Untitled

You ZUO, Jingzhuo HUI

2019/9/16

```r
tab <- as.data.frame(as.matrix(read.table("Files/TP1/immo.txt",header = T,sep = ";")))
```

```r
head(tab)
```

```
##   surface valeur prix
## 1   153.1    573  748
## 2   152.0    638  740
## 3   162.5    654  729
## 4   143.3    570  700
## 5   145.7    638  749
## 6   173.3    632  760
```

```r
names(tab)
```

```
## [1] "surface" "valeur"  "prix"
```

```r
tab[,1]
```

```
##  [1] 153.1 152.0 162.5 143.3 145.7 173.3 144.8 149.1 152.5 138.9 151.8
## [12] 144.4 148.7 186.3 152.0 257.6 190.5 153.7 180.6 163.5
```

```r
tab$surface
```

```
##  [1] 153.1 152.0 162.5 143.3 145.7 173.3 144.8 149.1 152.5 138.9 151.8
## [12] 144.4 148.7 186.3 152.0 257.6 190.5 153.7 180.6 163.5
```

```r
modreg <- lm(prix~., data=tab)
print(modreg)
```

```
##
## Call:
## lm(formula = prix ~ ., data = tab)
##
## Coefficients:
## (Intercept)      surface       valeur
##   309.66566      2.63440      0.04518
```

```r
summary(modreg)
```

```
##
## Call:
## lm(formula = prix ~ ., data = tab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.894 -15.411  -0.718  13.507  64.605
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 309.66566   78.82208   3.929  0.00108 **
