Funfair

PART 1

```
raw_data <- read.csv(file = 'networkfunfairTX.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: 243617
```

```
summary(raw_data)
```

```
##      Buyer             Seller          Timestamp
##  Min.   :      5   Min.   :      2   Min.   :1.500e+09
##  1st Qu.:     17   1st Qu.: 350630   1st Qu.:1.510e+09
##  Median :1796774   Median :1796774   Median :1.515e+09
##  Mean   :1619556   Mean   :1961871   Mean   :1.513e+09
##  3rd Qu.:3853714   3rd Qu.:3851811   3rd Qu.:1.517e+09
##  Max.   :3897986   Max.   :3897986   Max.   :1.526e+09
##   TokenAmount
##  Min.   :1.000e+00
##  1st Qu.:1.228e+11
##  Median :5.236e+11
##  Mean   :1.676e+73
##  3rd Qu.:2.597e+12
##  Max.   :5.790e+76
```

```
Total_circulation_amount = 6548879189 * 10^8
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: 91
```

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

```
##      Buyer             Seller          Timestamp
##  Min.   :      5   Min.   :      2   Min.   :1.500e+09
##  1st Qu.:     17   1st Qu.: 351355   1st Qu.:1.510e+09
##  Median :1796774   Median :1796774   Median :1.515e+09
##  Mean   :1618814   Mean   :1962402   Mean   :1.513e+09
##  3rd Qu.:3853814   3rd Qu.:3851824   3rd Qu.:1.517e+09
##  Max.   :3897986   Max.   :3897986   Max.   :1.526e+09
##   TokenAmount
##  Min.   :1.000e+00
##  1st Qu.:1.228e+11
##  Median :5.227e+11
##  Mean   :1.294e+13
##  3rd Qu.:2.597e+12
##  Max.   :3.174e+17
```

```
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>
17	3844401	1
17	3245908	1
560	3844402	1
3844402	5	1
17	3844403	1
323412	323596	2
17	3844404	1
323596	5	2
77052	320213	1
3844405	3844406	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.   :      5   Min.   :      2   Min.   :   1.000
##  1st Qu.:     17   1st Qu.: 398570   1st Qu.:   1.000
##  Median : 406916   Median :2404557   Median :   1.000
##  Mean   :1389604   Mean   :2313709   Mean   :   1.805
##  3rd Qu.:2186266   3rd Qu.:3864878   3rd Qu.:   1.000
##  Max.   :3897986   Max.   :3897986   Max.   :1732.000
```

