Exercice_pairs_Sujet_4

Vivien Gaujoux

4/20/2020

Exercice évalué par les pairs

Importation et tri des données

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
liglab2 <- read_table("http://mescal.imag.fr/membres/arnaud.legrand/teaching/2014/RICM4_EP_ping/liglab2
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character()
## )
```

```
liglab2
```

```
## # A tibble: 44,413 x 2
##    X1               X2
##    <chr>            <chr>
##  1 [1421761682.052~ 665 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=1~
##  2 [1421761682.277~ 1373 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
##  3 [1421761682.502~ 262 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=1~
##  4 [1421761682.729~ 1107 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
##  5 [1421761682.934~ 1128 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
##  6 [1421761683.160~ 489 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=1~
##  7 [1421761683.443~ 1759 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
```

```
##  8 [1421761683.672~ 1146 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
##  9 [1421761683.899~ 884 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=1~
## 10 [1421761684.122~ 1422 bytes from lig-publig.imag.fr (129.88.11.7): icmp_seq=~
## # ... with 44,403 more rows
```

```
names(liglab2)[names(liglab2) == "X1"] <- "date"
liglab2<-separate(liglab2, X2, sep = " ", extra = "merge",
into = c("taille", "a_virer"))
liglab2<-separate(liglab2, a_virer, sep = "time=", extra = "merge",
into = c("a_virer", "time"))
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 377 rows [13,
## 117, 315, 509, 591, 648, 1088, 1092, 1132, 1154, 1351, 1543, 1692, 1766, 1958,
## 2100, 2183, 2210, 2456, 2479, ...].
```

```
liglab2$a_virer <- NULL
```

```
liglab2$time = substr(liglab2$time,1,nchar(liglab2$time)-3)
liglab2$date = substr(liglab2$date,2,nchar(liglab2$date)-1)
liglab2$time <- as.numeric(as.character(liglab2$time))
liglab2$taille <- as.numeric(as.character(liglab2$taille))
lignes_na = apply(liglab2, 1, function(x) any(is.na(x)))
```

```
lignes_na = apply(liglab2, 1, function(x) any(is.na(x)))
liglab2[lignes_na,]
```

```
## # A tibble: 377 x 3
##    date              taille  time
##    <chr>              <dbl> <dbl>
##  1 1421761684.770828     21    NA
##  2 1421761706.146320      9    NA
##  3 1421761748.647974      9    NA
##  4 1421761788.500516      9    NA
##  5 1421761805.331125     18    NA
##  6 1421761817.308839     18    NA
##  7 1421761907.694059     18    NA
##  8 1421761908.512826     13    NA
##  9 1421761916.713419     20    NA
## 10 1421761921.229704     21    NA
## # ... with 367 more rows
```

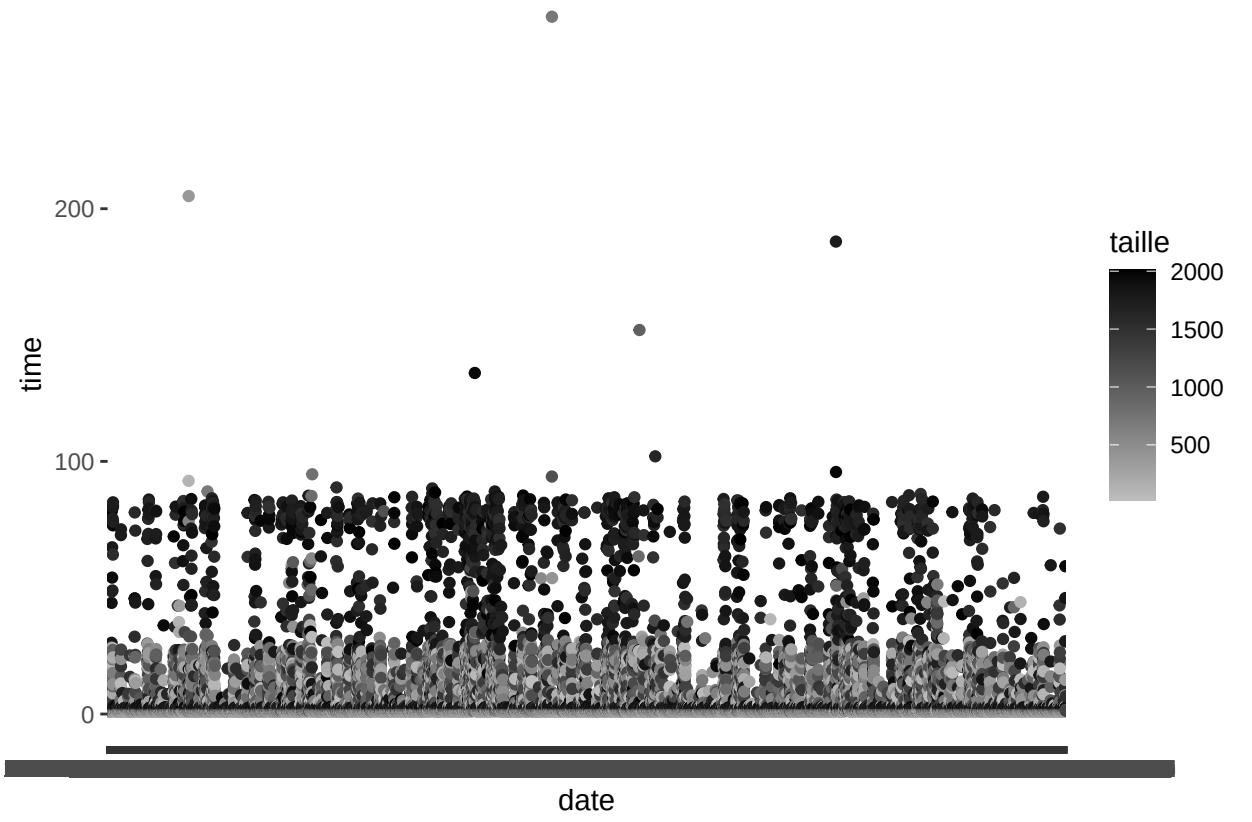
```
liglab2 <- liglab2 %>% drop_na("time")
liglab2
```