## surface       2.63440    0.78560   3.353  0.00377 **
```

```
## valeur        0.04518    0.28518   0.158  0.87598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.73 on 17 degrees of freedom
## Multiple R-squared:  0.8344, Adjusted R-squared:  0.8149
## F-statistic: 42.83 on 2 and 17 DF,  p-value: 2.302e-07
```

```
attributes(modreg)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

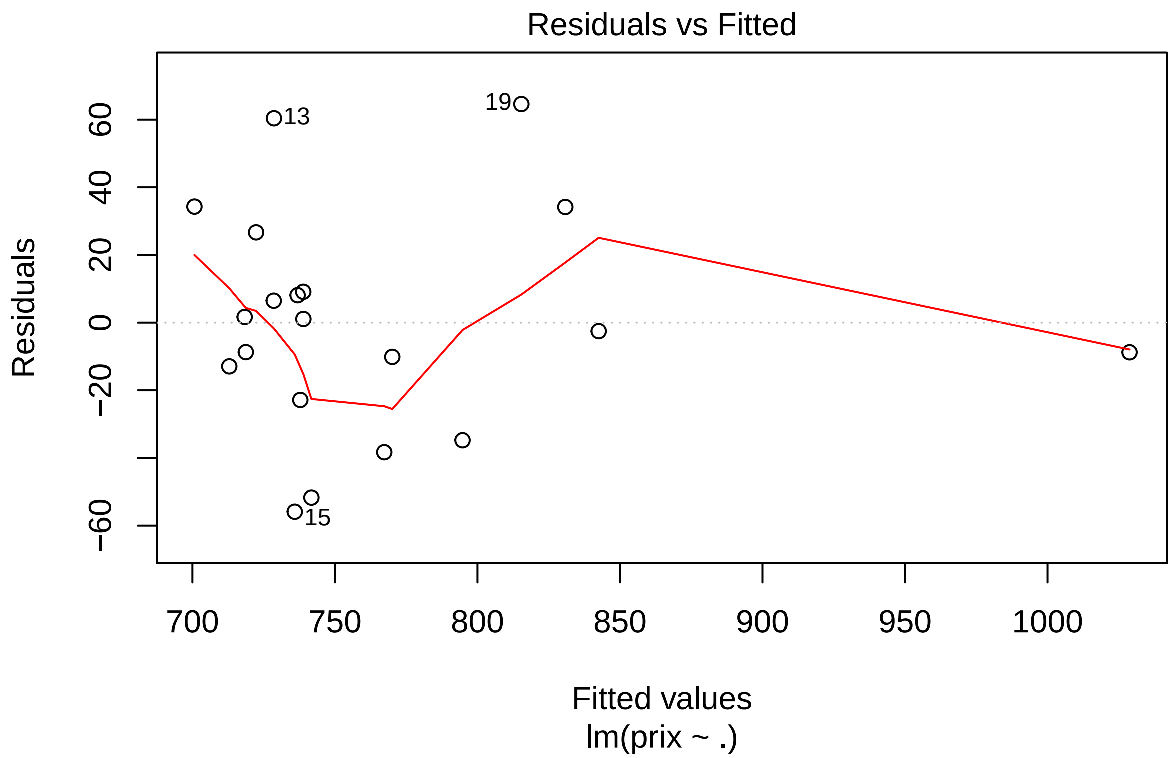
```
coef(modreg)
```

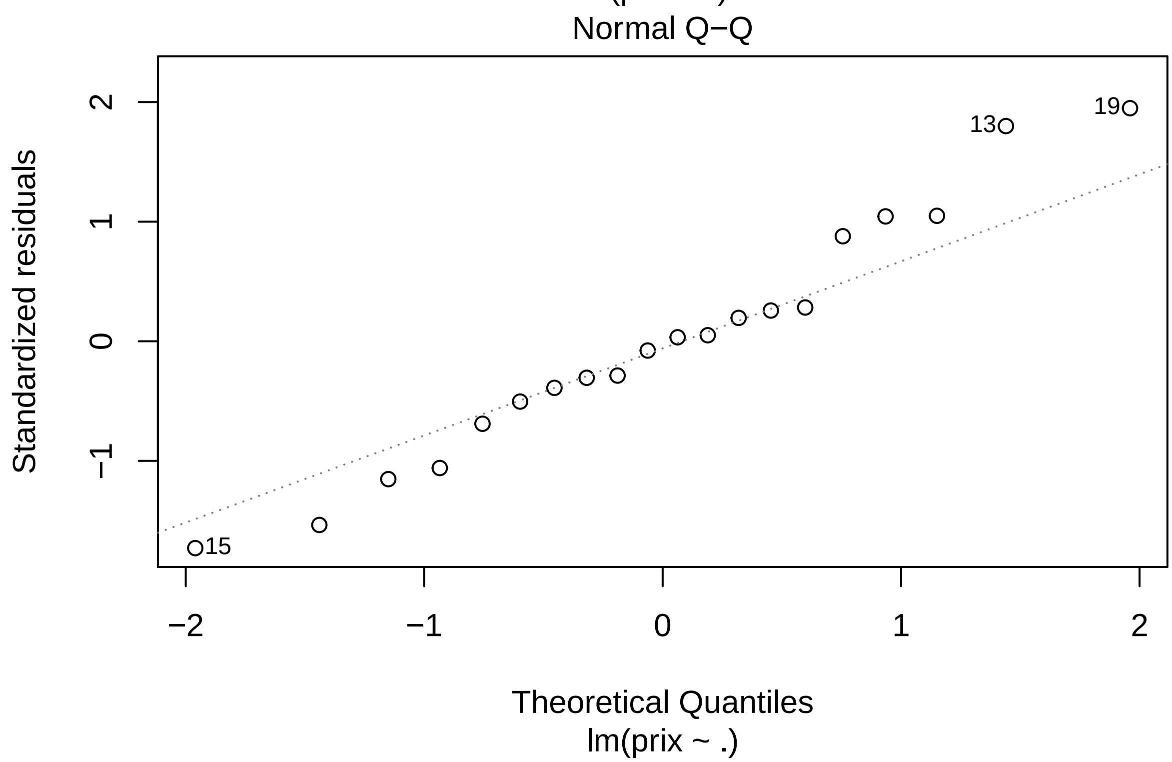
```
##  (Intercept)      surface       valeur
## 309.66566335   2.63439962   0.04518386
```

