Loopring

PART 1

```
raw_data <- read.csv(file = 'networkloopringTX.txt', header = F, sep = ' ')
colnames(raw_data) <- c("Buyer", "Seller", "Timestamp", "TokenAmount")
message ('The number of rows are: ', nrow(raw_data))
```

```
## The number of rows are: 204222
```

```
summary(raw_data)
```

```
##      Buyer             Seller          Timestamp
##  Min.   :      4   Min.   :      4   Min.   :1.502e+09
##  1st Qu.:  85250   1st Qu.: 104502   1st Qu.:1.506e+09
##  Median :1878420   Median : 320500   Median :1.515e+09
##  Mean   :2358534   Mean   :1770397   Mean   :1.513e+09
##  3rd Qu.:4849192   3rd Qu.:3703596   3rd Qu.:1.519e+09
##  Max.   :4876603   Max.   :4876611   Max.   :1.526e+09
##   TokenAmount
##  Min.   :1.000e+00
##  1st Qu.:8.353e+20
##  Median :4.039e+21
##  Mean   :2.835e+71
##  3rd Qu.:1.281e+22
##  Max.   :5.790e+76
```

```
Total_circulation_amount = 828954240 * 10^18
outliers <- subset(raw_data, TokenAmount > Total_circulation_amount)
message ('The number of outliers in the dataset are: ', nrow(outliers))
```

```
## The number of outliers in the dataset are: 2
```

```
preprocessed_data <- subset(raw_data, TokenAmount <= Total_circulation_amount)
summary(preprocessed_data)
```

```
##      Buyer             Seller          Timestamp        
##  Min.   :      4   Min.   :      4   Min.   :1.502e+09  
##  1st Qu.:  85250   1st Qu.: 104502   1st Qu.:1.506e+09  
##  Median :1878420   Median : 320508   Median :1.515e+09  
##  Mean   :2358548   Mean   :1770415   Mean   :1.513e+09  
##  3rd Qu.:4849192   3rd Qu.:3703598   3rd Qu.:1.519e+09  
##  Max.   :4876603   Max.   :4876611   Max.   :1.526e+09  
##   TokenAmount       
##  Min.   :1.000e+00  
##  1st Qu.:8.353e+20  
##  Median :4.039e+21  
##  Mean   :4.181e+22  
##  3rd Qu.:1.281e+22  
##  Max.   :4.185e+26  
```

```
library(plyr)
Buyer_Seller_Pair_Frequencies <- ddply(preprocessed_data, .(preprocessed_data$Buyer, preprocesse
d_data$Seller), nrow)
names(Buyer_Seller_Pair_Frequencies) <- c("Buyer", "Seller", "Frequency")
Buyer_Seller_Pair_Frequencies
```

Buyer <int>	Seller <int>	Frequency <int>
82	2964307	1
6	2964307	1
40002	3274516	1
82	1815762	42
44	4848203	1
3078280	3300522	1
222770	4848204	1
5	1991385	1
3300522	5	1
5	305723	1

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
summary(Buyer_Seller_Pair_Frequencies)
```

```
##      Buyer              Seller           Frequency
##  Min.   :      4   Min.   :      4   Min.   :   1.000
##  1st Qu.:  89826   1st Qu.: 289397   1st Qu.:   1.000
##  Median :1871198   Median :1935441   Median :   1.000
##  Mean   :2262788   Mean   :2345840   Mean   :   1.854
##  3rd Qu.:4849192   3rd Qu.:4852638   3rd Qu.:   1.000
##  Max.   :4876603   Max.   :4876611   Max.   :1469.000
```

```
message ('Variance: ', var(Buyer_Seller_Pair_Frequencies$Frequency))
```

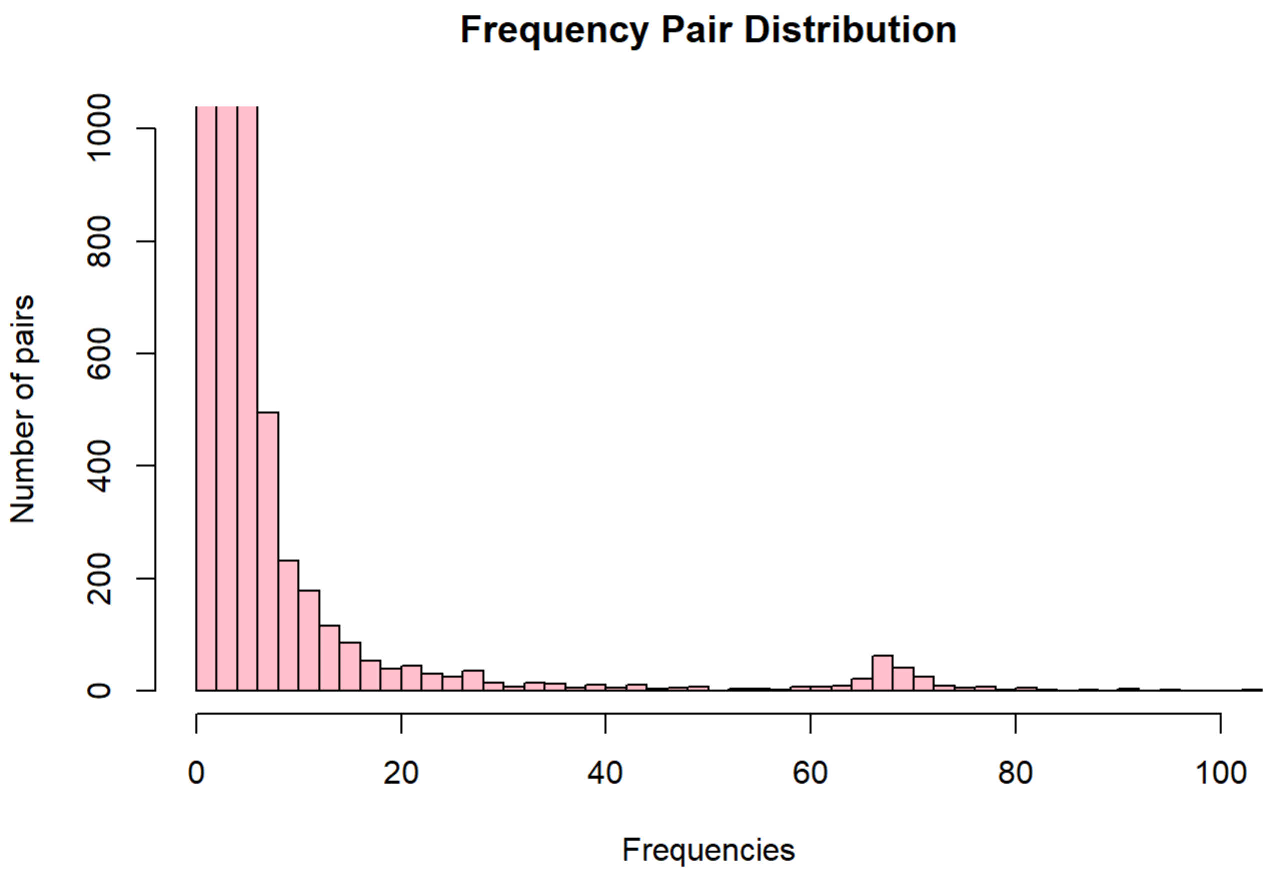
```
## Variance: 215.627814313744
```

```
message ('Standard Deviation: ', sd(Buyer_Seller_Pair_Frequencies$Frequency))
```

```
## Standard Deviation: 14.6842709833939
```

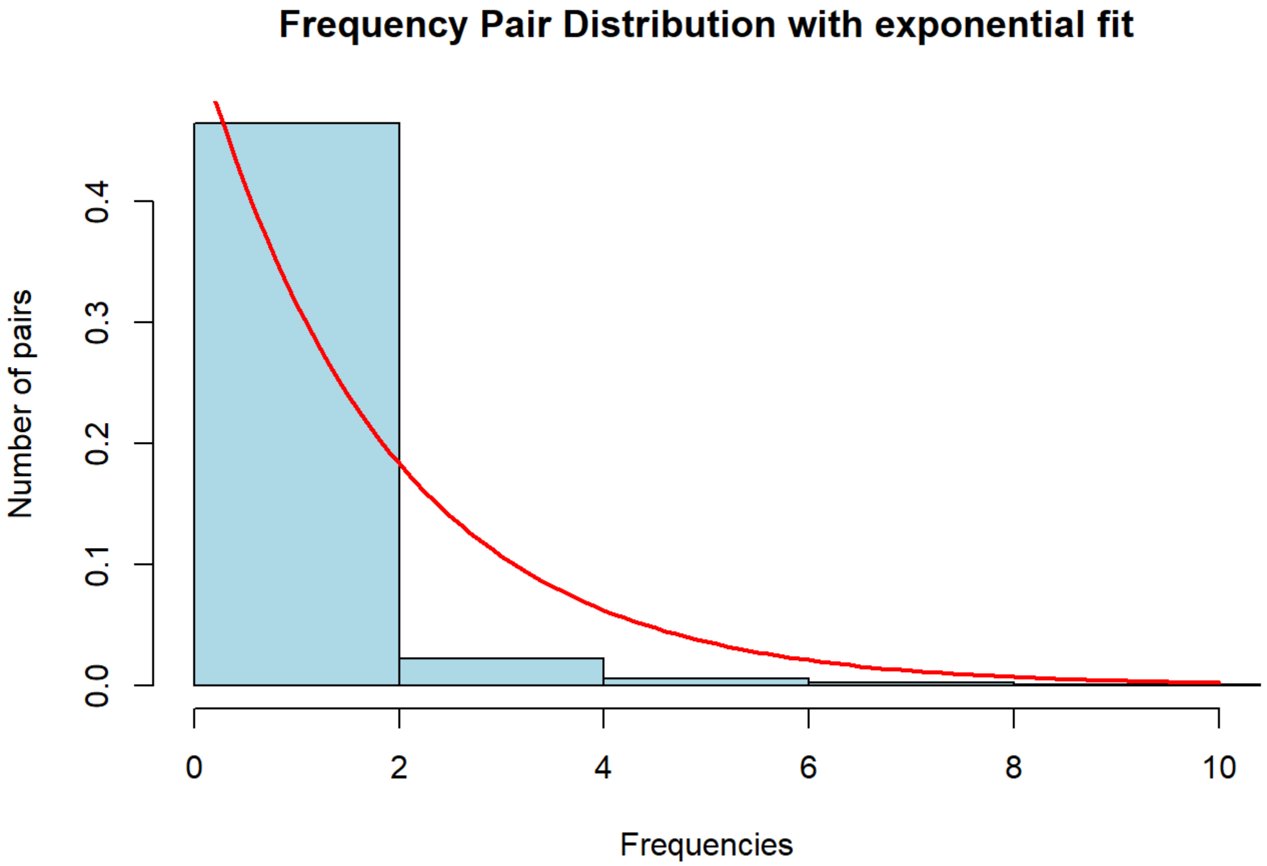
```
freq <- Buyer_Seller_Pair_Frequencies$Frequency
library(MASS)