```
## # A tibble: 44,036 x 3
##    date              taille  time
##    <chr>              <dbl> <dbl>
##  1 1421761682.052172    665 22.5
##  2 1421761682.277315   1373 21.2
##  3 1421761682.502054    262 21.2
##  4 1421761682.729257   1107 23.3
```

```
##  5 1421761682.934648   1128  1.41
##  6 1421761683.160397    489 21.9
##  7 1421761683.443055   1759 78.7
##  8 1421761683.672157   1146 25.1
##  9 1421761683.899933    884 24
## 10 1421761684.122687   1422 19.5
## # ... with 44,026 more rows
```

1) Représentation graphique :

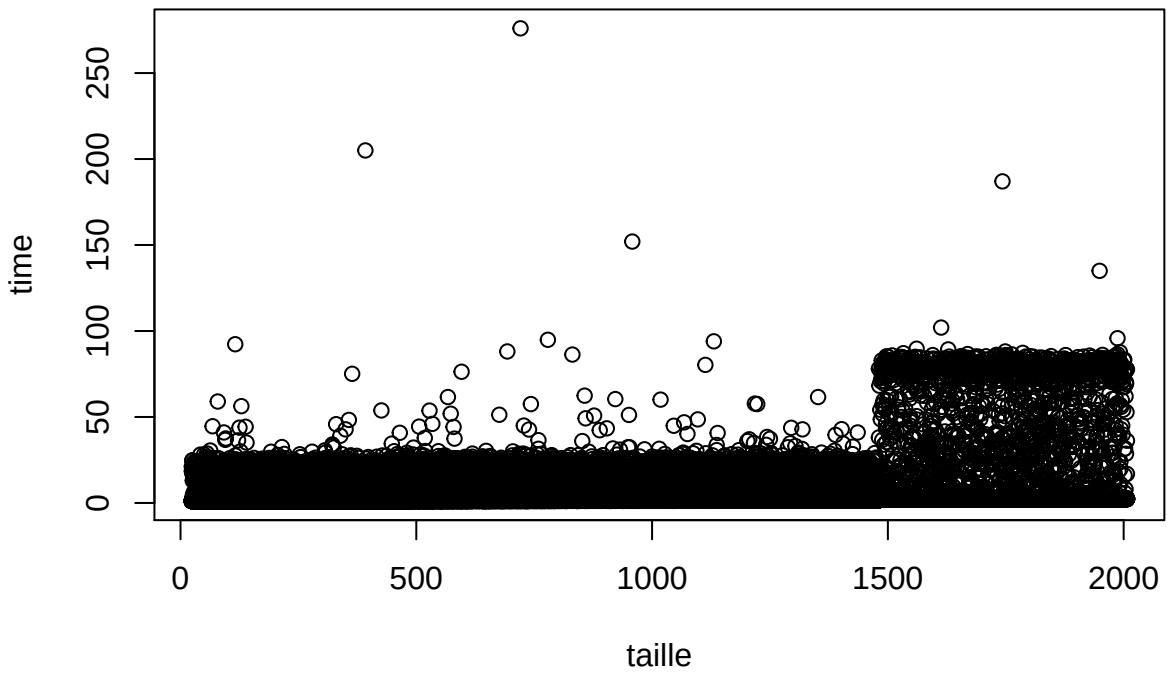
```
ggplot(liglab2) +
geom_point(aes(x = date, y = time, color = taille)) +
scale_color_gradient(low = 'gray', high = 'gray0')
```



Il semble que la taille des messages ne soit pas la seule en cause dans le temps de transmission.

2) Représentation graphique : temps de transmission en fonction de la taille des messages

