r
# Corrélation des variables quantitatives
corrplot(cor(houses[c(-5,-6)]))
```




```r
# Données d'entraînement
houses_entr = houses[-sampl,]

# Régression lineaire : modéle linéaire gaussien multiple
# regmult<-glm(Price~SQFT+Age+Features+Tax,houses_entr,family=gaussian())
regmult<-lm(Price~SQFT+Age+Features+Tax,houses_entr)
summary(regmult)
```

```
##
## Call:
## lm(formula = Price ~ SQFT + Age + Features + Tax, data = houses_entr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -530.39  -66.87    2.02   71.29  346.62
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -49.49336  127.74438  -0.387   0.7000
## SQFT          0.61772    0.05091  12.133   <2e-16 ***
```

```
## Age           1.45834    2.50938   0.581   0.5637
## Features     35.18351   20.76952   1.694   0.0964 .
## Tax          -0.78987    0.81475  -0.969   0.3369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170.9 on 51 degrees of freedom
## Multiple R-squared:  0.8103, Adjusted R-squared:  0.7954
## F-statistic: 54.46 on 4 and 51 DF,  p-value: < 2.2e-16
```

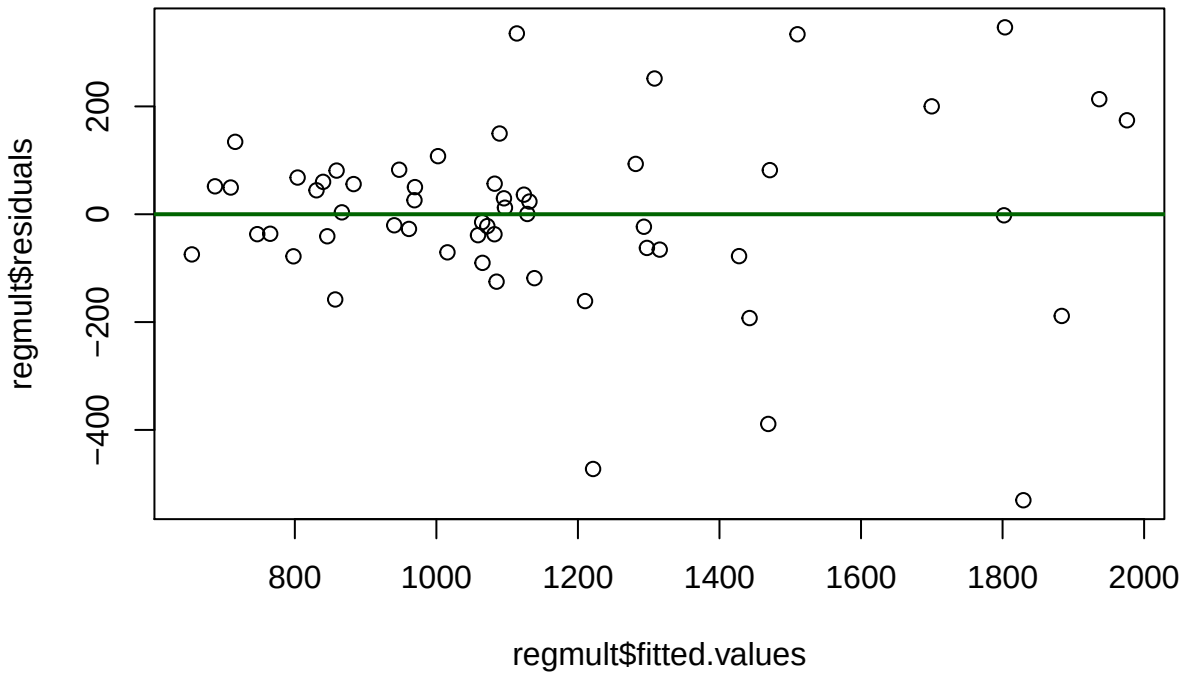
Comme en régression linéaire simple, les informations données par la fonction summary() concernent :

_ les résidus (maximum, minimum, quartiles)

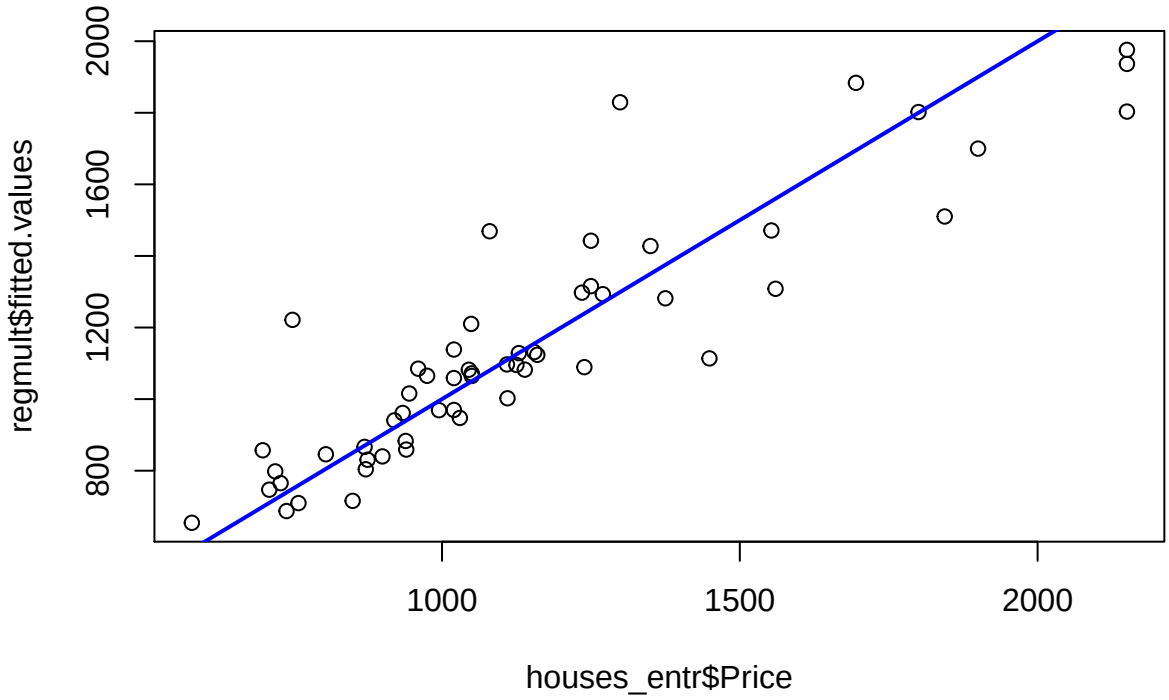
_ les coefficients : en plus des estimations des β_j (Estimate), nous avons l'écart-type estimé des estimateurs correspondants (Std. Error), ainsi que la valeur de la statistique de test (t value) et la p-value (Pr(>|t|)) associées aux tests de Student ($H_0 : \beta_j = 0$ contre $H_1 : \beta_j \neq 0$) correspondants. A noter que des étoiles pour chaque coefficient indiquent le niveau de significativité des différents tests.