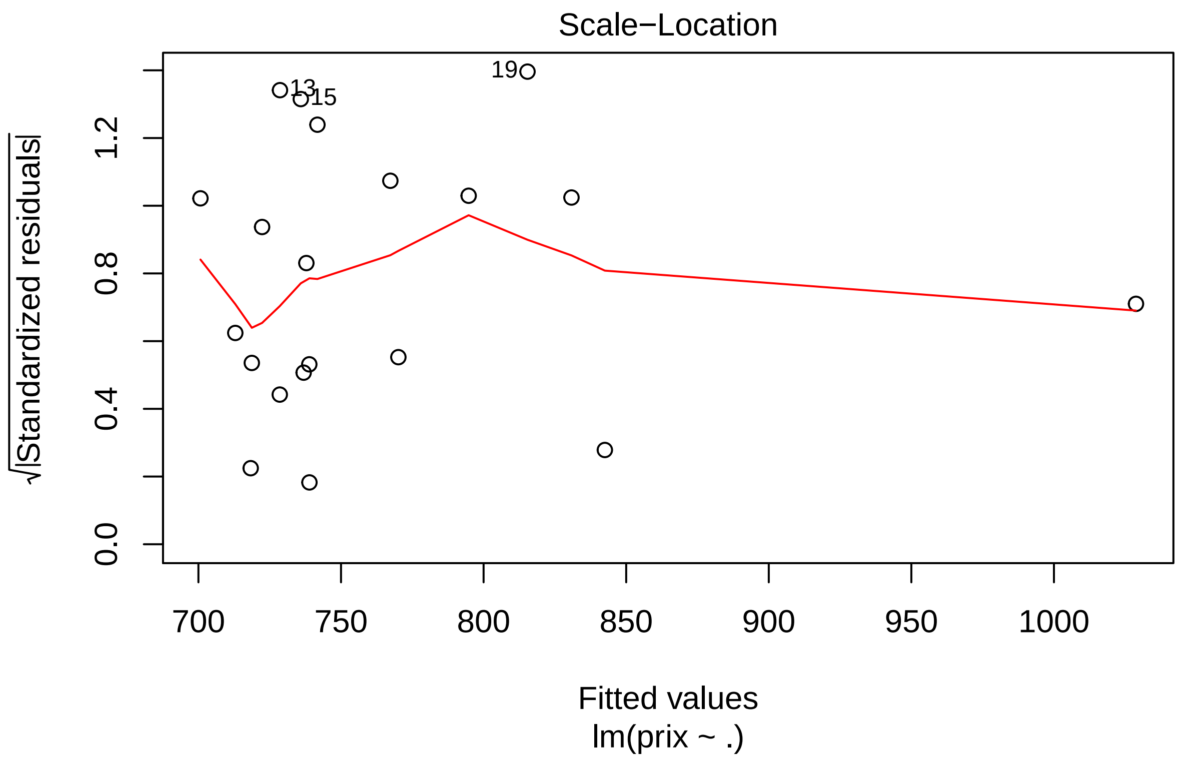
```
modreg$residuals
```

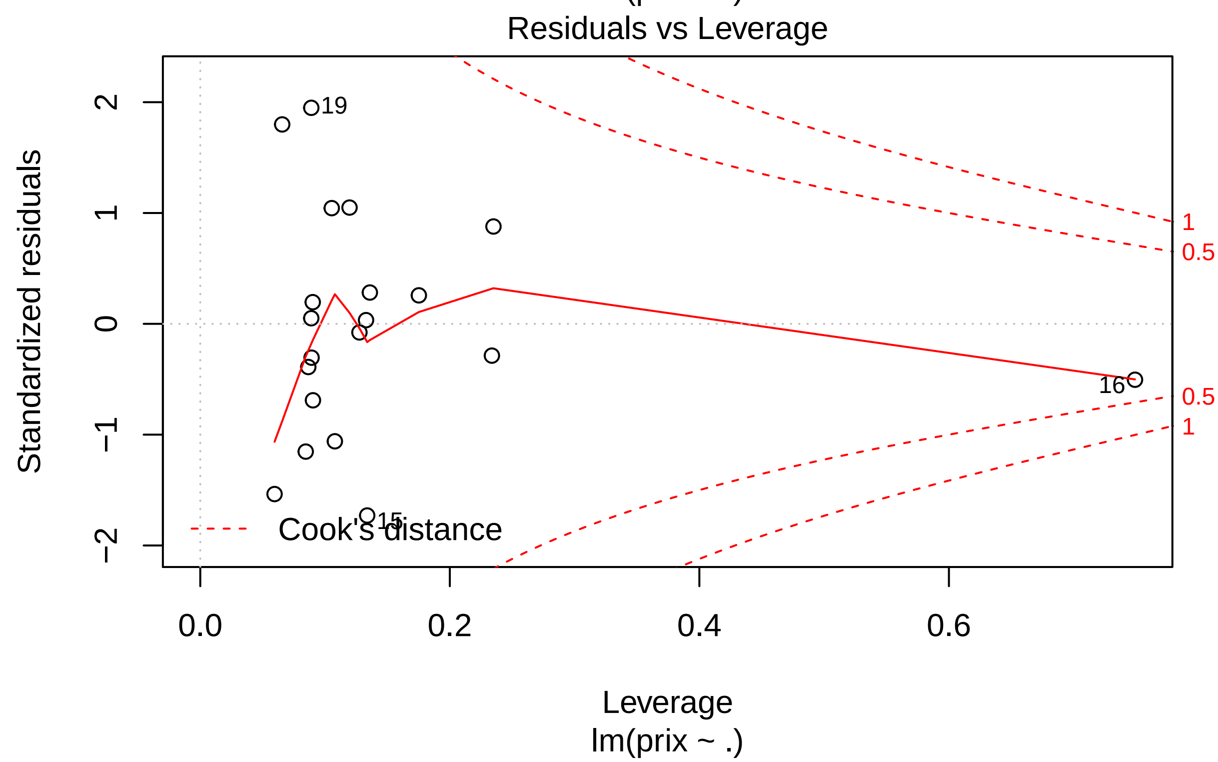
```
##          1          2          3          4          5          6
##   9.117402   1.078291 -38.305847 -12.929930  26.675008 -34.763318
##          7          8          9         10         11         12
##   1.672587   6.474265   8.104697  34.294002 -22.852623  -8.719537
##         13         14         15         16         17         18
##  60.398429  34.182132 -55.894391  -8.771746  -2.514920 -51.728386
##         19         20
##  64.604865 -10.120982
```