h <- hist(freq, main = 'Frequency Pair Distribution',
     xlab = 'Frequencies', ylab = 'Number of pairs', col = 'pink',
     breaks = 1000, xlim = c(0, 100), ylim = c(0, 1000))
```



```
h <- hist(freq, main = 'Frequency Pair Distribution with exponential fit',
     xlab = 'Frequencies', ylab = 'Number of pairs', col = 'lightblue', freq = FALSE,
     breaks = 1000, xlim = c(0, quantile(freq, 0.99)))

fit <- fitdistr(freq, 'exponential')
curve(dexp(x, rate = fit$estimate), from = 0, col = 'red', add = TRUE, lwd = 2)
```

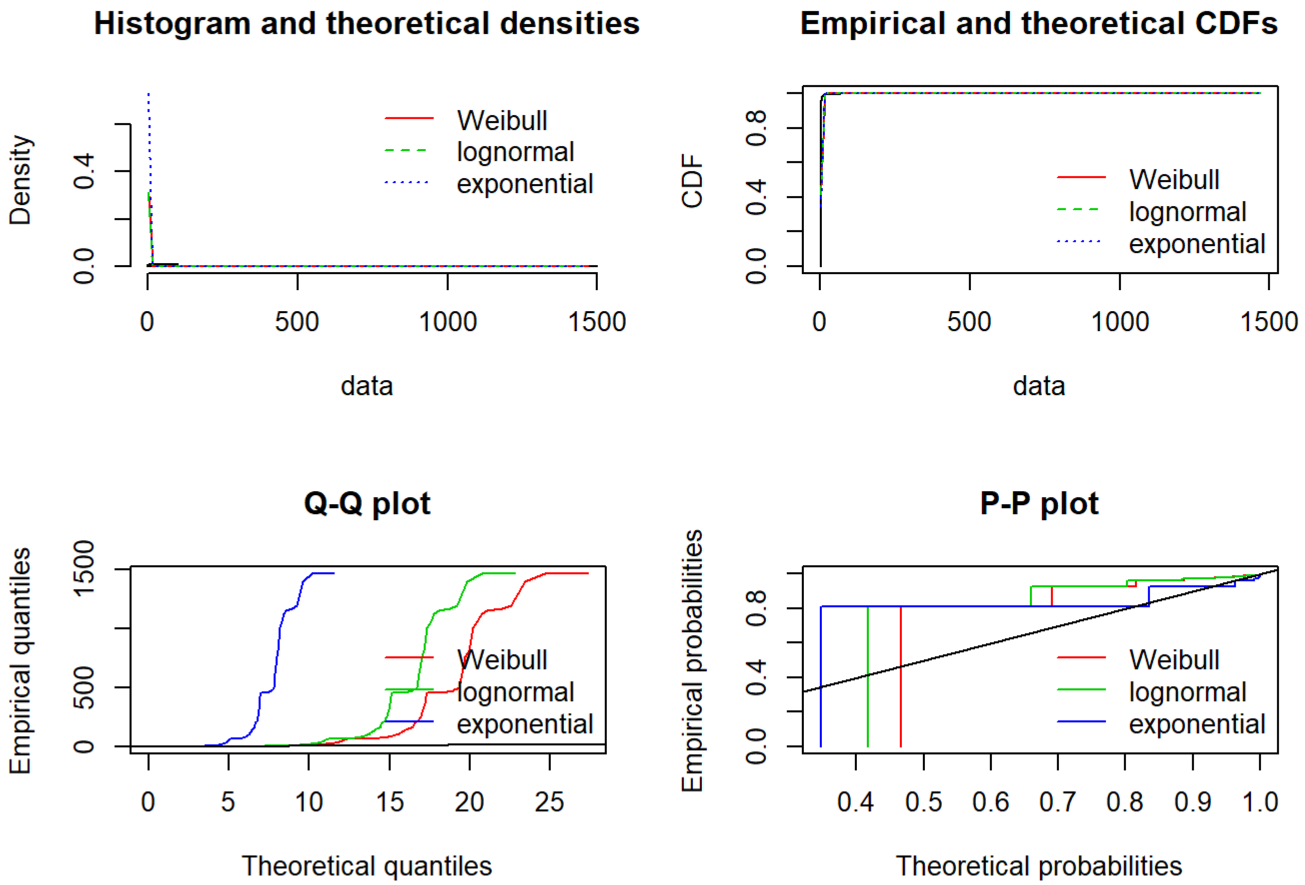


```
library(fitdistrplus)
```

```
## Loading required package: survival
```

```
fit_w  <- fitdist(freq, "weibull")
fit_ln <- fitdist(freq, "lnorm")
fit_ex <- fitdist(freq, "exp")

par(mfrow=c(2,2))
plot.legend <- c("Weibull", "lognormal", "exponential")
denscomp(list(fit_w, fit_ex, fit_ln), legendtext = plot.legend)
cdfcomp (list(fit_w, fit_ex, fit_ln), legendtext = plot.legend)
qqcomp  (list(fit_w, fit_ex, fit_ln), legendtext = plot.legend)
ppcomp  (list(fit_w, fit_ex, fit_ln), legendtext = plot.legend)
```



PART 2 BEGINS HERE

```
preprocessed_data$TokenAmount <- preprocessed_data$TokenAmount/10^18
Time <- as.Date(as.POSIXct(preprocessed_data$Timestamp, origin = '1970-01-01'))
preprocessed_data$Timestamp <- Time
preprocessed_data
```

	Buyer	Seller	Timestamp	TokenAmount
	<int>	<int>	<date>	<dbl>
1	82	2964307	2018-04-24	9.510000e+01
2	6	2964307	2018-04-24	1.870149e+04
3	40002	3274516	2018-04-24	1.839461e+03
4	82	1815762	2018-04-24	8.389100e+03
5	44	4848203	2018-04-24	1.001000e+02
6	3078280	3300522	2018-04-24	2.619100e+04
7	222770	4848204	2018-04-24	5.000000e+00
8	5	1991385	2018-04-24	4.221000e+02
9	3300522	5	2018-04-24	2.619100e+04

	Buyer	Seller	Timestamp	TokenAmount
	<int>	<int>	<date>	<dbl>
10	5	305723	2018-04-24	5.785160e+02

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
library("readxl")
my_data <- read_excel("Loopring_CoinMarketCap.xlsx")
colnames(my_data) <- c('Timestamp', 'Open', 'High', 'Low', 'Close', 'Volume', 'MarketCap')
my_data$Timestamp <- as.Date(my_data$Timestamp, "%d%B%Y")
```

```
## Warning in as.POSIXlt.POSIXct(x, tz = tz): unknown timezone '%d%B%Y'
```

```
my_data$MarketCap <- as.double(my_data$MarketCap)
```

```
## Warning: NAs introduced by coercion
```

```
my_data
```

Timestamp	Open	High	Low	Close	Volume	MarketCap
<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2019-04-23	0.067044	0.068005	0.062899	0.063292	13938070	52466100
2019-04-22	0.068028	0.069217	0.065807	0.067036	15094051	55569829
2019-04-21	0.074267	0.074674	0.065887	0.068242	17061511	56569575
2019-04-20	0.076099	0.076315	0.073206	0.074280	17978822	61575008
2019-04-19	0.073970	0.078721	0.071970	0.076099	21377192	63082414
2019-04-18	0.071932	0.076205	0.071384	0.073996	21031814	61339614
2019-04-17	0.073975	0.074648	0.071543	0.071937	20962660	59632516
2019-04-16	0.071938	0.076779	0.070557	0.073891	20373289	61252102
2019-04-15	0.076367	0.076942	0.070921	0.071921	20047133	59619449
2019-04-14	0.075409	0.077245	0.074073	0.076367	27770512	63304611

1-10 of 602 rows Previous **1** 2 3 4 5 6 ... 61 Next

```
new_data <- preprocessed_data[order (- preprocessed_data$TokenAmount),]
new_data$Seller <- NULL
new_data
```

	Buyer	Timestamp	TokenAmount
	<int>	<date>	<dbl>

	Buyer <int>	Timestamp <date>	TokenAmount <dbl>
202883	4876467	2017-08-17	418522816.00
155913	1937577	2017-08-30	279015211.26
203563	4876467	2017-08-22	279015211.26
38324	4853589	2018-03-12	161067928.53
38319	299810	2018-03-12	140000000.00
121124	1936377	2017-11-13	140000000.00
202948	4861571	2017-08-18	139507605.00
202950	4861571	2017-08-18	139507605.00
120038	4865589	2017-11-07	138644070.02
197891	2336987	2017-09-12	100000000.00