_ la qualité d'adéquation du modèle : une estimation de σ, l'écart-type du terme d'erreur (Residual standard error), la valeur du R2 (Multiple R-squared) et celle du R2 ajusté (Adjusted R-squared), et enfin la valeur de la statistique de test (F-statistic) et la p-value du test de Fisher de significativité du modèle ($H_0 : \beta_1 = ... = \beta_p = 0$ contre H_1 : au moins un $\beta_j \neq 0$). Ce dernier test permet de tester la nullité simultanée de tous les coefficients associés aux variables explicatives. Ainsi accepter H0 signifie que le modèle proposé n'est pas adéquat.

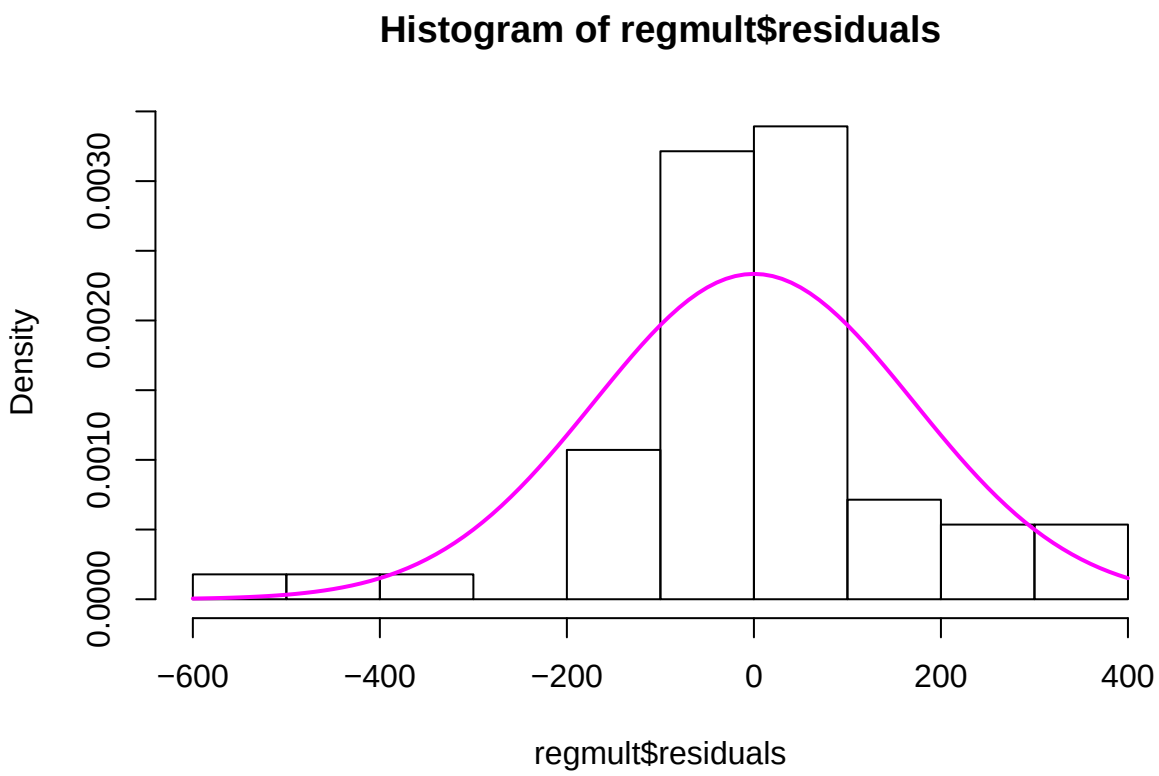
```r
# Graphique des résidus
plot(regmult$fitted.values, regmult$residuals)
abline(h = 0, col = "darkgreen", lwd = 2)
```



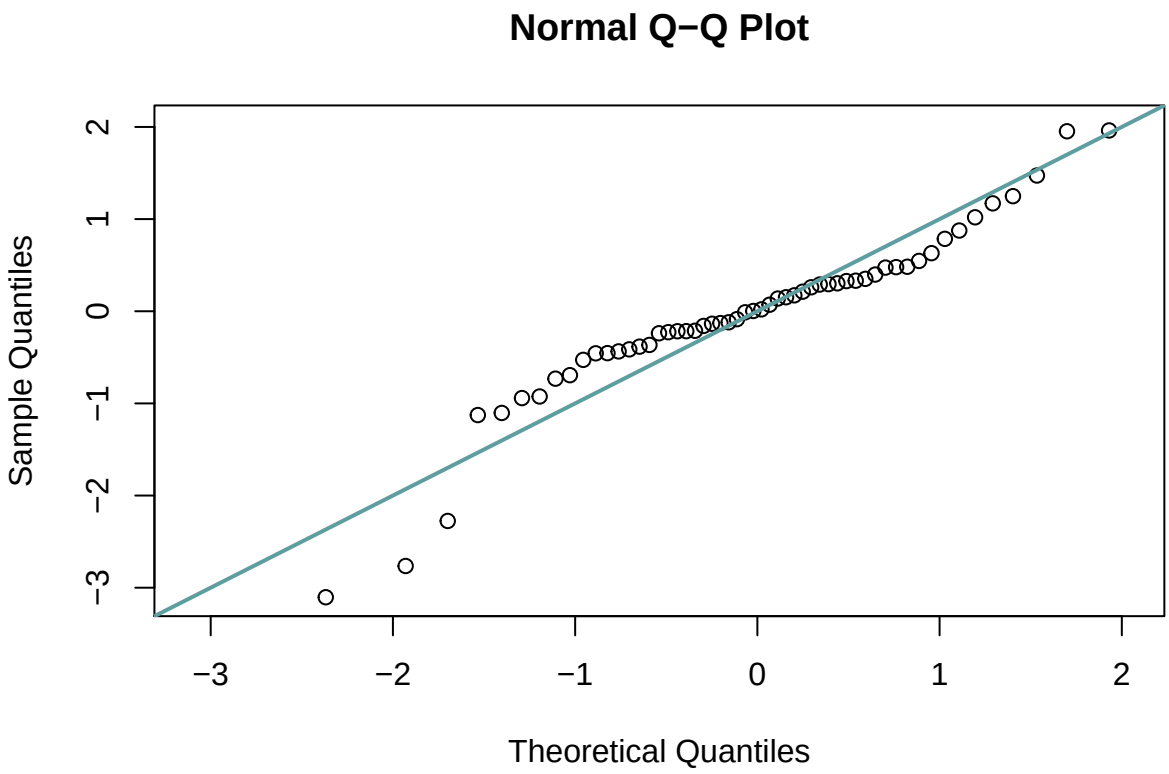
```r
# Graphique des valeurs ajsutées en fonction des valeurs observées
plot(houses_entr$Price, regmult$fitted.values)
abline(a = 0, b = 1, col = "blue", lwd = 2)
```