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

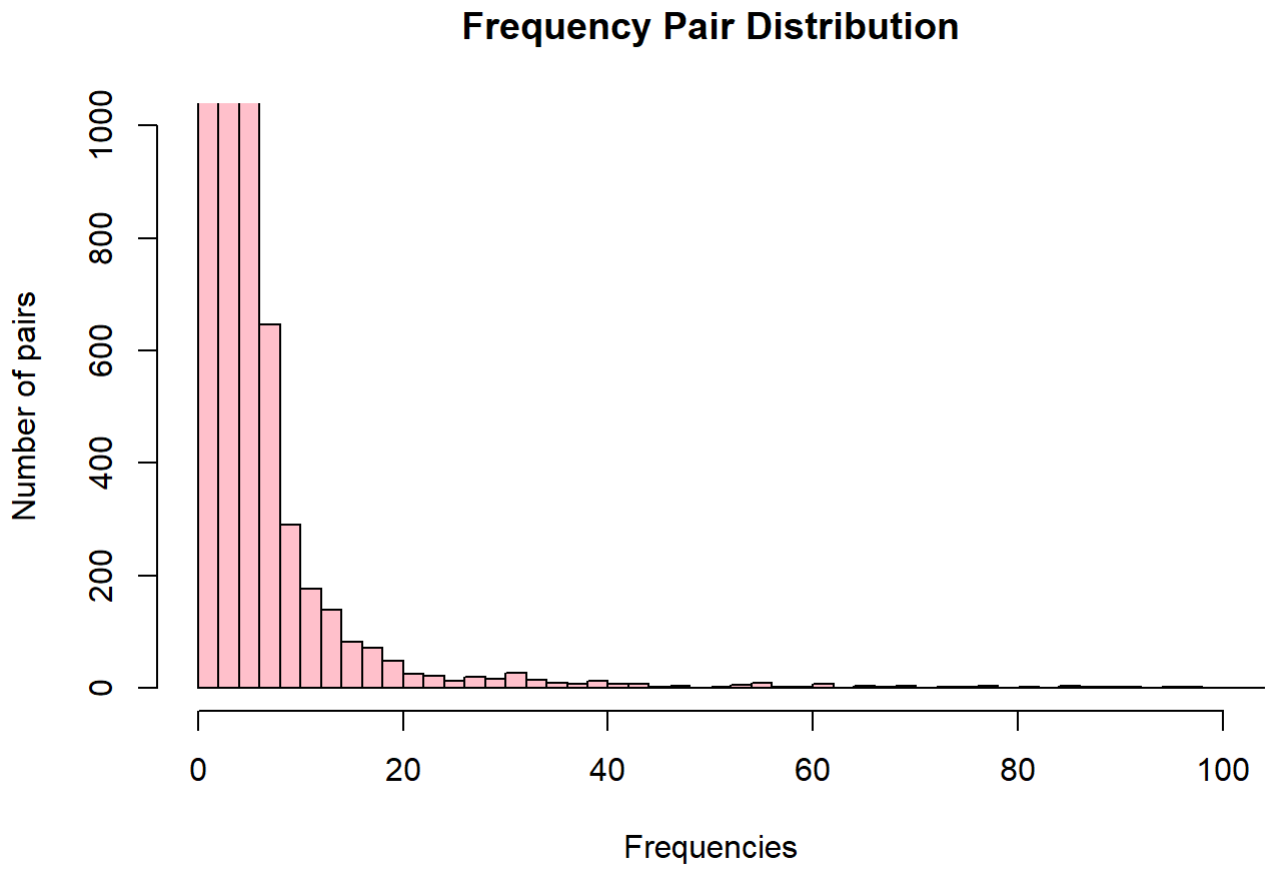
```
## Variance: 203.742497637121
```

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

```
## Standard Deviation: 14.2738396248914
```