```
with(liglab2, plot(taille, time, type = "p"))
```



On peut voir la rupture à partir de 1500 octets environ.

```
donnees_low <- subset(liglab2, taille <= 1500)
donnees_high <-subset(liglab2, taille > 1500)
donnees_high
```

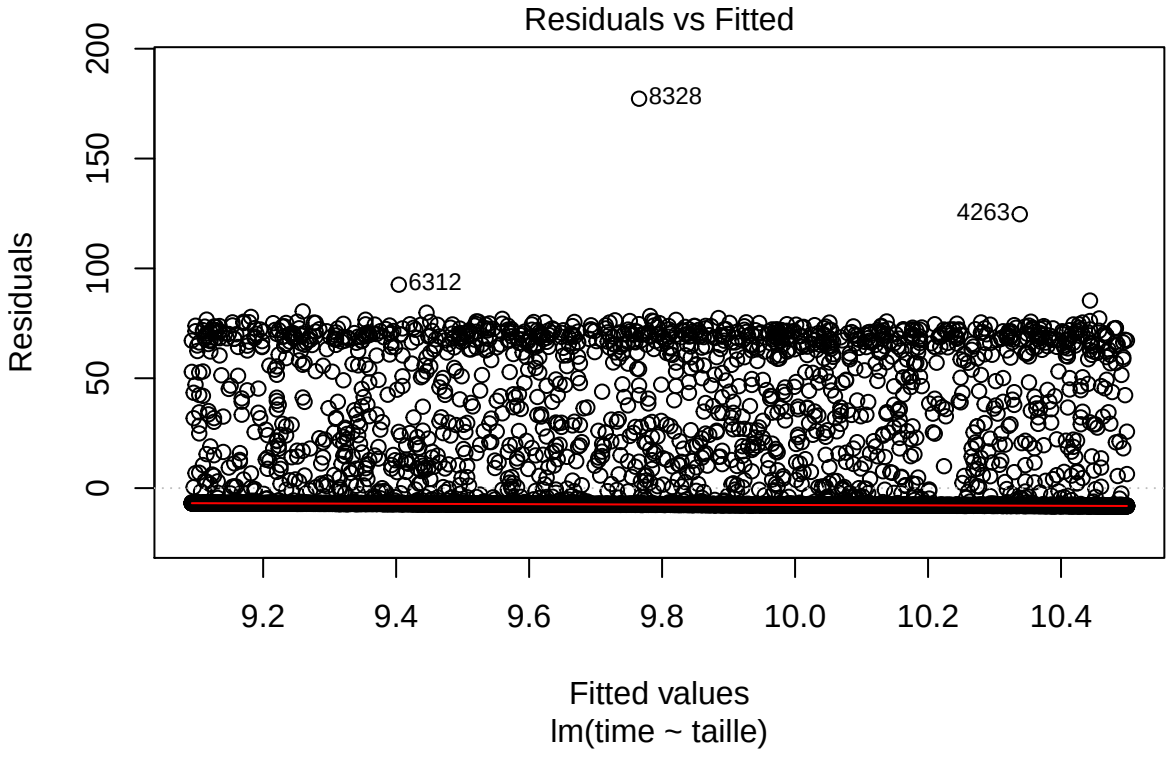
```
## # A tibble: 10,958 x 3
##    date              taille  time
##    <chr>              <dbl> <dbl>
##  1 1421761683.443055   1759 78.7
##  2 1421761685.620117   1843  2.31
##  3 1421761686.234464   1511  2.18
##  4 1421761687.463275   