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
joined_df <- join(new_data, my_data)
```

```
## Joining by: Timestamp
```

```
joined_df <- na.omit(joined_df)
joined_df
```

	Buyer <int>	Timestamp <date>	TokenAmount <dbl>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <dbl>	Market <
4	4853589	2018-03-12	161067928.53	0.376652	0.377374	0.338550	0.346355	3356110	198140
5	299810	2018-03-12	140000000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
6	1936377	2017-11-13	140000000.00	0.130328	0.148434	0.130328	0.142889	457430	40890
9	4865589	2017-11-07	138644070.02	0.156029	0.157926	0.147550	0.155454	463039	44486
10	2336987	2017-09-12	100000000.00	0.044543	0.052202	0.042257	0.050735	316986	35389
11	4865590	2017-10-27	90000000.00	0.139625	0.149028	0.136067	0.147354	246288	42168
12	4853608	2018-03-12	89980000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
13	298944	2017-09-13	70789635.00	0.050891	0.050891	0.043732	0.046747	177864	32608
17	4853600	2018-03-12	54001283.71	0.376652	0.377374	0.338550	0.346355	3356110	198140
19	298944	2017-09-09	50000000.00	0.047309	0.049558	0.046045	0.047638	2811530	33229

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
joined_df$percentage <- (joined_df$TokenAmount/joined_df$MarketCap)*100
joined_df
```

	Buyer	Timestamp	TokenAmount	Open	High	Low	Close	Volume	Market
	<int>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<c
4	4853589	2018-03-12	161067928.53	0.376652	0.377374	0.338550	0.346355	3356110	198140
5	299810	2018-03-12	140000000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
6	1936377	2017-11-13	140000000.00	0.130328	0.148434	0.130328	0.142889	457430	40890
9	4865589	2017-11-07	138644070.02	0.156029	0.157926	0.147550	0.155454	463039	44486
10	2336987	2017-09-12	100000000.00	0.044543	0.052202	0.042257	0.050735	316986	35389
11	4865590	2017-10-27	90000000.00	0.139625	0.149028	0.136067	0.147354	246288	42168
12	4853608	2018-03-12	89980000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
13	298944	2017-09-13	70789635.00	0.050891	0.050891	0.043732	0.046747	177864	32608
17	4853600	2018-03-12	54001283.71	0.376652	0.377374	0.338550	0.346355	3356110	198140
19	298944	2017-09-09	50000000.00	0.047309	0.049558	0.046045	0.047638	2811530	33229

1-10 of 10,000 rows | 1-10 of 11 columns Previous **1** 2 3 4 5 6 ... 1000 Next

```
Top_Buyers <- subset(joined_df, percentage < 100)
track_k_buyers <- head(Top_Buyers, 70)
nk <- (unique(track_k_buyers))
Top_Buyers
```

	Buyer	Timestamp	TokenAmount	Open	High	Low	Close	Volume	Market
	<int>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<c
4	4853589	2018-03-12	161067928.53	0.376652	0.377374	0.338550	0.346355	3356110	198140
5	299810	2018-03-12	140000000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
12	4853608	2018-03-12	89980000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
17	4853600	2018-03-12	54001283.71	0.376652	0.377374	0.338550	0.346355	3356110	198140
20	4861571	2018-01-03	49969915.13	0.488586	0.573725	0.450985	0.569930	11634500	163097
21	4853606	2018-03-12	48214412.33	0.376652	0.377374	0.338550	0.346355	3356110	198140
23	2336987	2018-03-12	44733843.75	0.376652	0.377374	0.338550	0.346355	3356110	198140
24	4853598	2018-03-12	42324660.58	0.376652	0.377374	0.338550	0.346355	3356110	198140
28	4864859	2017-09-02	29000000.00	0.075317	0.075590	0.049599	0.060716	2840680	42351
30	4853590	2018-03-12	27876598.50	0.376652	0.377374	0.338550	0.346355	3356110	198140

1-10 of 10,000 rows | 1-10 of 11 columns Previous **1** 2 3 4 5 6 ... 1000 Next

```
message('The value of K is: ',nrow(count(nk)))
```

```
## The value of K is: 35
```

```
cor.test(track_k_buyers$TokenAmount, track_k_buyers$MarketCap, method = "pearson")
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  track_k_buyers$TokenAmount and track_k_buyers$MarketCap
## t = 4.466, df = 68, p-value = 3.085e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2716537 0.6396279
## sample estimates:
##       cor 
## 0.4762292
```

```
cor.test(Top_Buyers$Open, Top_Buyers$High, method = "pearson")
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  Top_Buyers$Open and Top_Buyers$High
## t = 1799.4, df = 183760, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9725313 0.9730224
## sample estimates:
##       cor 
## 0.9727779
```

```
cor.test(Top_Buyers$Close, Top_Buyers$High, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  Top_Buyers$Close and Top_Buyers$High
## t = 3111.5, df = 183760, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9905571 0.9907275
## sample estimates:
##       cor
## 0.9906427
```

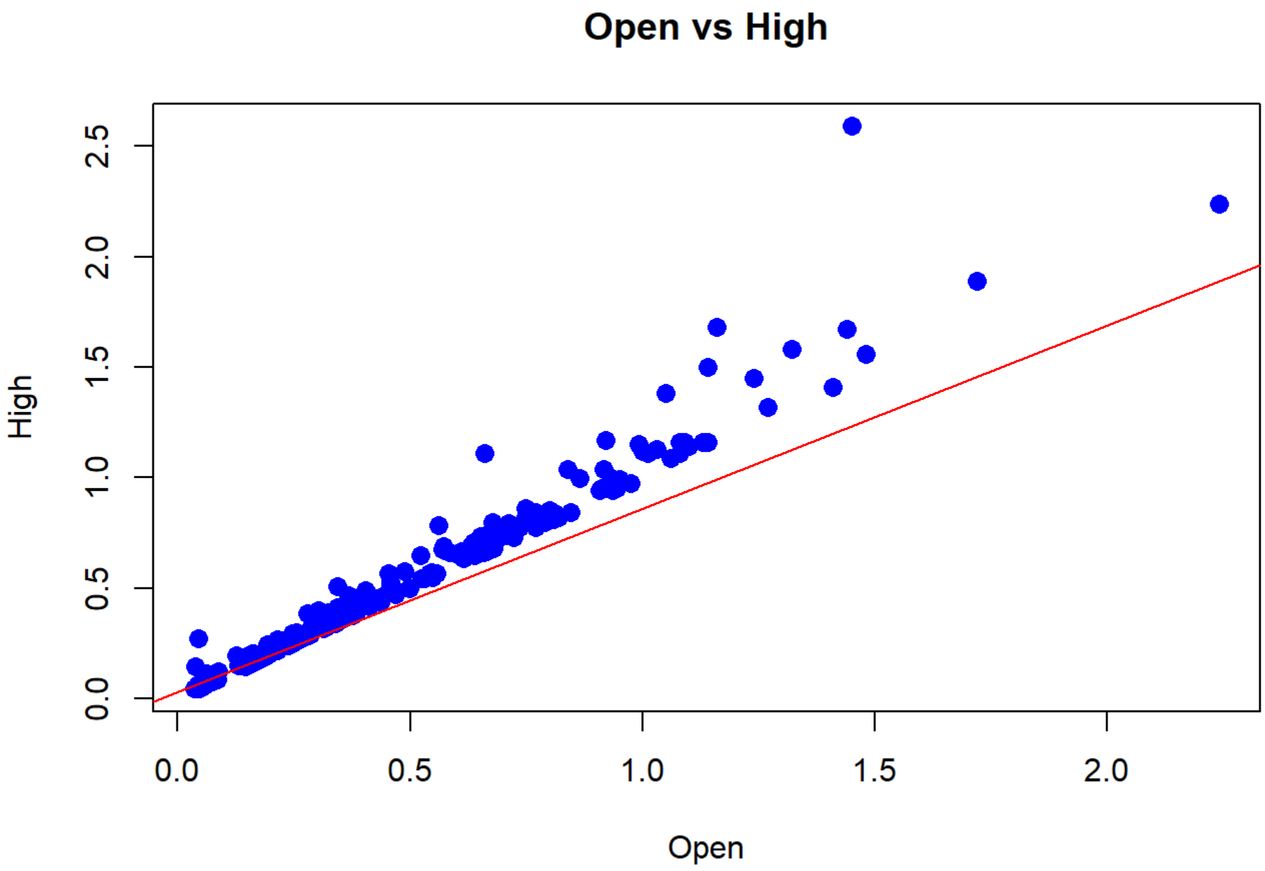
```
linearModOH <- lm(Open ~ High, data=Top_Buyers)  # build linear regression model on full data
linearModCH <- lm(Close ~ High, data=Top_Buyers)
linearModTM <- lm(Close ~ High, data=Top_Buyers)
summary(linearModOH)
```

```
##
## Call:
## lm(formula = Open ~ High, data = Top_Buyers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73137 -0.02545 -0.00422  0.03887  0.34917
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0314175  0.0003413   92.06   <2e-16 ***
## High        0.8300967  0.0004613 1799.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09558 on 183757 degrees of freedom
## Multiple R-squared:  0.9463, Adjusted R-squared:  0.9463
## F-statistic: 3.238e+06 on 1 and 183757 DF,  p-value: < 2.2e-16
```