```
plot(modreg)
```


Residuals vs Fitted
Residuals
Fitted values
lm(prix ~ .)
13
19
15
60
40
20
0
−20
−60
700
750
800
850
900
950
1000


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(prix ~ .)
13
19
15
2
1
0
−1
−2






```
hat_y <- fitted(modreg)
ipsilon <- tab$prix - hat_y
```

ii

a)

```
icedata <- as.data.frame(read.table("Files/TP1/Icecreamdata.txt",sep = ";",header = T))
dim(icedata)
```

```
## [1] 30  4
```

$$cons = \beta_0 + \beta_1 \times income + \beta_2 \times price + \beta_3 \times temp$$

```
mod.ice <- lm(formula = cons~.,data = icedata)
mod.ice
```

```
##
## Call:
## lm(formula = cons ~ ., data = icedata)
##
## Coefficients:
## (Intercept)       income        price         temp
##    0.197315     0.003308    -1.044414     0.003458
```

```
summary(mod.ice)
```

```
##
## Call:
## lm(formula = cons ~ ., data = icedata)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.065302 -0.011873  0.002737  0.015953  0.078986
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1973151  0.2702162   0.730  0.47179
## income       0.0033078  0.0011714   2.824  0.00899 **
## price       -1.0444140  0.8343573  -1.252  0.22180
## temp         0.0034584  0.0004455   7.762  3.1e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03683 on 26 degrees of freedom
## Multiple R-squared:  0.719,  Adjusted R-squared:  0.6866
## F-statistic: 22.17 on 3 and 26 DF,  p-value: 2.451e-07
```

b) Estimated coefficients

```
library(matlib)
X <- as.matrix(cbind(rep(1,nrow(icedata)), icedata[,-1]))
Y <- as.matrix(icedata[,1])
beta <- inv(t(X)%*%X)%*%t(X)%*%Y
beta
```

```
##                [,1]
## [1,]  0.197320045
## [2,]  0.003309825
## [3,] -1.044415662
## [4,]  0.003463629
```

our statistic is

$$\frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 S_{j,j}}}$$

with $S_{j,j}$ jth term of the diagonal of $(X^T X)^{-1}$ et $\hat{\sigma}^2 = \frac{||\hat{\epsilon}||^2}{n-p}$ the condition for us to reject our hypothesis H_0) is that

$$\frac{|\hat{\beta}_j|}{\sqrt{\hat{\sigma}^2 S_{j,j}}} > t_{n-p}(1 - \frac{\alpha}{2})$$

or

$$p - value < \alpha$$

the value of the statistics

```r
d <- inv(t(X)%*%X)
epsilon.ice <- icedata$cons  - as.vector(fitted(mod.ice))
n <- nrow(icedata)
p <- ncol(icedata[,-1])
sigma2_est <- crossprod(epsilon.ice)/(n-p)
stat <- sapply(1:length(beta), function(i){
  beta[i]/(sqrt(sigma2_est*d[i,i]))
})
stat
```

```
## [1]  0.7441408  2.8793073 -1.2756059  7.9218457
```

```r
alpha <- 0.05
qt(p = 1-alpha/2, df = n-p)
```

```
## [1] 2.051831
```

the associated-pvalue the p-value or probability value is, for a given statistical model, the probability that, when the null hypothesis is true, the statistical summary (such as the sample mean difference between two groups) would be equal to, or more extreme than, the actual observed results.

```r
pt(q = stat, df = n-p, lower.tail = F) <= alpha/2
```

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

From the results below we can see that the second and the fourth variable “income” and “temps” are very significant to our variable target.