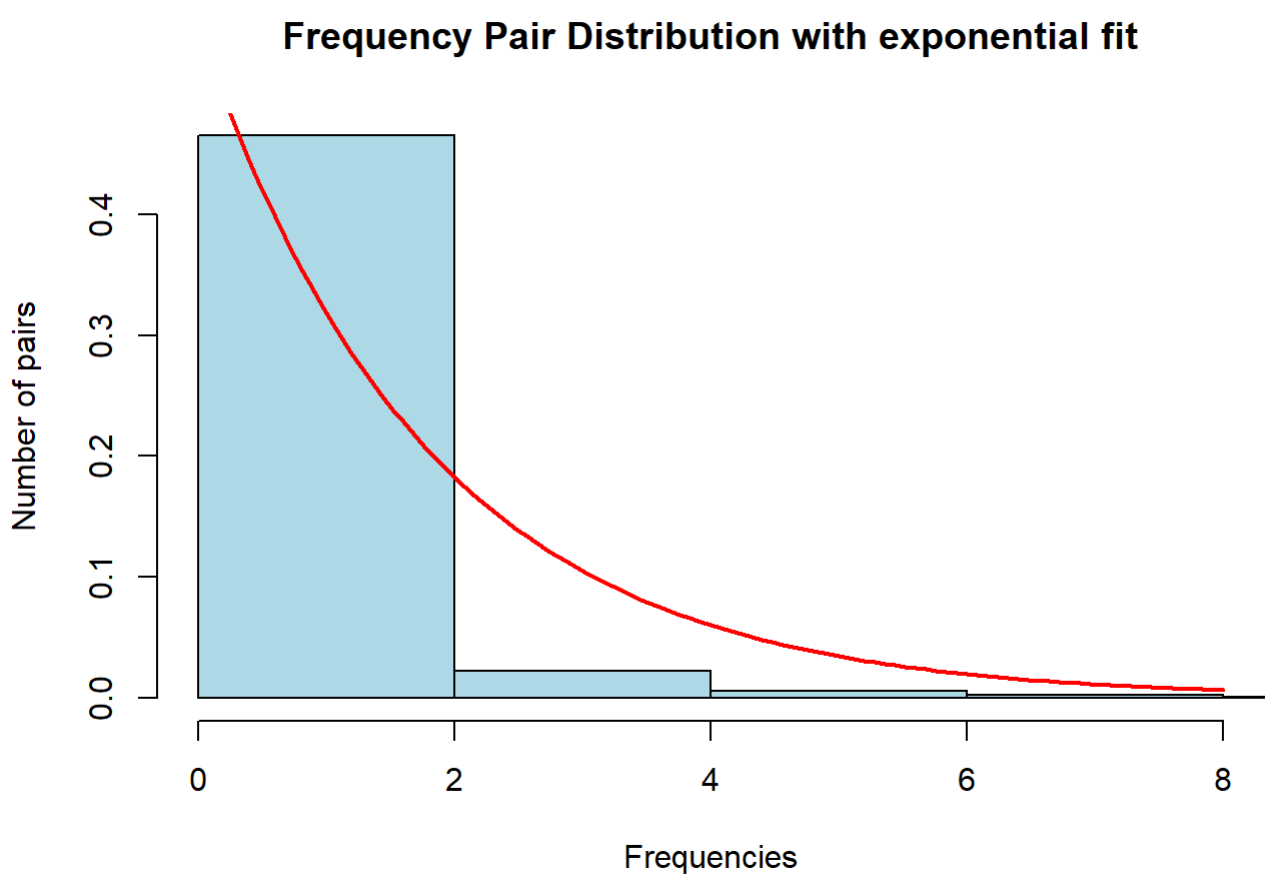
```
freq <- Buyer_Seller_Pair_Frequencies$Frequency
library(MASS)

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)
```