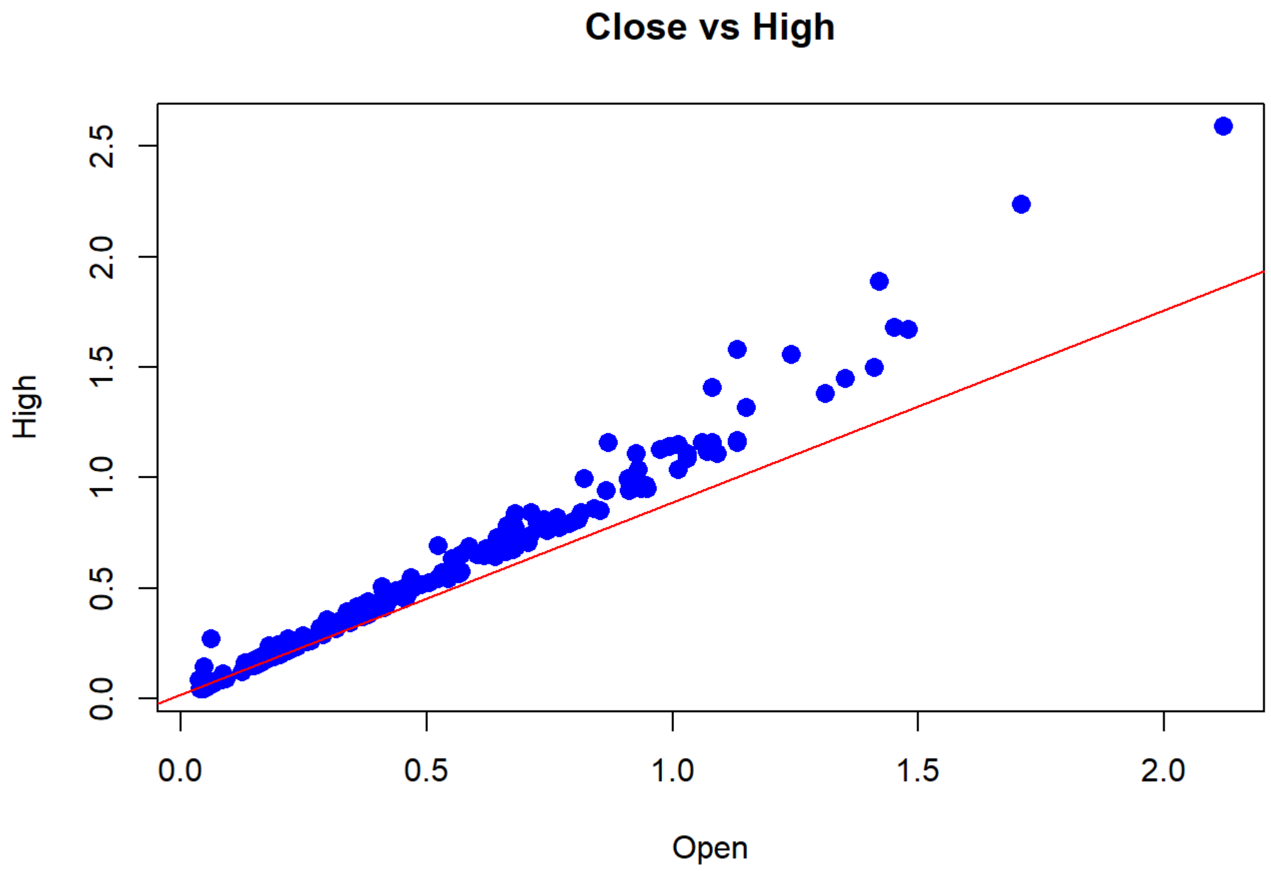
```
modelSummary <- summary(linearModOH)  # capture model summary as an object
modelCoeffs <- modelSummary$coefficients  # model coefficients
modelCoeffs
```

```
##               Estimate   Std. Error    t value Pr(>|t|)
## (Intercept) 0.03141752 0.0003412632   92.06243        0
## High        0.83009667 0.0004613096 1799.43489        0
```

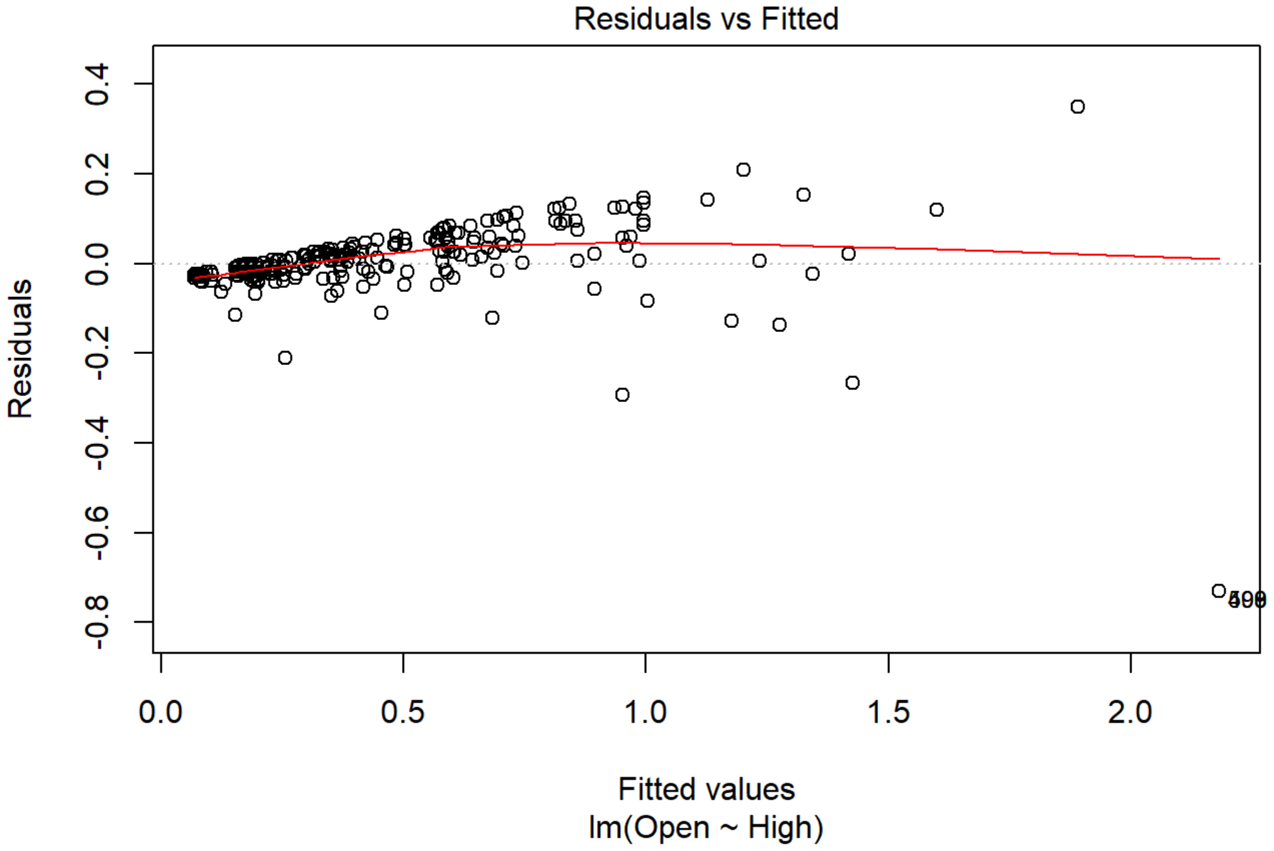
```
plot(Top_Buyers$Open, Top_Buyers$High, pch = 16, cex = 1.3, col = "blue", main = "Open vs High",
xlab = "Open", ylab = "High")
abline(lm(Top_Buyers$Open ~Top_Buyers$High), col = 'red')
```