```r
# Histogramme des résidus
histo <- hist(regmult$residuals, probability = TRUE)
ec_typ <- summary(regmult)$sigma
curve(dnorm(x, 0, ec_typ), from = min(histo$breaks), to = max(histo$breaks),
    add = TRUE, type = "l", col = "magenta", lwd = 2)
```




```r
# On voit qu'on a des résidus gaussiens centrés réduits

# Graphe quantile par quantile
ec_typ <- summary(regmult)$sigma
normed_res <- regmult$residuals/ec_typ
qqnorm(normed_res, xlim = range(normed_res), ylim = range(normed_res))
abline(0, 1, col = "cadetblue", lwd = 2)
```




```r
# Test de normalité des résidus
shapiro.test(regmult$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  regmult$residuals
## W = 0.92221, p-value = 0.001455
```

```r
# p<0.05 donc on ne rejette pas l'hypothèse de gaussianité de l'échantillon
```

AIC(Akaike Information Criterion)

Le critère utilisé par défaut dans R est le critère AIC (pour "An Information Criterion"", proposé par Akaike, on parle aussi de critère d'Akaike).

La formule du critère AIC (sur lequel la méthodologie de sélection de variables est fondée) est la suivante :

$$AIC = n \log(\frac{RSS^*}{n}) + 2(p^* + 1)$$

où p^* correspond au nombre de variables explicatives considérées dans le modèle courant (i.e. celui pour lequel on est en train de calculer l'AIC), et $RSS^* = \sum_{i=1}^{n}(y_i - y_i^*)^2$ est la somme des carrés des résidus du modèle courant (RSS pour Residual Sum of Squares en anglais).

Notons que la quantité RSSn correspond à l'estimation du paramètre σ^2.

Trois types de sélections avec la fonction la step :

_ **Sélection descendante** : *step(regmult, direction='backward')*
On part du modèle complet puis on enlève à chaque itération la variable la variable qui explique le moins Y.

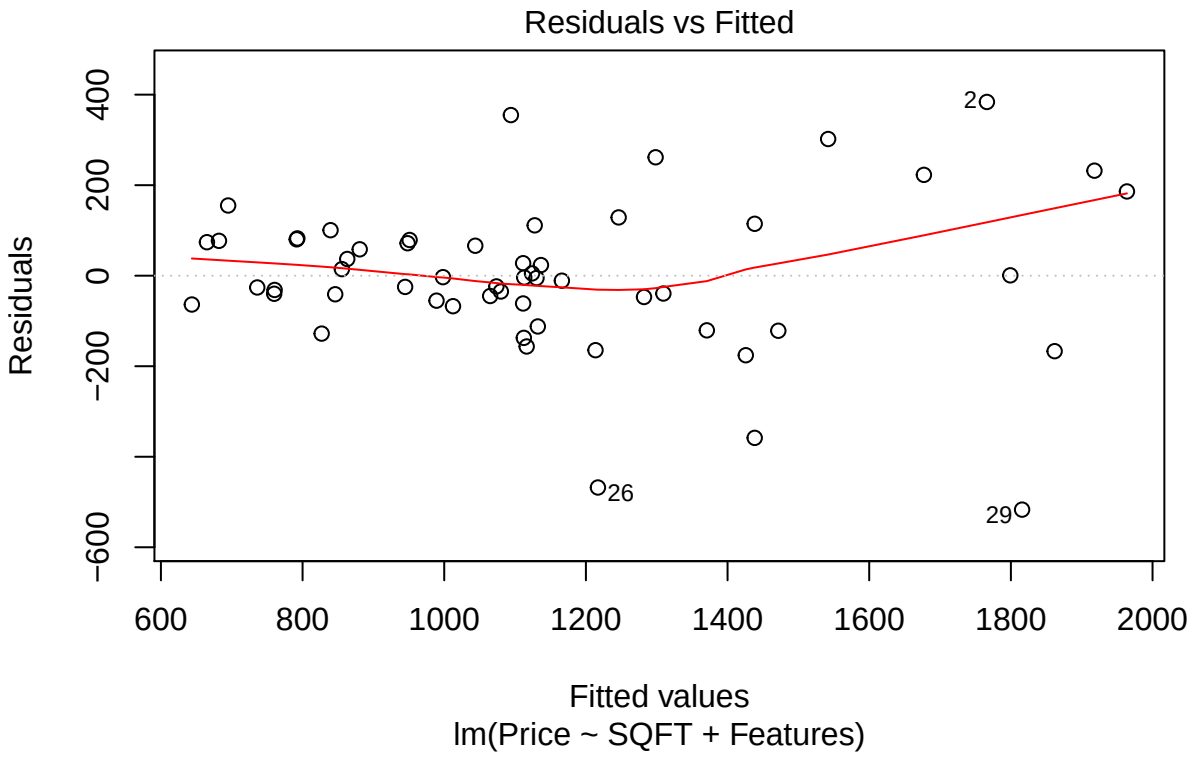
_ **Sélection ascendante** : *step(regmult, direction='forward')*
On part du modèle sans covariable et on insère à chaque itération la variable qui explique le plus Y et qui est le plus significative.

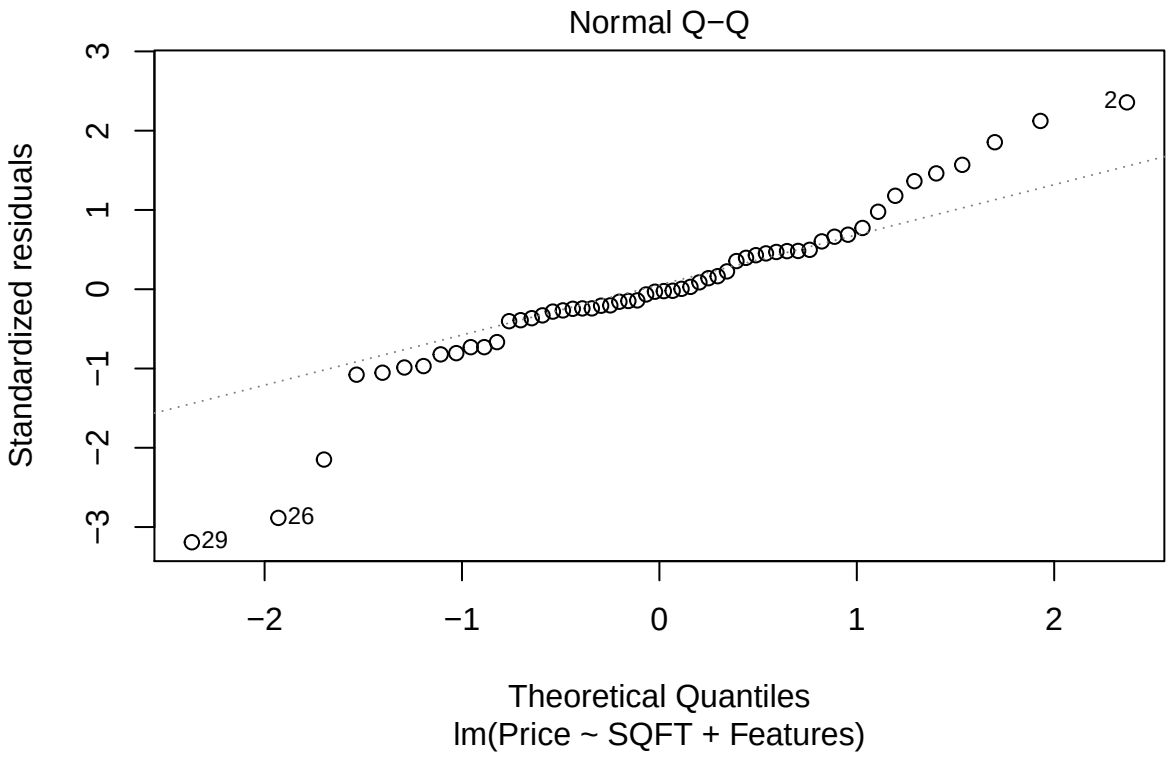
_ **Sélection pas à pas (stepwise)** : *step(regmult,direction='both')*