1510  2.17
##  5 1421761687.874230   1966  2.2
##  6 1421761688.694652   1518  2.19
##  7 1421761690.335289   1732  2.29
##  8 1421761691.359824   1520  2.1
##  9 1421761691.974735   1509  2.23
## 10 1421761692.385087   1539  2.17
## # ... with 10,948 more rows
```

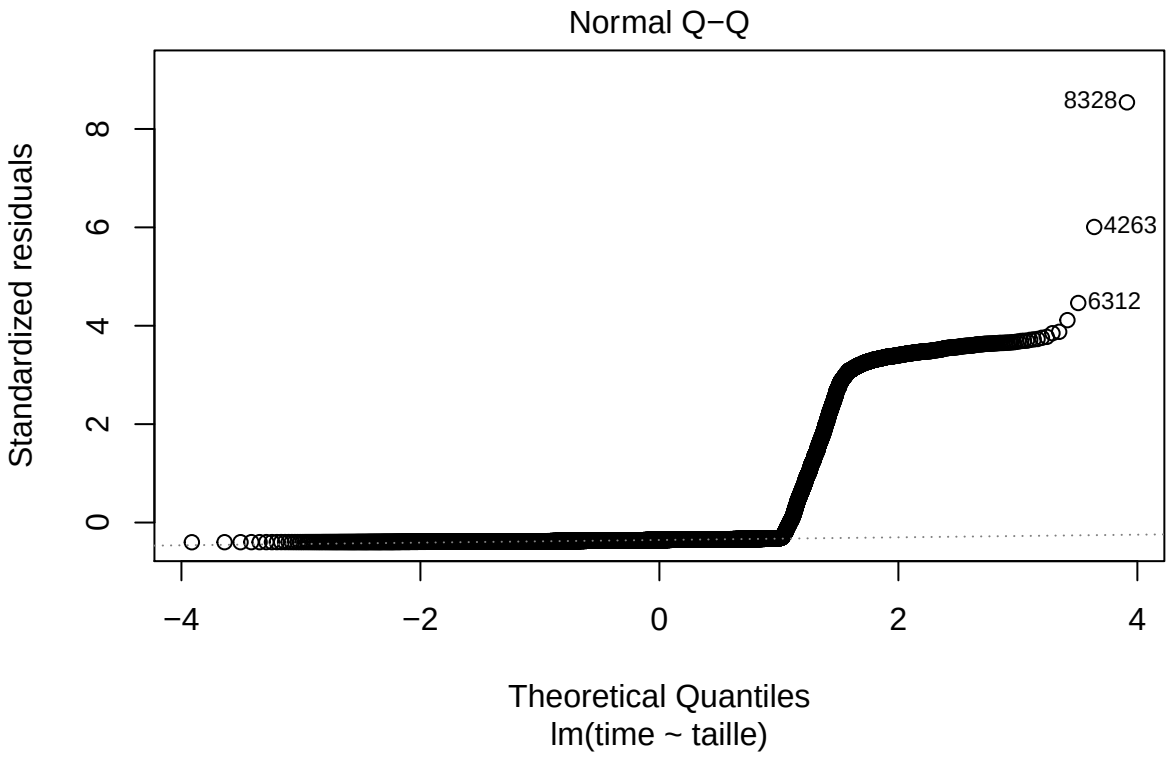
3) Régression linéaire :

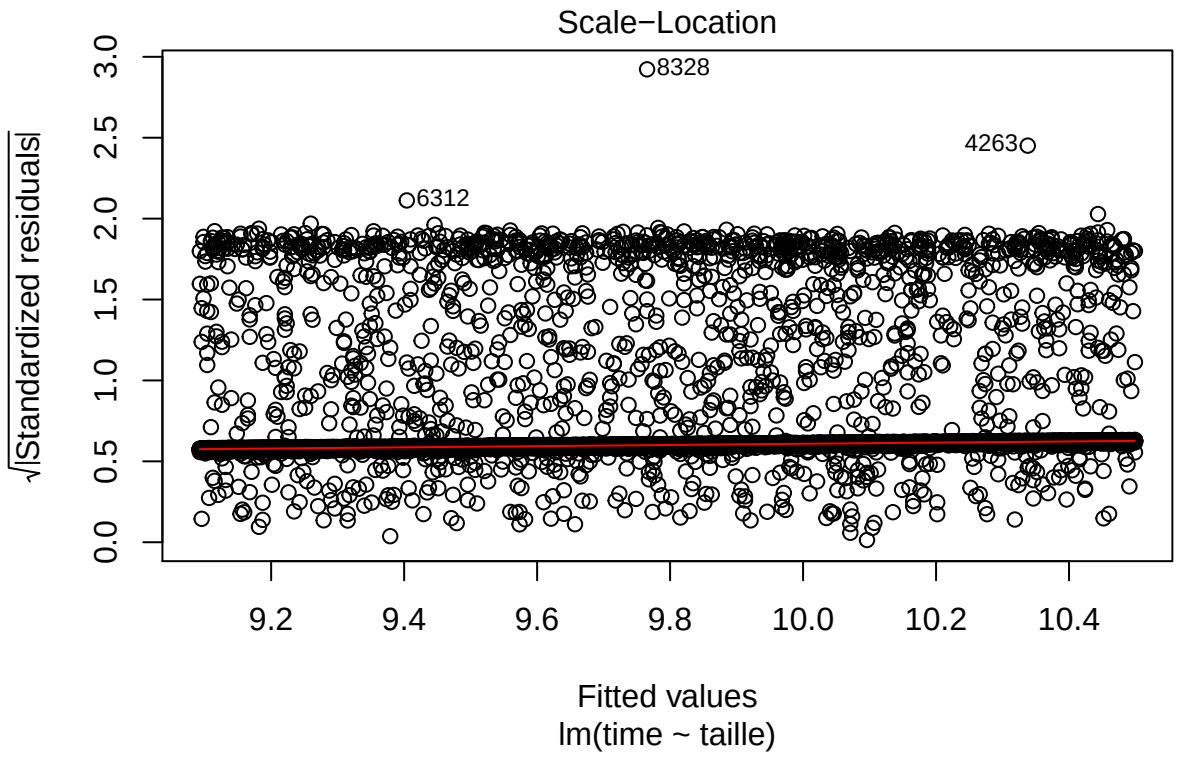
```r
high_time.lm <- lm(time~taille, data = donnees_high)
summary(high_time.lm)
```

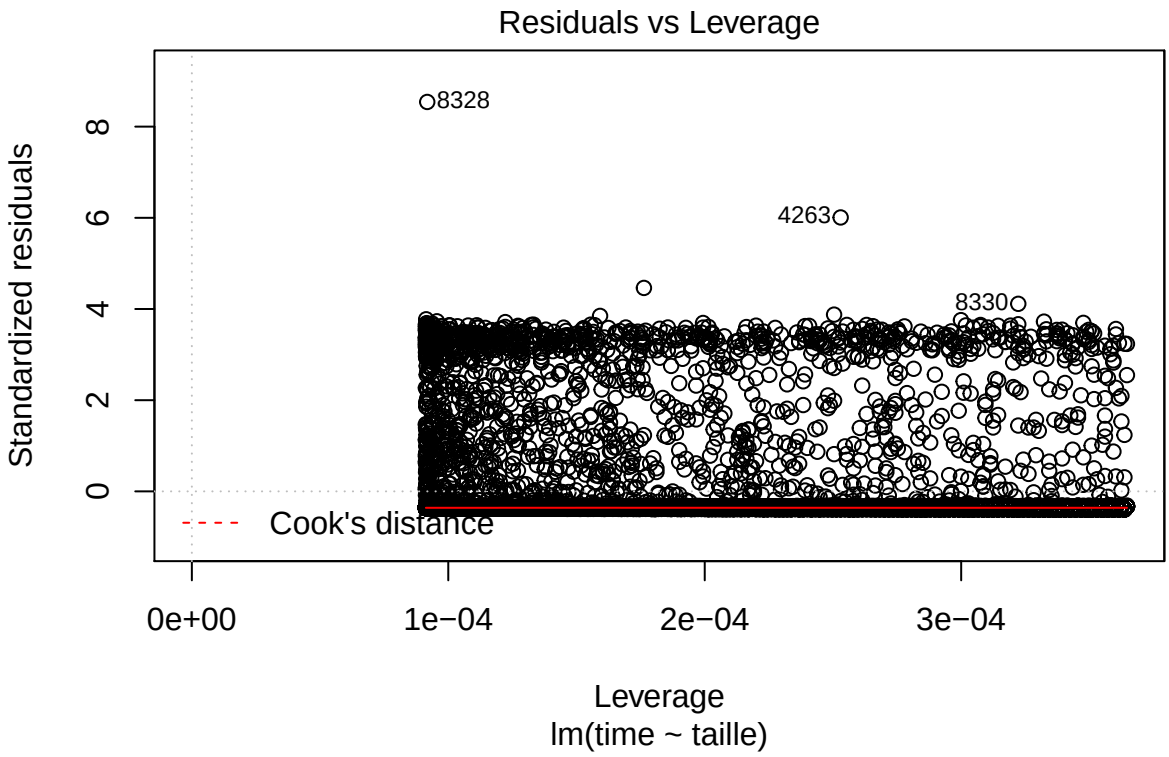
```
##
## Call:
## lm(formula = time ~ taille, data = donnees_high)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -8.309  -7.743  -7.356  -6.986 177.235
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.920748   2.385179   2.063   0.0391 *
## taille      0.002779   0.001355   2.051   0.0403 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.75 on 10956 degrees of freedom
## Multiple R-squared:  0.0003839,  Adjusted R-squared:  0.0002927
## F-statistic: 4.208 on 1 and 10956 DF,  p-value: 0.04026
```