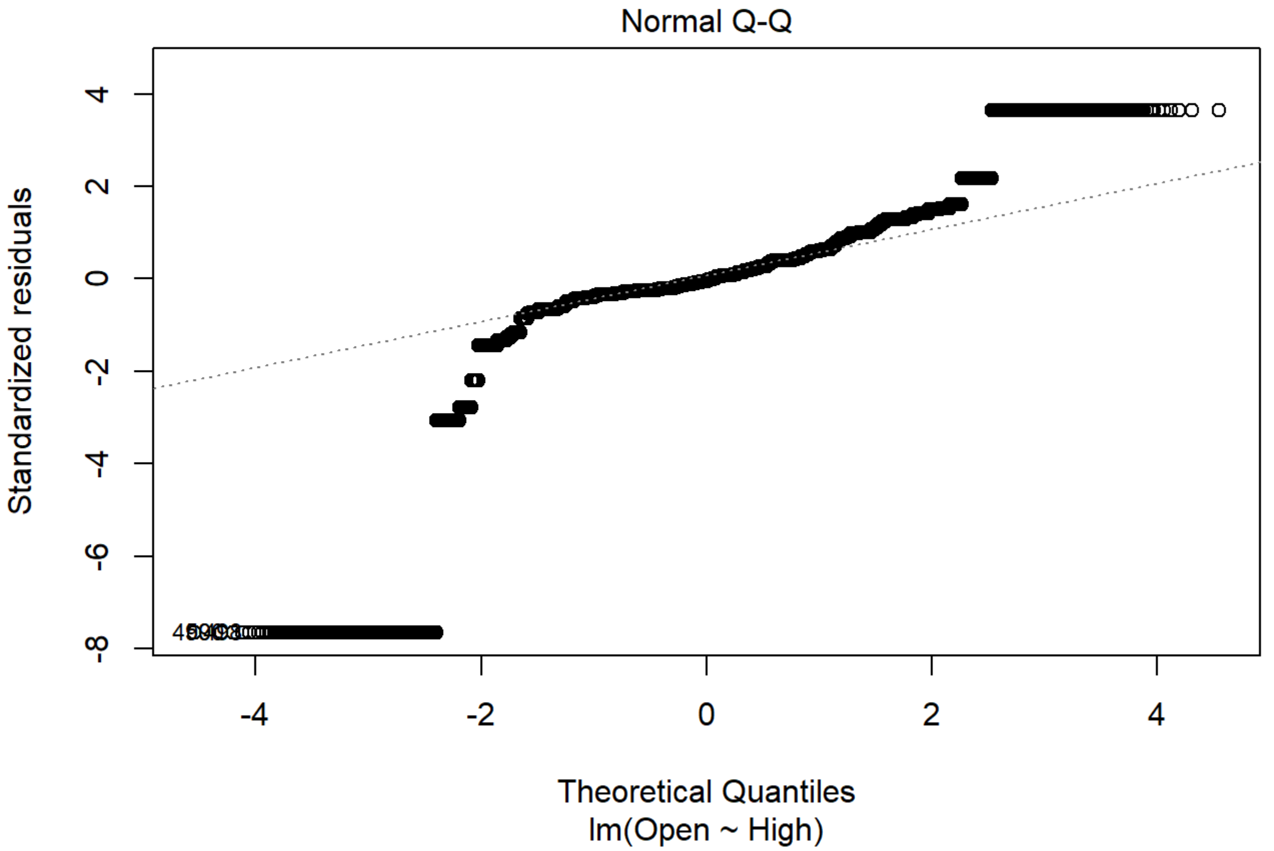


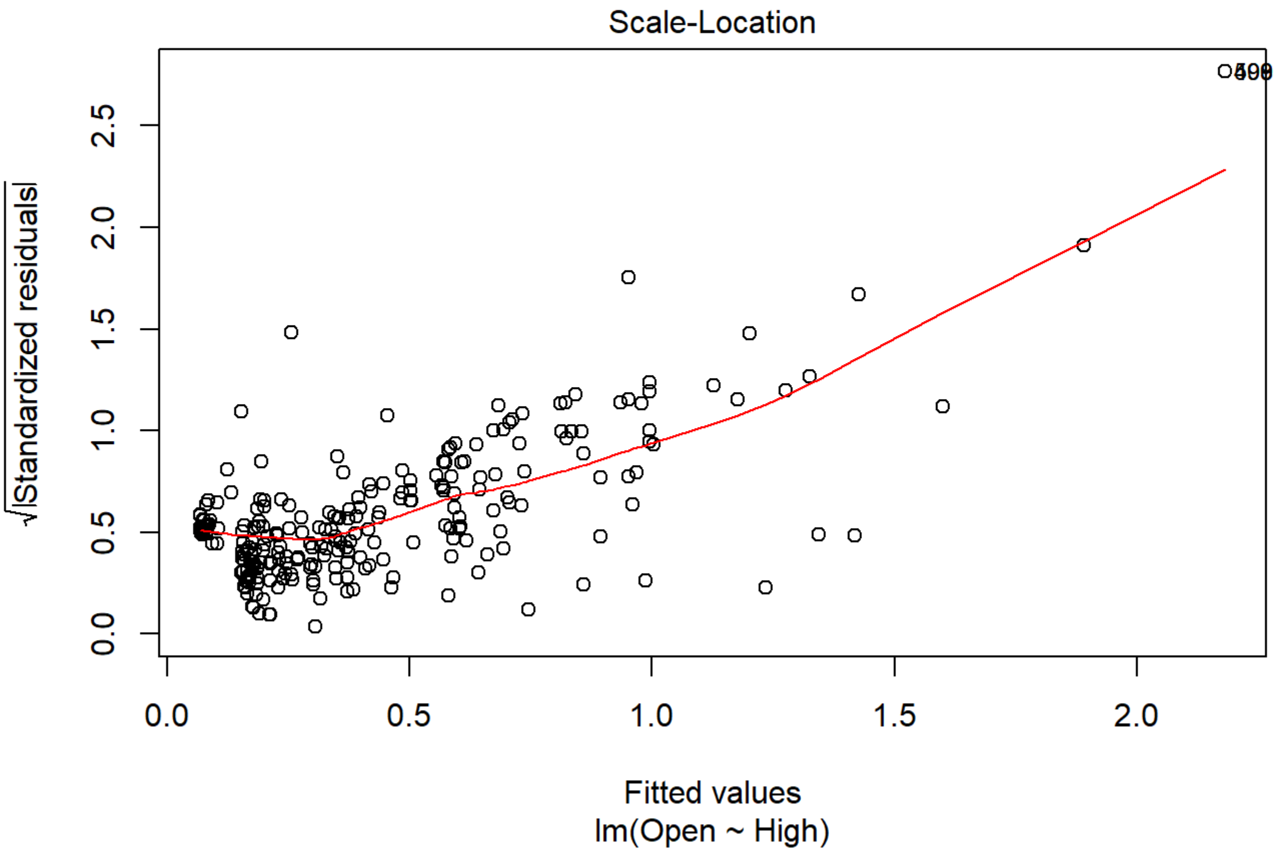
```
plot(Top_Buyers$Close, Top_Buyers$High, pch = 16, cex = 1.3, col = "blue", main = "Close vs Hig
h", xlab = "Open", ylab = "High")
abline(lm(Top_Buyers$Close ~Top_Buyers$High), col = 'red')
```