On part de la méthode ascendante avec remise en cause à chaque étape des variables déjà introduites. Cette pratique permet d'éléminer les variables qui ne sont plus informatives compte tenu de celle qui vient d'être sélectionnée.

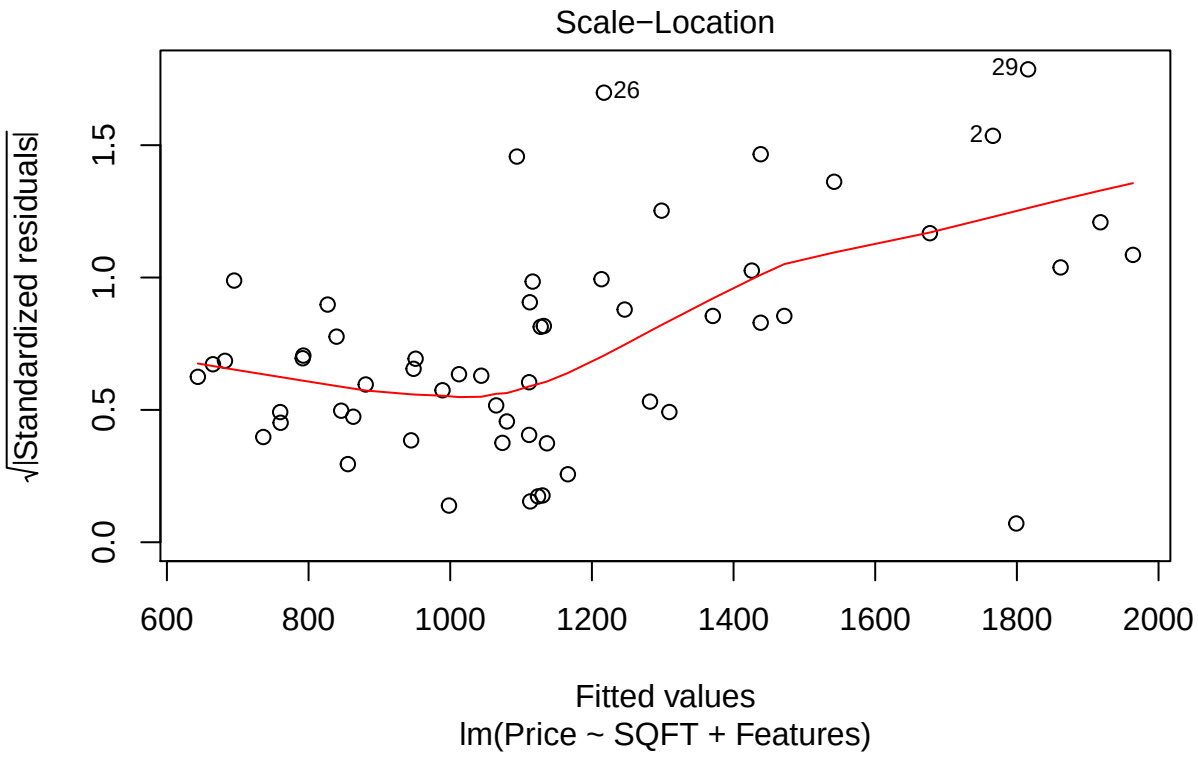
```
aic<-step(regmult)
```

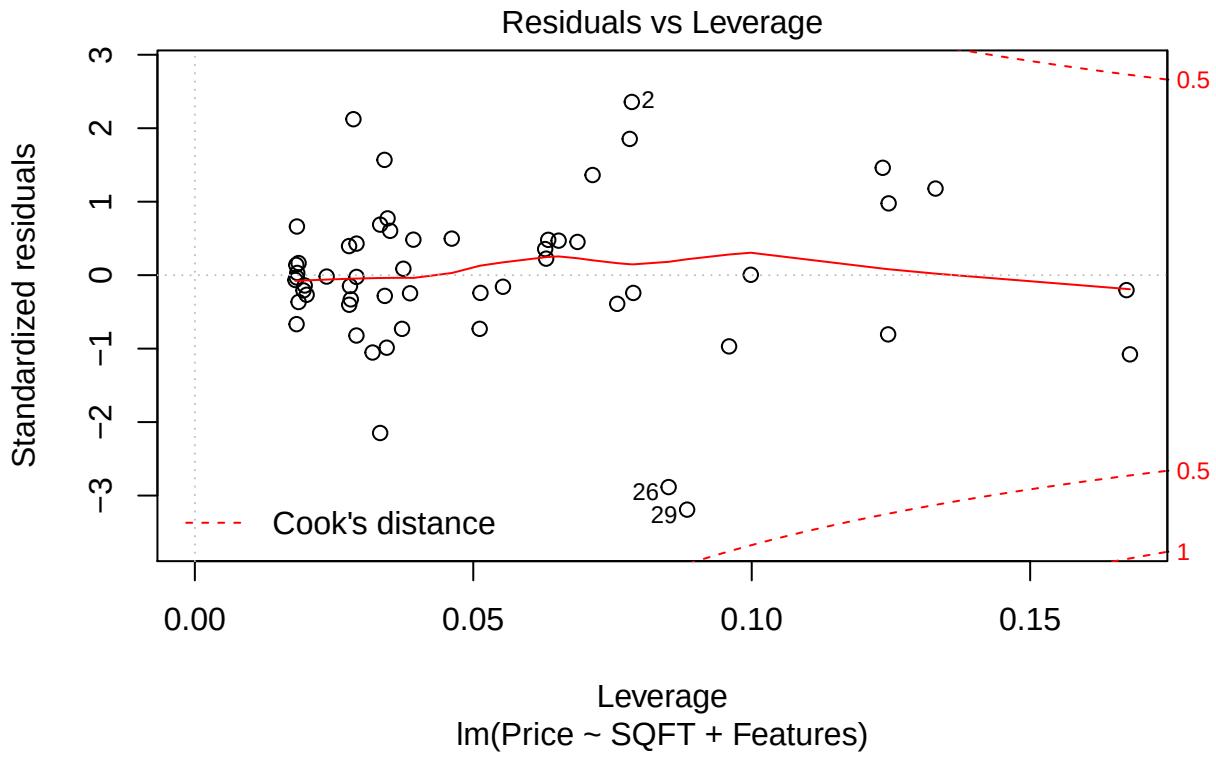
```
## Start:  AIC=580.56
## Price ~ SQFT + Age + Features + Tax
##
##             Df Sum of Sq     RSS    AIC
## - Age        1      9864 1499423 578.93
## - Tax        1     27451 1517009 579.59
## <none>                   1489559 580.56
## - Features   1     83813 1573372 581.63
## - SQFT       1   4299748 5789307 654.59
##
## Step:  AIC=578.93
## Price ~ SQFT + Features + Tax
##
##             Df Sum of Sq     RSS    AIC
## - Tax        1     25179 1524602 577.87
## <none>                   1499423 578.93
## - Features   1     85668 1585092 580.04
## - SQFT       1   4300610 5800034 652.69
##
## Step:  AIC=577.87
## Price ~ SQFT + Features
##
##             Df Sum of Sq     RSS    AIC
## <none>                   1524602 577.87
## - Features   1     88557 1613159 579.03
## - SQFT       1   4925672 6450274 656.64
```