```r
plot(high_time.lm)
```




Normal Q−Q
Standardized residuals
0
2
4
6
8
8328
4263
6312
−4
−2
0
2
4
Theoretical Quantiles
lm(time ~ taille)

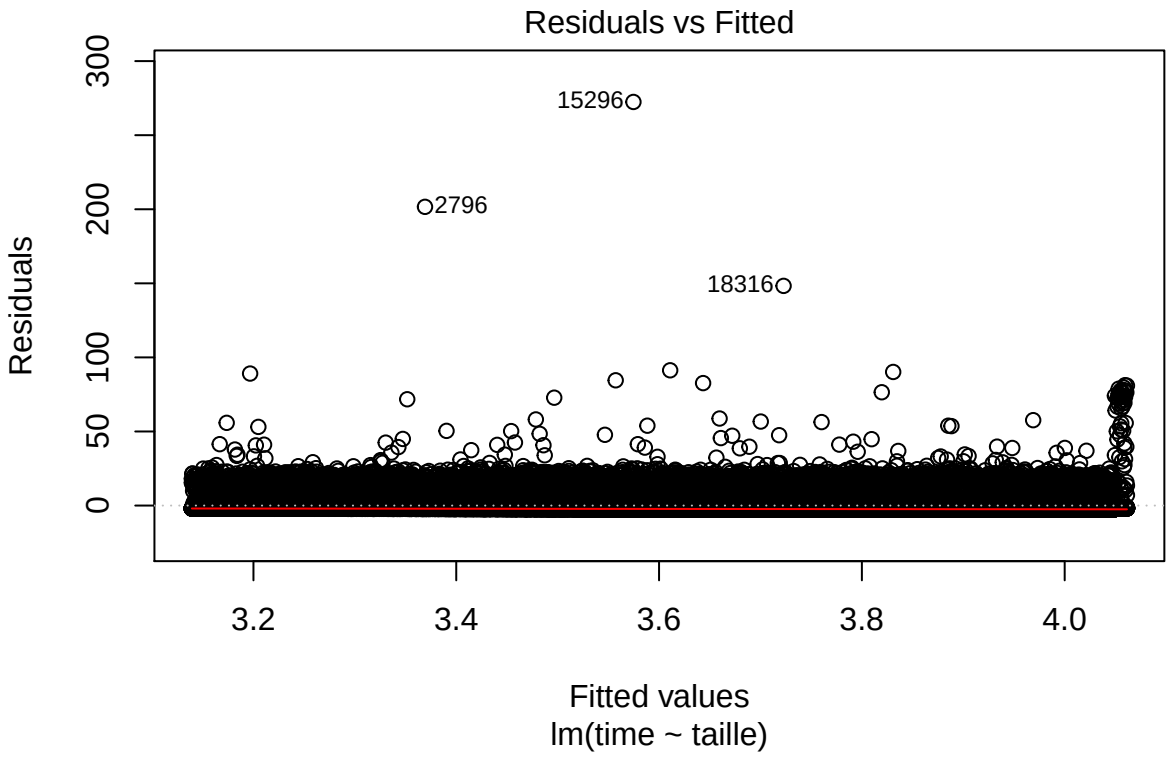

Scale−Location
8328
4263