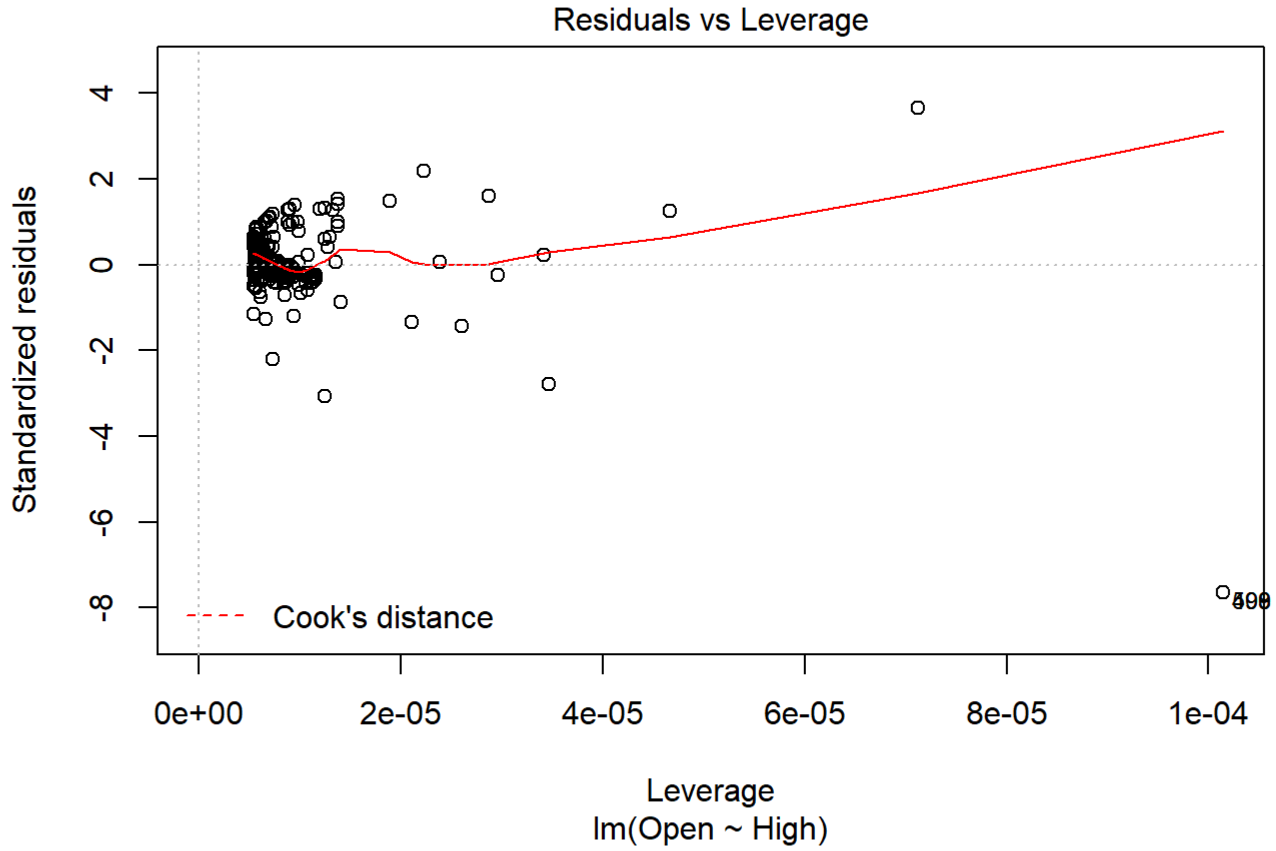


```
plot(linearModOH)
```

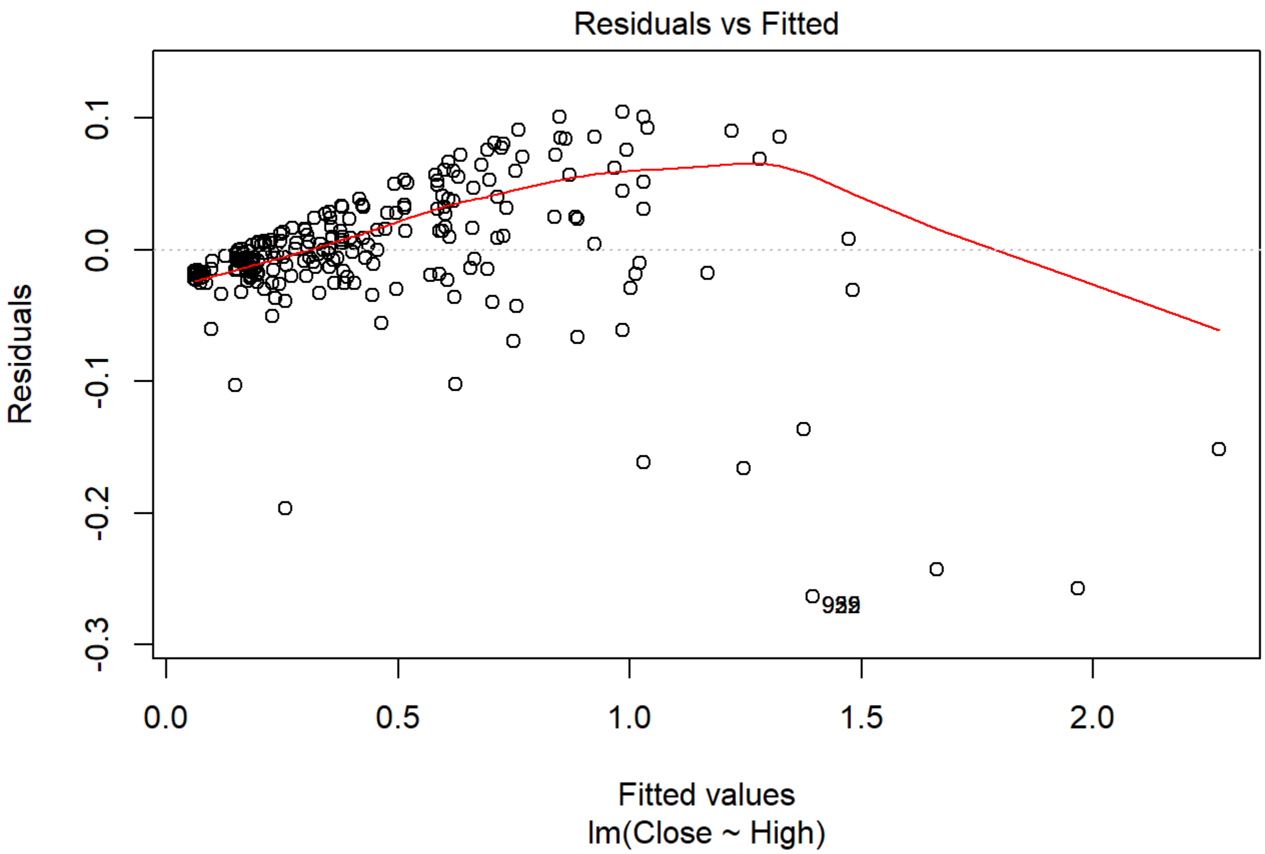

Residuals vs Fitted
Residuals
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
0.0
0.5
1.0
1.5
2.0
500
498
Fitted values
lm(Open ~ High)

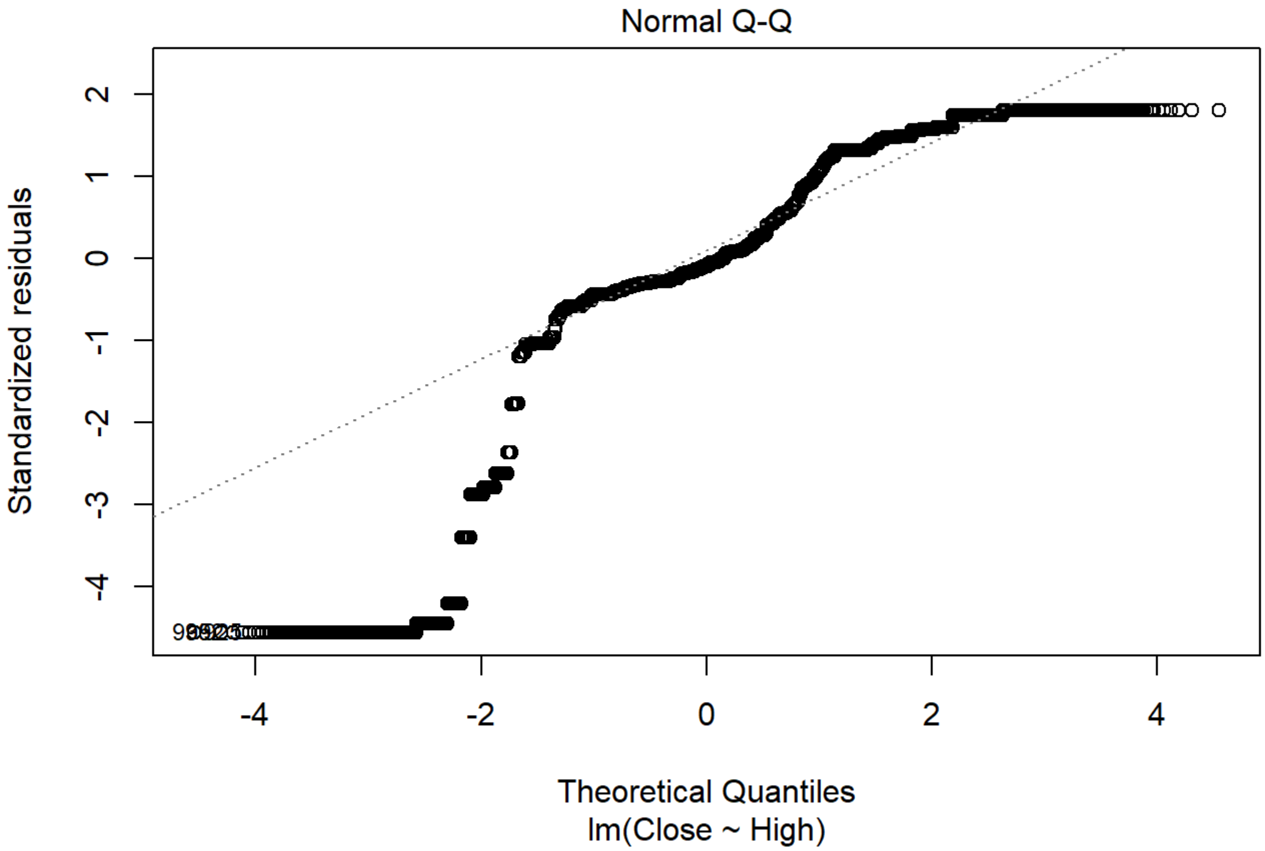

Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-6
-8
-4
-2
0
2
4
499
500
498
Theoretical Quantiles
lm(Open ~ High)

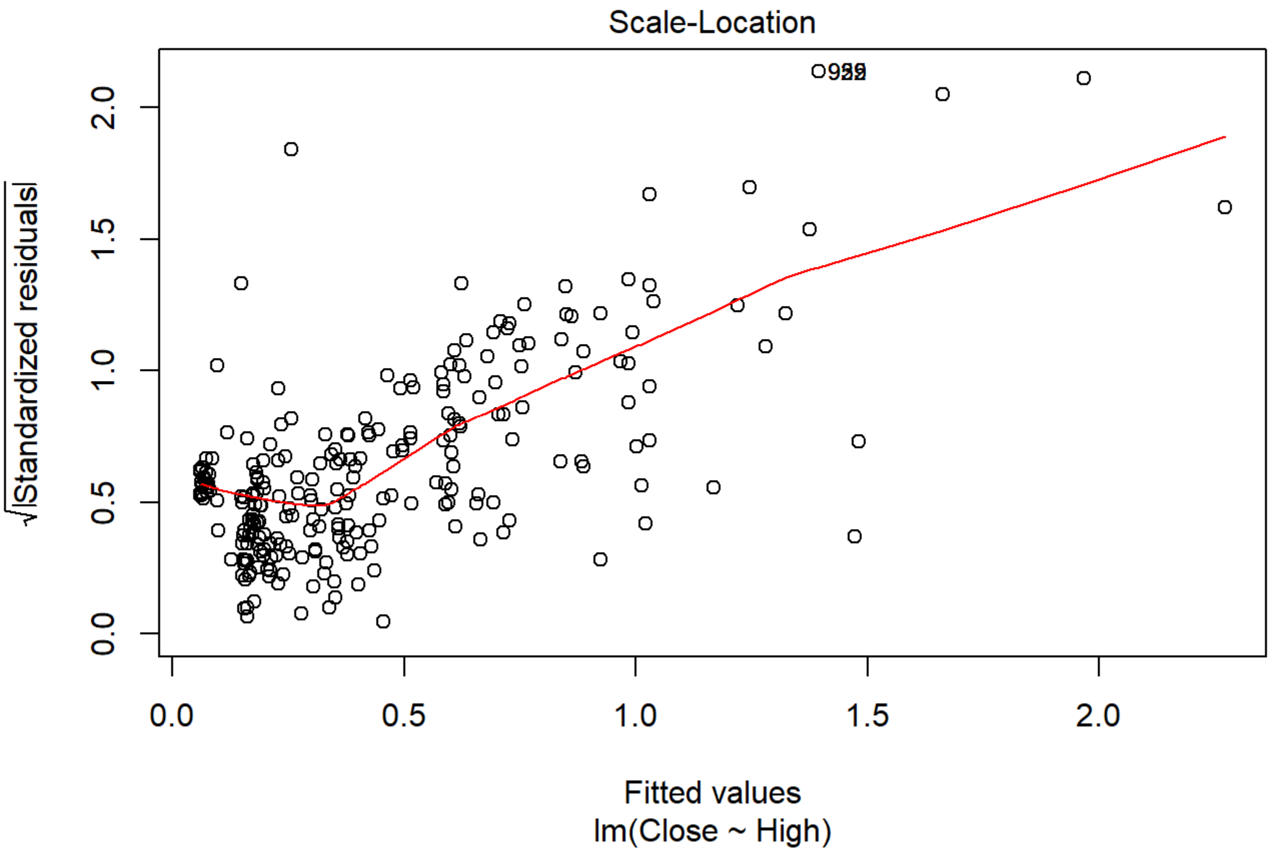

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
1.5
2.0
Fitted values
lm(Open ~ High)
500