```
# On garde les variables SQFT et Tax qui expliquent le mieux Price
plot(aic)
```


Residuals vs Fitted
Residuals
400
200
0
−200
−600
600
800
1000
1200
1400
1600
1800
2000
2
26
29
Fitted values
lm(Price ~ SQFT + Features)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Price ~ SQFT + Features)
29
26
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
26
29
2
600
800
1000
1200
1400
1600
1800
2000
Fitted values
lm(Price ~ SQFT + Features)




```
aic$residuals
```

```
##           2           3           5           6           7
##  383.8347416  186.1246900  222.7783579    0.8090606  261.5932839
##           8           9          10          11          12
##  354.8862408  128.7760245  -39.3905754 -120.7451552  -47.0603154
##          14          15          16          17          18
##  -11.0822672   66.1828582   27.6153043   -3.2173064   36.9178015
##          19          20          21          22          23
## -156.3773059 -166.7658831  114.6872908   71.7464314 -112.1320504
##          24          25          26          27          28
##  155.0785903  -39.9370812 -467.9705408  232.0203534 -121.6342182
##          29          30          31          32          33
## -516.8331682 -175.7385548  111.2689036   -5.2459272 -358.3127092
##          34          35          36          37          38
##  -61.3846957 -164.5682257  -55.1195703   82.3701174  -41.0700386
##          39          40          41          43          45
##   77.0226758  -31.5287374  -26.0461879   58.3139854  -63.6632049
##          46          47          48          49          50
##  301.9875689 -127.9500305   23.4669956   -3.9749908    5.0411499
##          51          52          54          56          57
##  -23.6622326  -34.9493098  -44.8603245   78.8987209 -137.3462830
##          58          59          60          62          63
##  100.5128667  -24.7771504  -67.3817558   80.2946532   14.4993456
##          66
##   73.9977842
```

```r
aic$coefficients
```

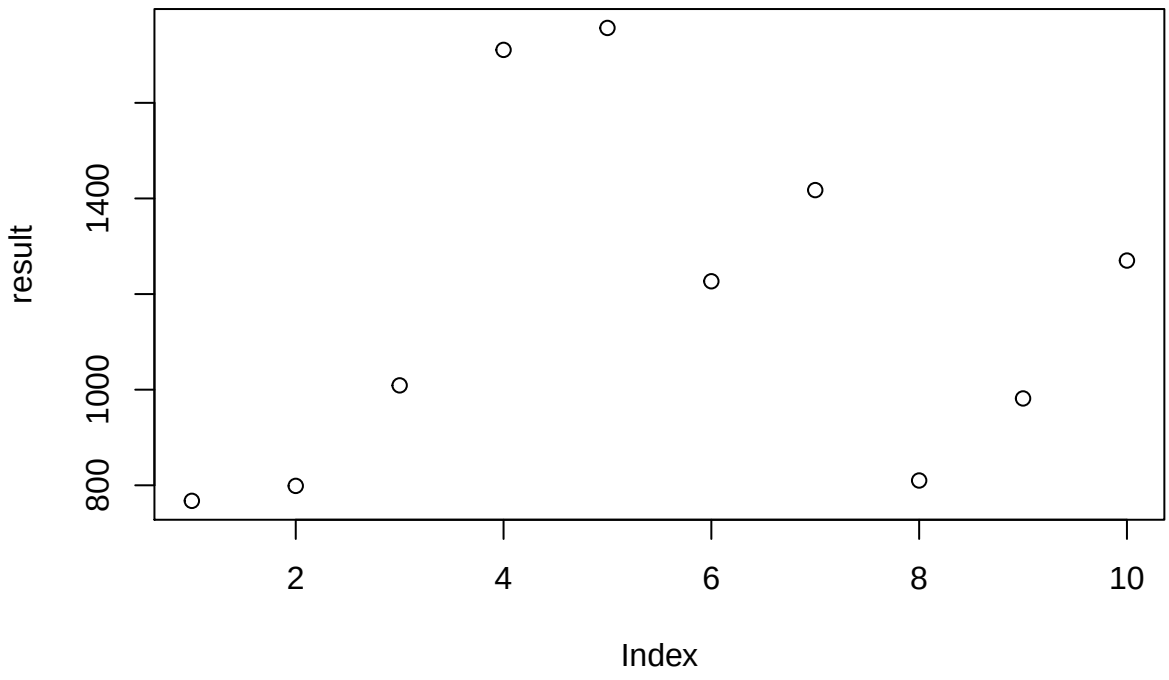
```
## (Intercept)        SQFT    Features
## -89.3729430   0.6287077  36.1321680
```

Prévisions