6312
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
9.2
9.4
9.6
9.8
10.0
10.2
10.4
Fitted values
lm(time ~ taille)

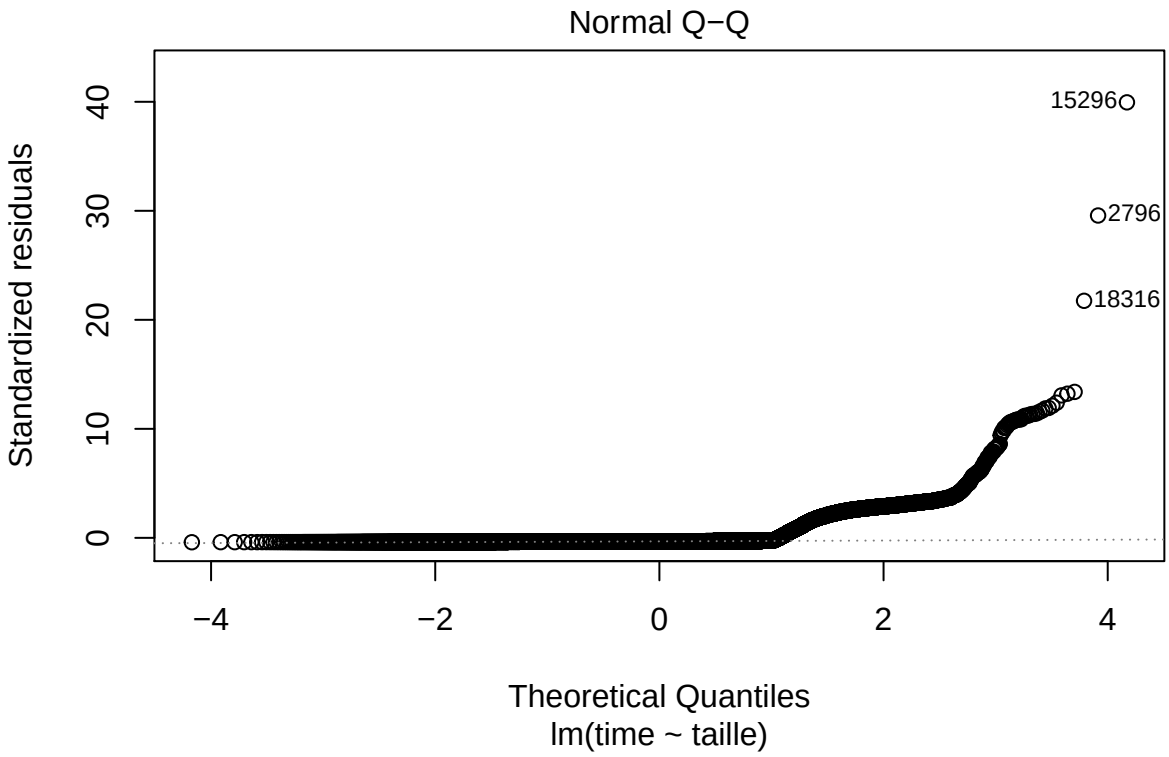


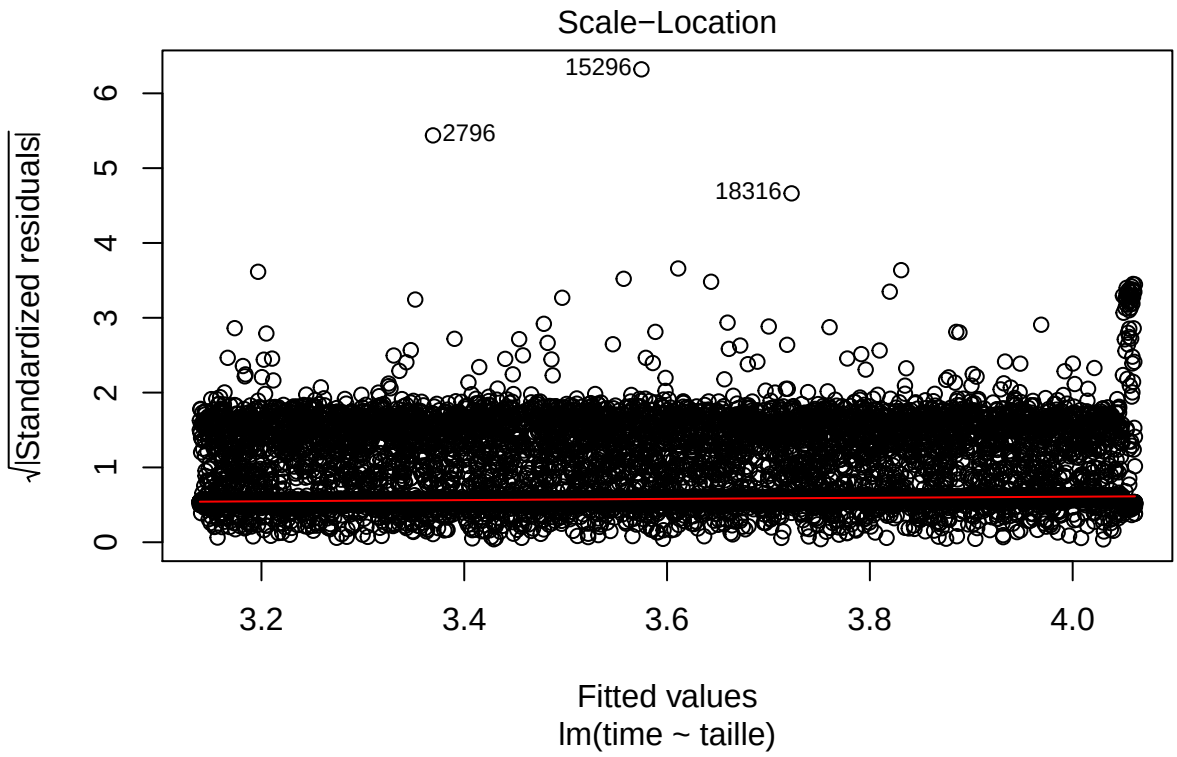