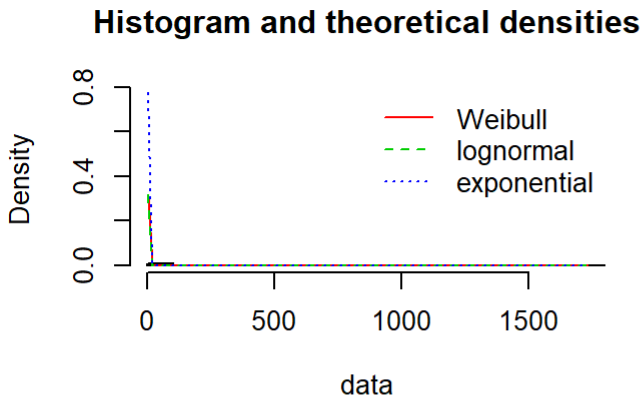


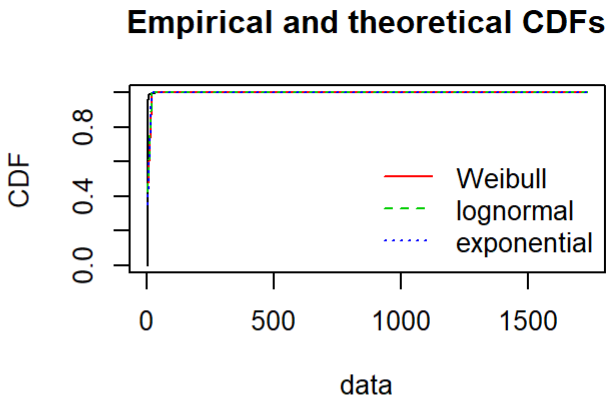

```
library(fitdistrplus)
```

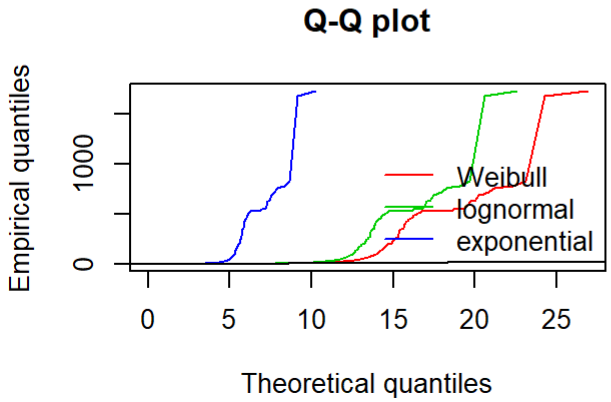
```
## Loading required package: survival
```

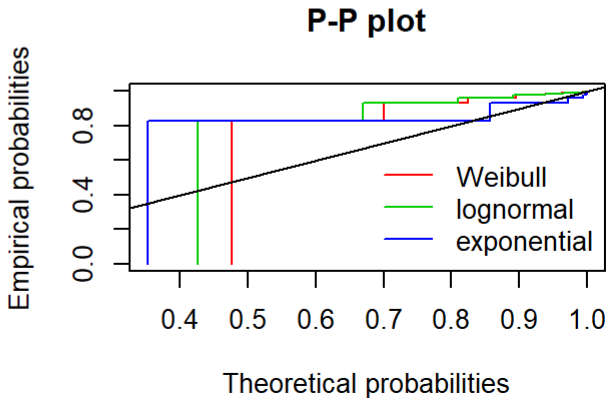
```
fit_w  <- fitdist(freq, "weibull")
fit_ln <- fitdist(freq, "lnorm")
fit_ex <- fitdist(freq, "exp")

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/10^8
Time <- as.Date(as.POSIXct(preprocessed_data$Timestamp, origin = '1970-01-01'))
preprocessed_data$Timestamp <- Time
preprocessed_data
```

	Buyer <int>	Seller <int>	Timestamp <date>	TokenAmount <dbl>
1	17	3844401	2018-04-24	8.492000e+03
2	17	3245908	2018-04-24	1.951000e+03
3	560	3844402	2018-04-24	1.204900e+04
4	3844402	5	2018-04-24	1.204900e+04
5	17	3844403	2018-04-24	1.499951e+06
6	323412	323596	2018-04-24	1.540749e+05
7	17	3844404	2018-04-24	7.489721e+03
8	323596	5	2018-04-24	1.540749e+05
9	77052	320213	2018-04-24	3.816000e+03

	Buyer <int>	Seller <int>	Timestamp <date>	TokenAmount <dbl>
10	3844405	3844406	2018-04-24	1.321279e+03

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

```
library("readxl")
my_data <- read_excel("FunFair_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)
my_data
```

Timestamp <date>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <dbl>	MarketCap <dbl>
2019-05-03	0.005504	0.005736	0.005414	0.005524	1353376	36178685
2019-05-02	0.005442	0.005538	0.005381	0.005503	860619	36035429
2019-05-01	0.005493	0.005683	0.005395	0.005446	967859	35662492
2019-04-30	0.005214	0.005549	0.005147	0.005508	738306	36068648
2019-04-29	0.005136	0.005298	0.004987	0.005214	742474	34145957
2019-04-28	0.005372	0.005454	0.005078	0.005134	1235780	33621167
2019-04-27	0.005182	0.005401	0.005147	0.005371	1466794	35176582
2019-04-26	0.005180	0.005327	0.005057	0.005182	1240067	33936193
2019-04-25	0.005680	0.005680	0.005166	0.005172	1543120	33873426
2019-04-24	0.006255	0.006255	0.005548	0.005681	1787524	37204119