```r
reg2<-lm(Price~SQFT+Tax,houses_entr)

# Calcul des prédictions
result<-predict(reg2,houses_sampl)
result
```

```
##        64        65        55         4         1        53        44
##  767.4775  798.7794 1008.9947 1710.7831 1756.6692 1226.7384 1417.4734
##        61        42        13
##  810.0273  981.6732 1270.1037
```

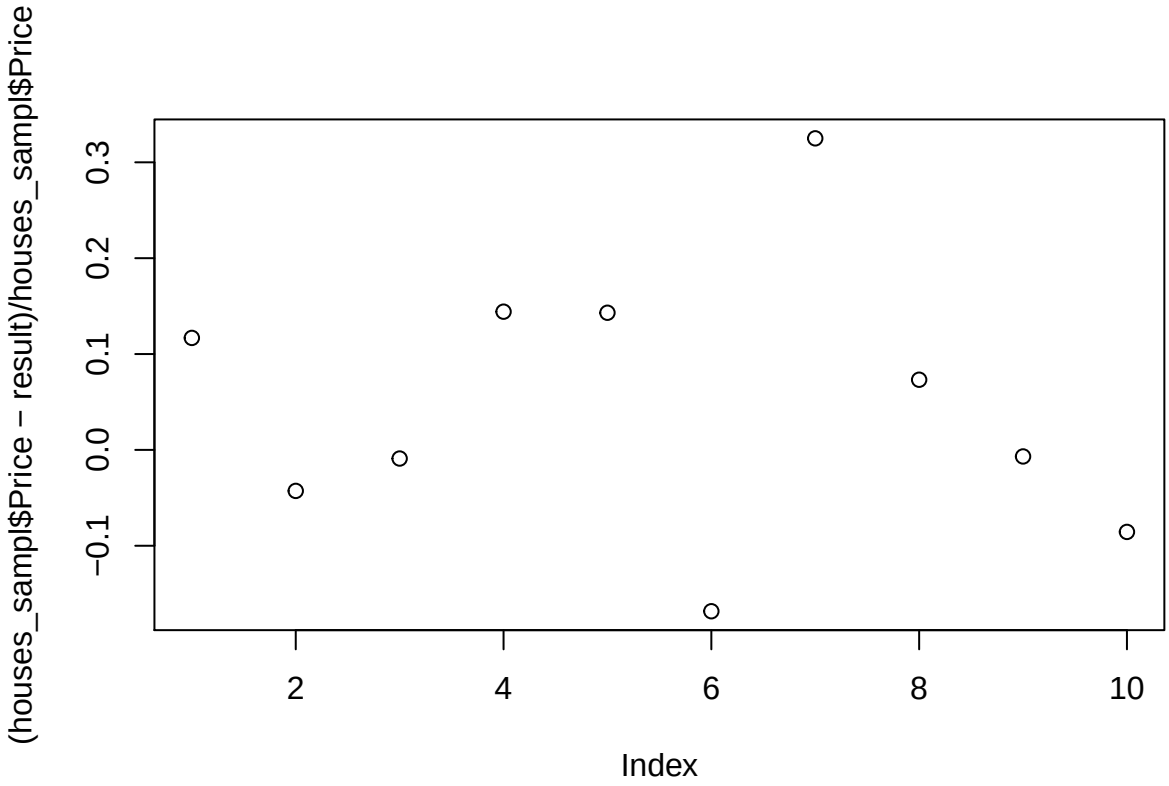
```r
plot(result)
```




```r
houses_sampl$Price
```

```
##  [1]  869  766 1000 1999 2050 1050 2100  874  975 1170
```

```r
# affichage de l'erreur pour chaque estimation
plot((houses_sampl$Price-result)/houses_sampl$Price,col='black')
```


(houses_sampl$Price − result)/houses_sampl$Price
−0.1
0.0
0.1
0.2
0.3
2
4
6
8
10
Index