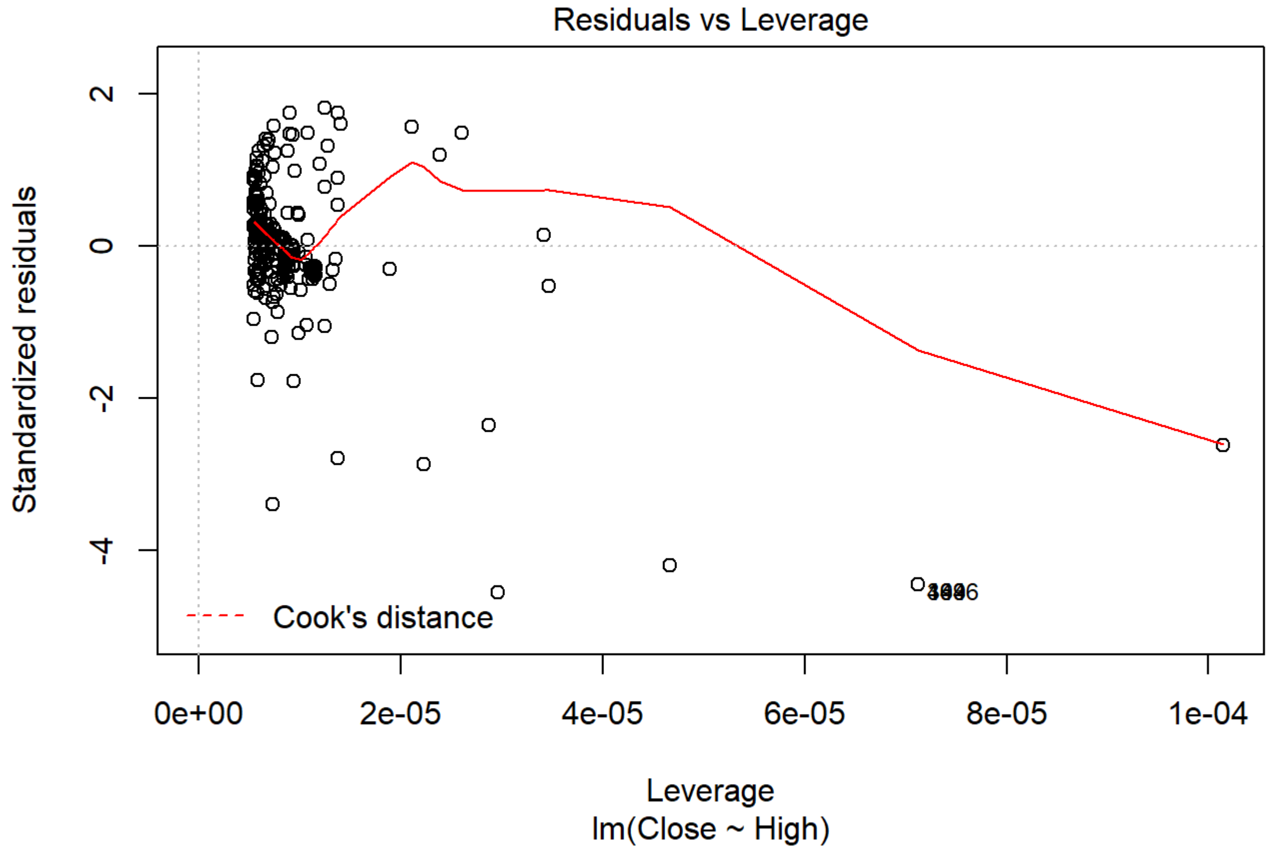

Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
-6
-8
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
Leverage
lm(Open ~ High)
Cook's distance
500

```
plot(linearModCH)
```

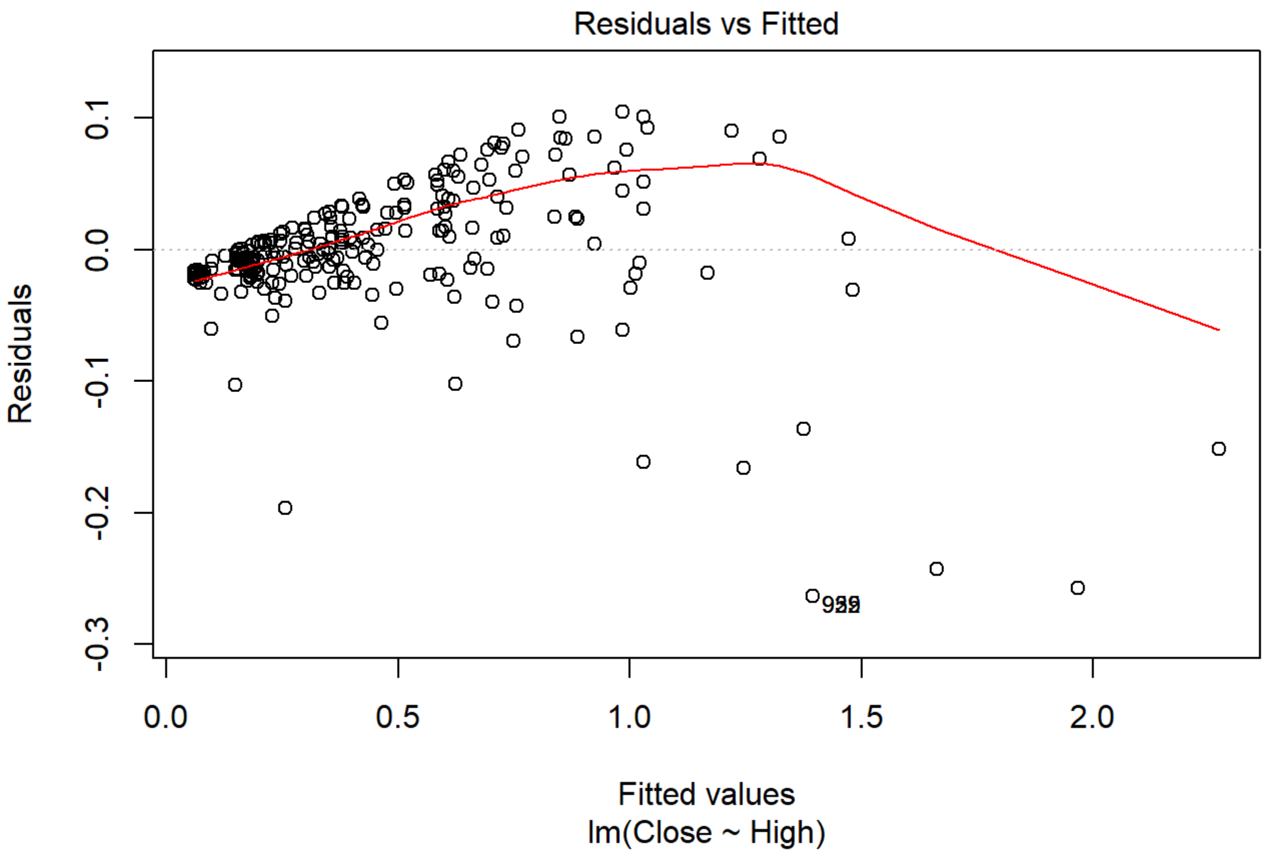

Residuals vs Fitted
Residuals
0.1
0.0
-0.1
-0.2
-0.3
952
0.0
0.5
1.0
1.5
2.0
Fitted values
lm(Close ~ High)