```
low_time.lm <- lm(time~taille, data = donnees_low)
summary(low_time.lm)
```

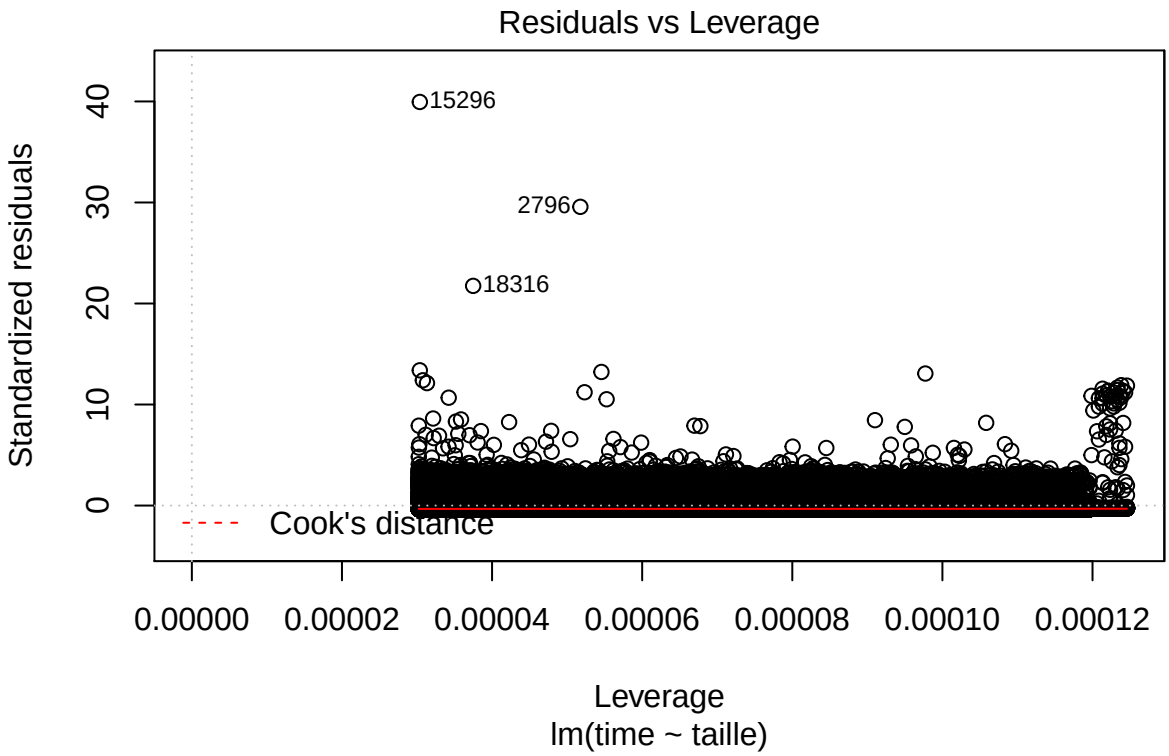
```
##
## Call:
## lm(formula = time ~ taille, data = donnees_low)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -2.753  -2.379  -2.207  -2.030 272.425
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.124e+00  7.614e-02  41.031  < 2e-16 ***
## taille      6.248e-04  8.834e-05   7.072 1.55e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.819 on 33076 degrees of freedom
## Multiple R-squared:  0.00151,    Adjusted R-squared:  0.00148
## F-statistic: 50.02 on 1 and 33076 DF,  p-value: 1.555e-12
```

```
plot(low_time.lm)
```


Residuals vs Fitted
Residuals
300
200
100
50
0
15296
2796
18316
3.2
3.4
3.6
3.8
4.0
Fitted values
lm(time ~ taille)


Normal Q−Q
Standardized residuals
0
10
20
30
40
15296
2796
18316
−4
−2
0
2
4
Theoretical Quantiles
lm(time ~ taille)


Scale−Location
15296
2796
18316
√|Standardized residuals|
0
1
2
3
4
5
6
3.2
3.4
3.6
3.8
4.0
Fitted values
lm(time ~ taille)



La valeur α et donc la latence des données de taille élevée est de 4.920748 et la valeur β est de 0.002779. La valeur α et donc la latence des données de taille basse est de 3.124 et la valeur β est de 0.0006248.

Ces valeurs indiquent une capacité respectivement de $1/0.002779$ et $1/0.0006248$, soit 359.84 pour les données de taille élevée et 1600.051 pour les données de taille basse.