```r
confint(object = mod.ice, level = 0.95)
```

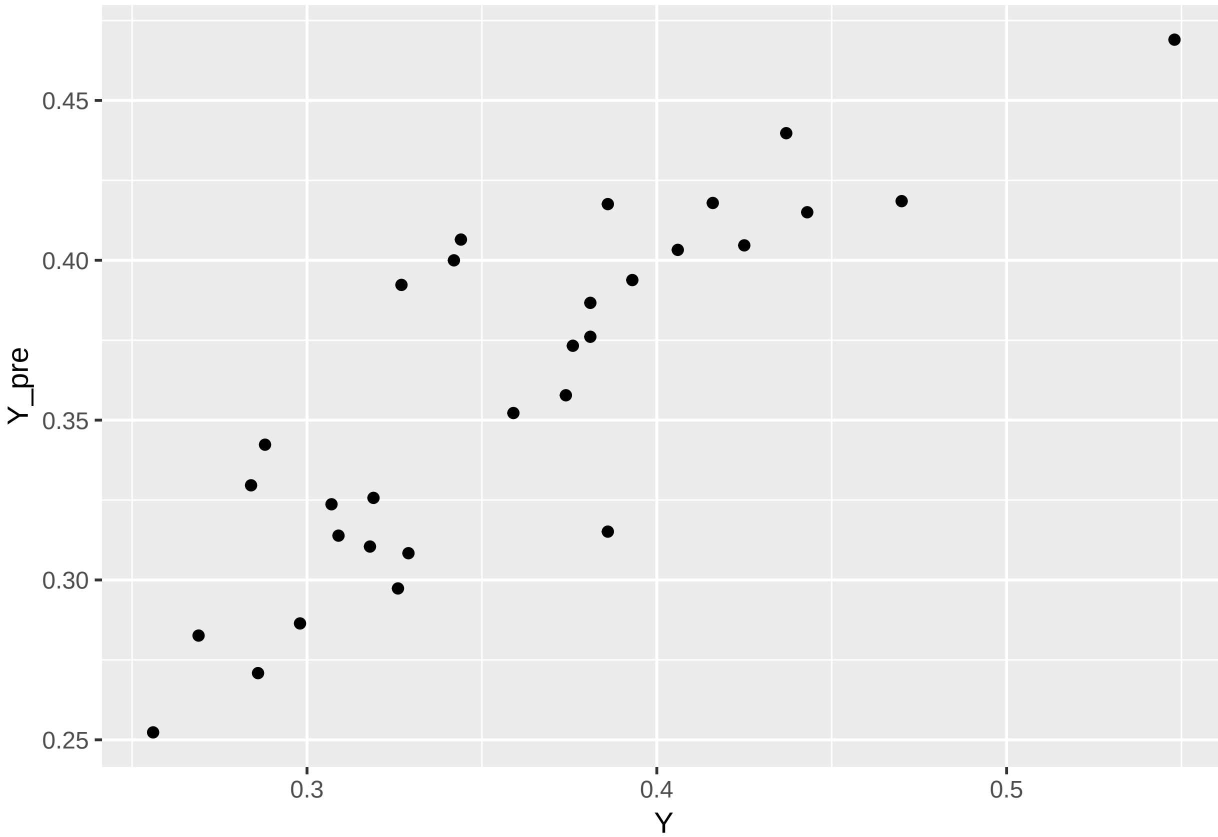
```
##                     2.5 %      97.5 %
## (Intercept) -0.3581221927 0.752752337
## income       0.0008998752 0.005715646
## price       -2.7594600283 0.670632044
## temp         0.0025425950 0.004374264
```

```r
confint(object = mod.ice, level = 0.99)
```

```
##                     0.5 %      99.5 %
## (Intercept) -5.535385e-01 0.948168633
## income       5.272283e-05 0.006562798
```

```
## price       -3.362855e+00 1.274026823
## temp         2.220382e-03 0.004696477
```

```r
library(ggplot2)
Y_pre <- fitted(object = mod.ice)
ggplot(data = data.frame(Y,Y_pre),aes(Y,Y_pre)) +
  geom_point()
```




```r
predict(object = mod.ice, level = "confidence")
```

```
##         1         2         3         4         5         6         7
## 0.3151242 0.3577755 0.3938221 0.4046732 0.4032559 0.4064819 0.3923018
##         8         9        10        11        12        13        14
## 0.3423158 0.2826049 0.2523278 0.2708627 0.2864021 0.3083716 0.3104496
##        15        16        17        18        19        20        21
## 0.3760779 0.3866857 0.4185069 0.4150247 0.4175791 0.3999856 0.3256767
##        22        23        24        25        26        27        28
## 0.3236806 0.3296116 0.2973486 0.3138636 0.3522185 0.3732705 0.4179287
##        29        30
## 0.4397578 0.4690144
```

```r
Y_pre
```

```
##         1         2         3         4         5         6         7
## 0.3151242 0.3577755 0.3938221 0.4046732 0.4032559 0.4064819 0.3923018
##         8         9        10        11        12        13        14
## 0.3423158 0.2826049 0.2523278 0.2708627 0.2864021 0.3083716 0.3104496
##        15        16        17        18        19        20        21
## 0.3760779 0.3866857 0.4185069 0.4150247 0.4175791 0.3999856 0.3256767
```