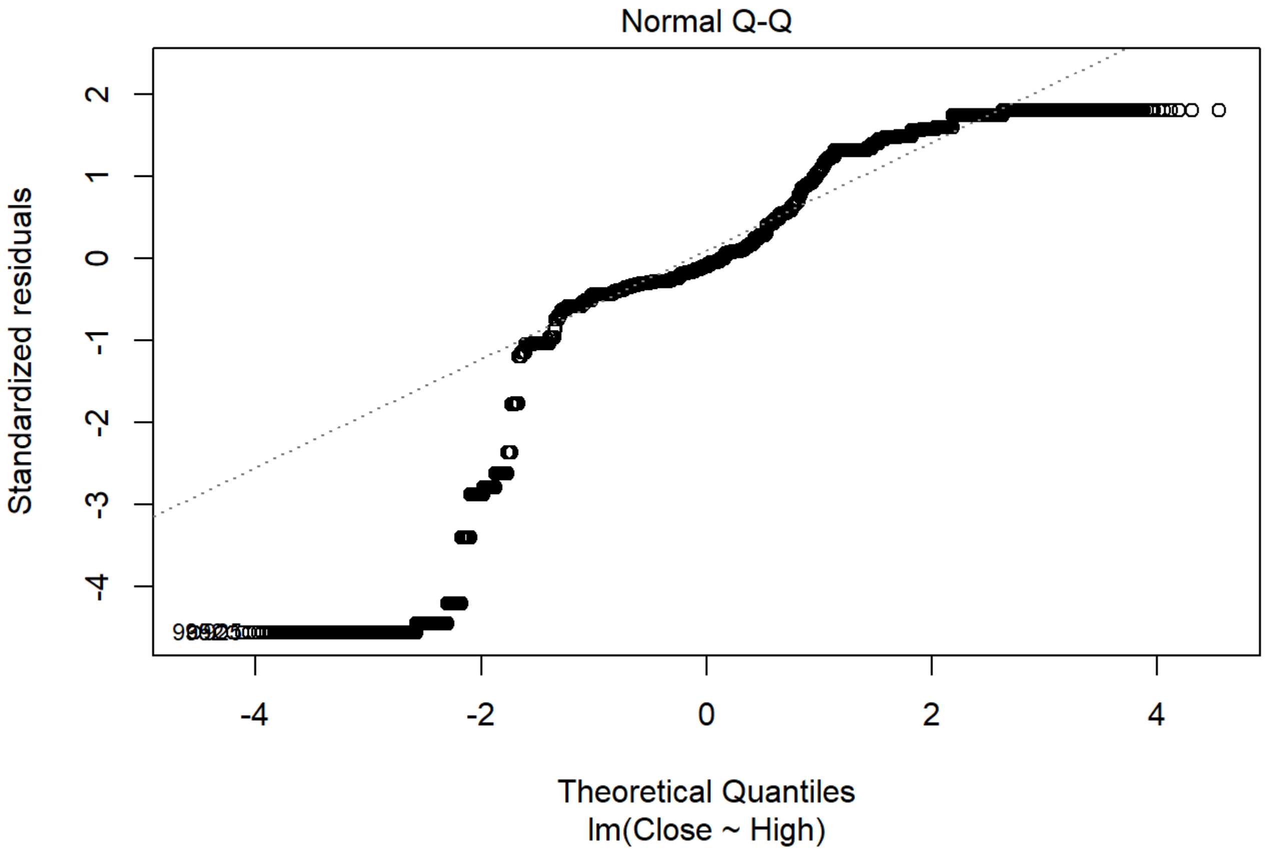

Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
Theoretical Quantiles
lm(Close ~ High)

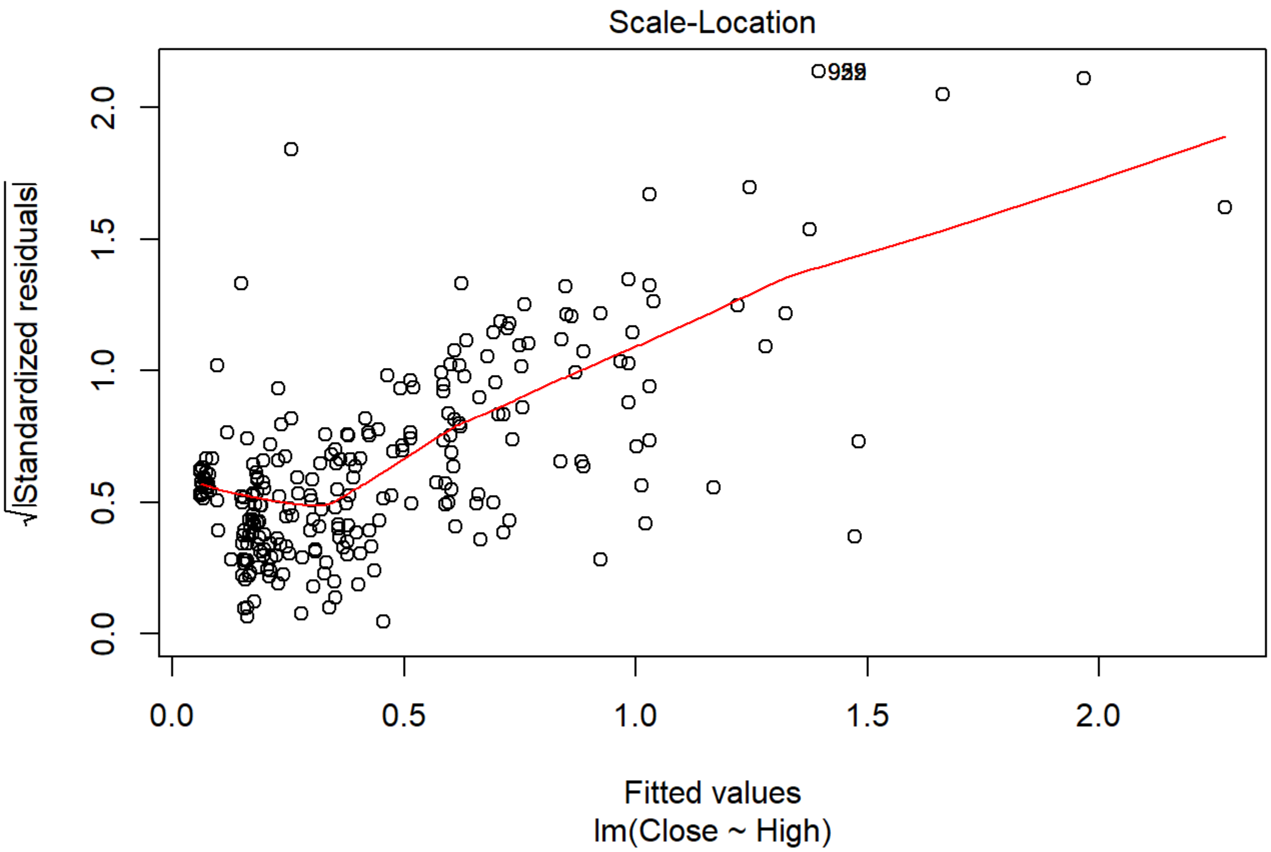

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
952
Fitted values
lm(Close ~ High)

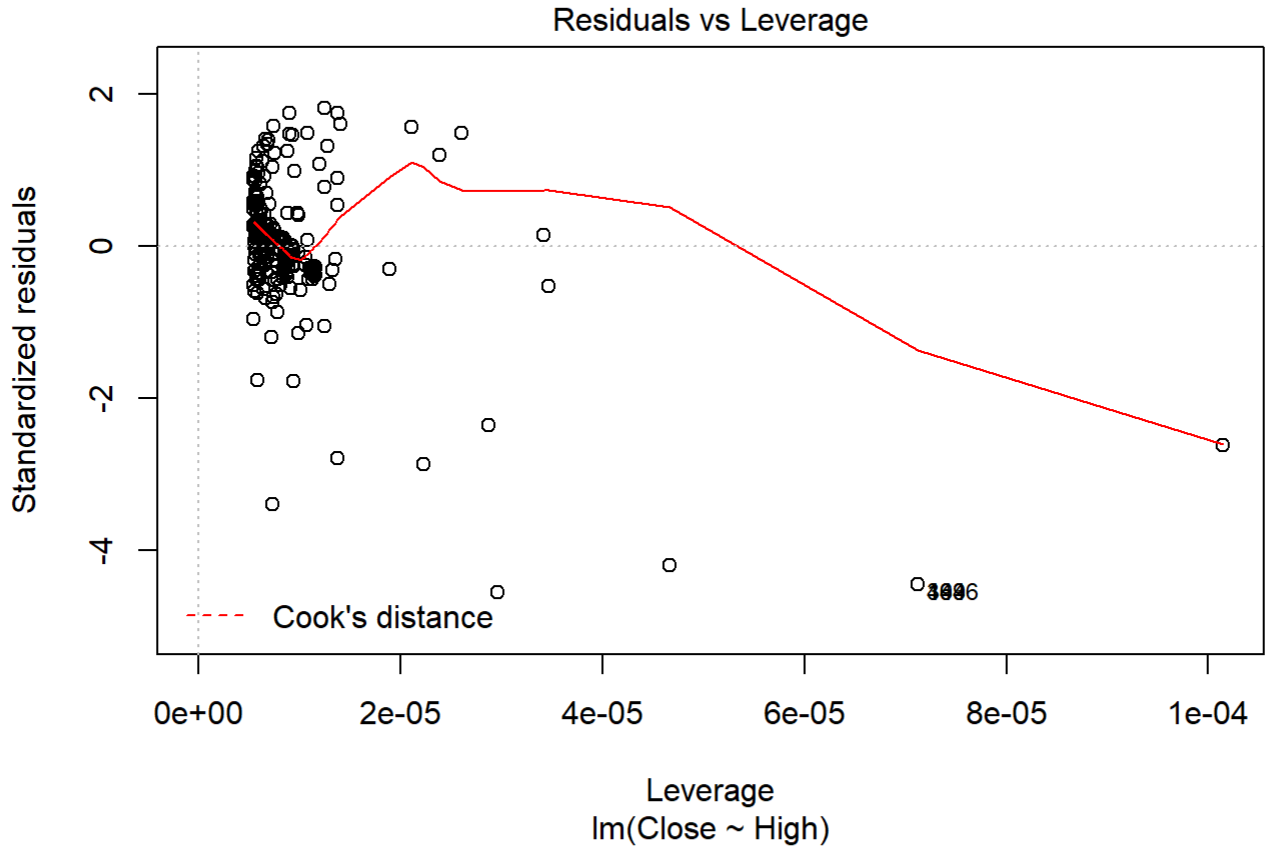

Residuals vs Leverage
Standardized residuals
2
0
-2
-4
Cook's distance
3006
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
Leverage
lm(Close ~ High)

```
plot(linearModTM)
```


Residuals vs Fitted
Residuals
0.1
0.0
-0.1
-0.2
-0.3
959
0.0
0.5
1.0
1.5
2.0
Fitted values
lm(Close ~ High)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-3
-4
959
-4
-2
0
2
4
Theoretical Quantiles
lm(Close ~ High)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
Fitted values
lm(Close ~ High)


Residuals vs Leverage
Standardized residuals
2
0
-2
-4
Cook's distance
0e+00
2e-05
4e-05
6e-05
8e-05
1e-04
Leverage
lm(Close ~ High)