```
##         22         23         24         25         26         27         28
## 0.3236806 0.3296116 0.2973486 0.3138636 0.3522185 0.3732705 0.4179287
##         29         30
## 0.4397578 0.4690144
```

RMSE

$$\sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} = \sqrt{\hat{\sigma}^2}$$

```r
library(hydroGOF)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
rmse(as.vector(Y),as.vector(Y_pre))
```

```
## [1] 0.03428938
```

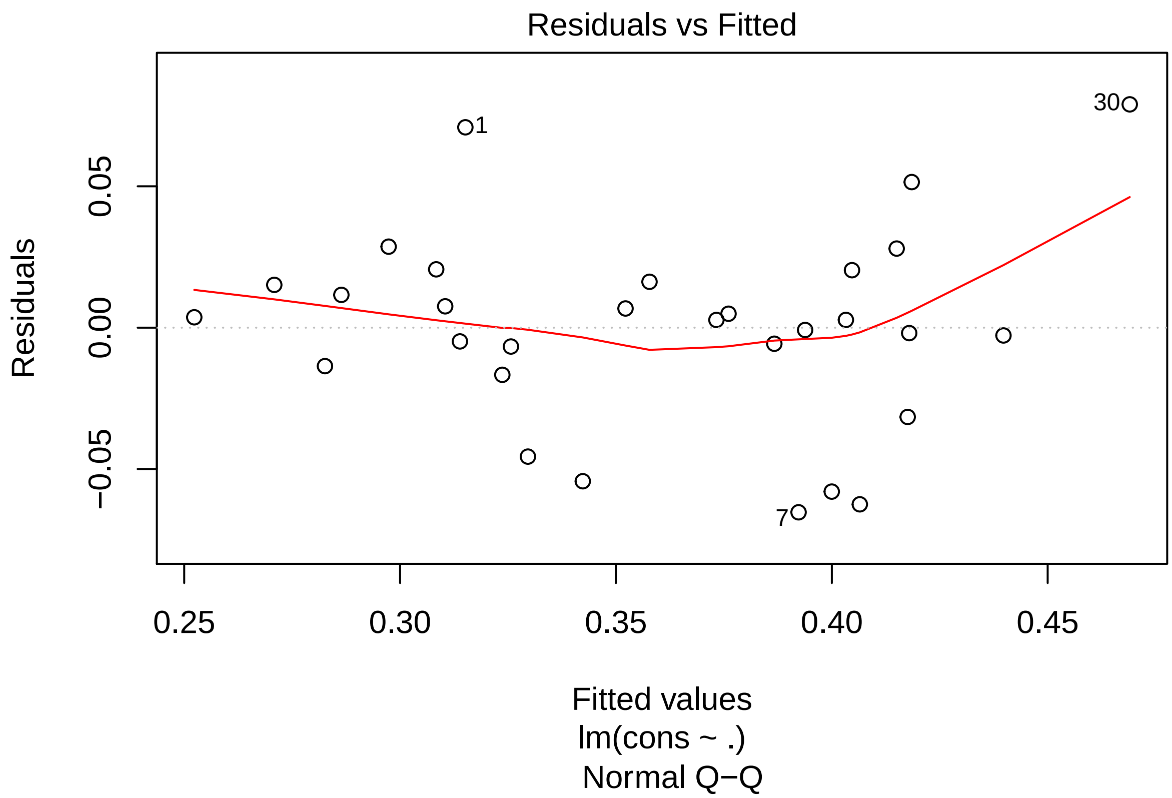
we are going to calculate the non-biased variance of $\epsilon_1, \epsilon_2, ..., \epsilon_n$, which the equation is

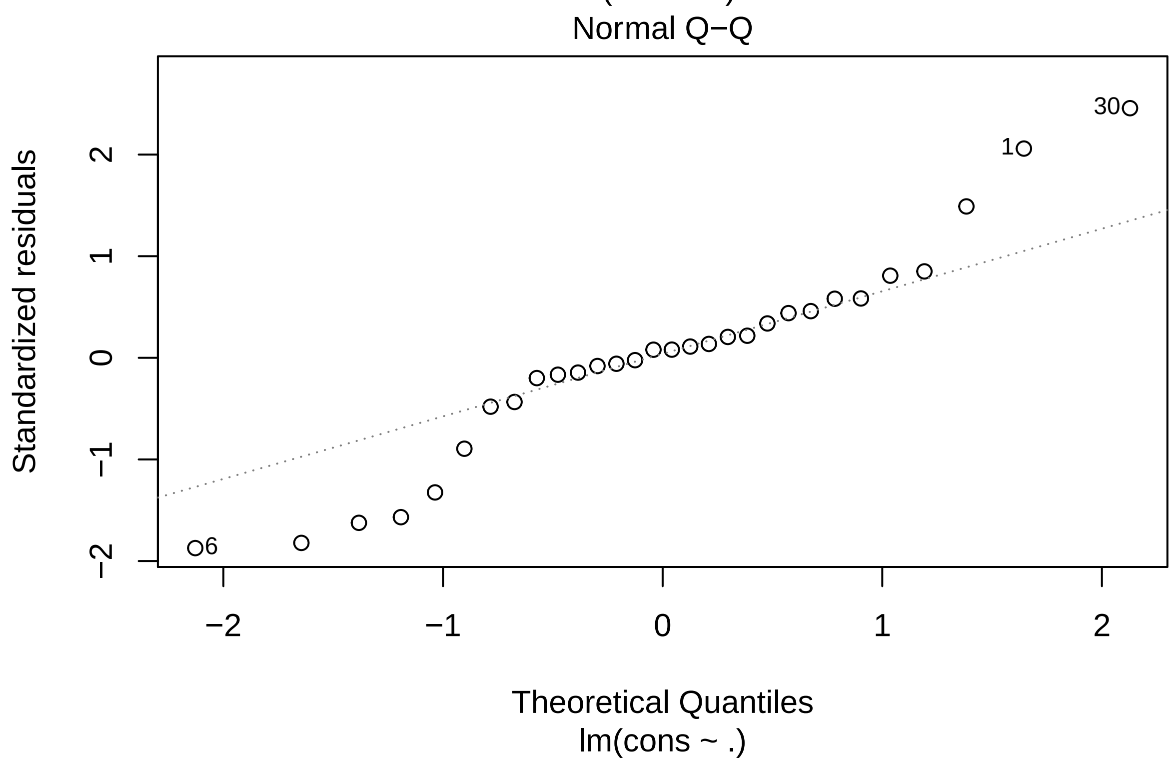
$$\mathcal{D}_{non\ biased}[\epsilon_1, \epsilon_2, ..., \epsilon_n] = \frac{1}{n-1}\sum_{i=1}^{n}(\epsilon_i - \bar{\epsilon})^2$$

```r
var(epsilon.ice)
```

```
## [1] 0.001216305
```

```r
plot(mod.ice, which = 1:2)
```





From the qq-plot we can see that those predicted values and the obseved targets only fit in the middle part, and at the same time it has a very obvious symmetry tendency for the head and tail, which infers that we may have ignored some rules between our values.

```
shapiro.test(epsilon.ice)
```

```
##
```

Nom	Source	Pays	Date	Horaire	Actual	Forecast	Previous	Importance
"CPM"	DailyFX	USD	2011-01-31	14:45:00.000	68.8	64.5	66.8	1
"GDP"	DailyFX	EUR	2015-03-06	10:00:00.000	0.3	0.3	0.3	2

```
##  Shapiro-Wilk normality test
##
## data:  epsilon.ice
## W = 0.9444, p-value = 0.1195
```

\documentclass[xcolor=table]{beamer}