1-10 of 671 rows Previous **1** 2 3 4 5 6 … 68 Next

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

	Buyer <int>	Timestamp <date>	TokenAmount <dbl>
190354	2335751	2017-09-26	3173696075.7
190355	2335751	2017-09-26	3000000000.0
190356	2335751	2017-09-26	1000000000.0

	Buyer <int>	Timestamp <date>	TokenAmount <dbl>
190357	2335751	2017-09-26	1000000000.0
190358	2335751	2017-09-26	1000000000.0
190361	2335751	2017-09-26	1000000000.0
110998	2335752	2018-01-05	270362400.0
111017	2335752	2018-01-05	253579700.0
182339	3893030	2017-11-20	249999000.0
110997	2335752	2018-01-05	236567100.0

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>	Marke
1	2335751	2017-09-26	3173696075.7	0.020853	0.025125	0.020444	0.022800	861307	8754
2	2335751	2017-09-26	3000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
3	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
4	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
5	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
6	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
7	2335752	2018-01-05	270362400.0	0.167685	0.189302	0.143586	0.160535	64546500	68225
8	2335752	2018-01-05	253579700.0	0.167685	0.189302	0.143586	0.160535	64546500	68225
9	3893030	2017-11-20	249999000.0	0.016968	0.018724	0.016759	0.017957	1412660	7182
10	2335752	2018-01-05	236567100.0	0.167685	0.189302	0.143586	0.160535	64546500	68225

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 <int>	Timestamp <date>	TokenAmount <dbl>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <dbl>	Marke
1	2335751	2017-09-26	3173696075.7	0.020853	0.025125	0.020444	0.022800	861307	8754
2	2335751	2017-09-26	3000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
3	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
4	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
5	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
6	2335751	2017-09-26	1000000000.0	0.020853	0.025125	0.020444	0.022800	861307	8754
7	2335752	2018-01-05	270362400.0	0.167685	0.189302	0.143586	0.160535	64546500	68225
8	2335752	2018-01-05	253579700.0	0.167685	0.189302	0.143586	0.160535	64546500	68225
9	3893030	2017-11-20	249999000.0	0.016968	0.018724	0.016759	0.017957	1412660	7182
10	2335752	2018-01-05	236567100.0	0.167685	0.189302	0.143586	0.160535	64546500	68225

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 <int>	Timestamp <date>	TokenAmo… <dbl>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <dbl>	Market <(
7	2335752	2018-01-05	270362400.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
8	2335752	2018-01-05	253579700.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
10	2335752	2018-01-05	236567100.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
11	2335752	2018-01-05	219784400.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
12	2335752	2018-01-05	202771800.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
14	2335752	2018-01-05	185989100.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
15	2335479	2018-02-06	181183093.7	0.039127	0.054886	0.030425	0.053783	7741380	242221
17	2335752	2018-01-05	168976500.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
19	2335752	2018-01-05	152193800.0	0.167685	0.189302	0.143586	0.160535	64546500	682253
20	2335752	2018-01-05	135181200.0	0.167685	0.189302	0.143586	0.160535	64546500	682253

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: 58
```

```
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 = 4.8642, df = 68, p-value = 7.128e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3101132 0.6637941
## sample estimates:
##       cor
## 0.5080653
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  Top_Buyers$Open and Top_Buyers$High
## t = 2139.3, df = 243510, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9742111 0.9746124
## sample estimates:
##       cor
## 0.9744125
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  Top_Buyers$Close and Top_Buyers$High
## t = 3180.6, df = 243510, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9880833 0.9882701
## sample estimates:
##       cor
## 0.9881771
```

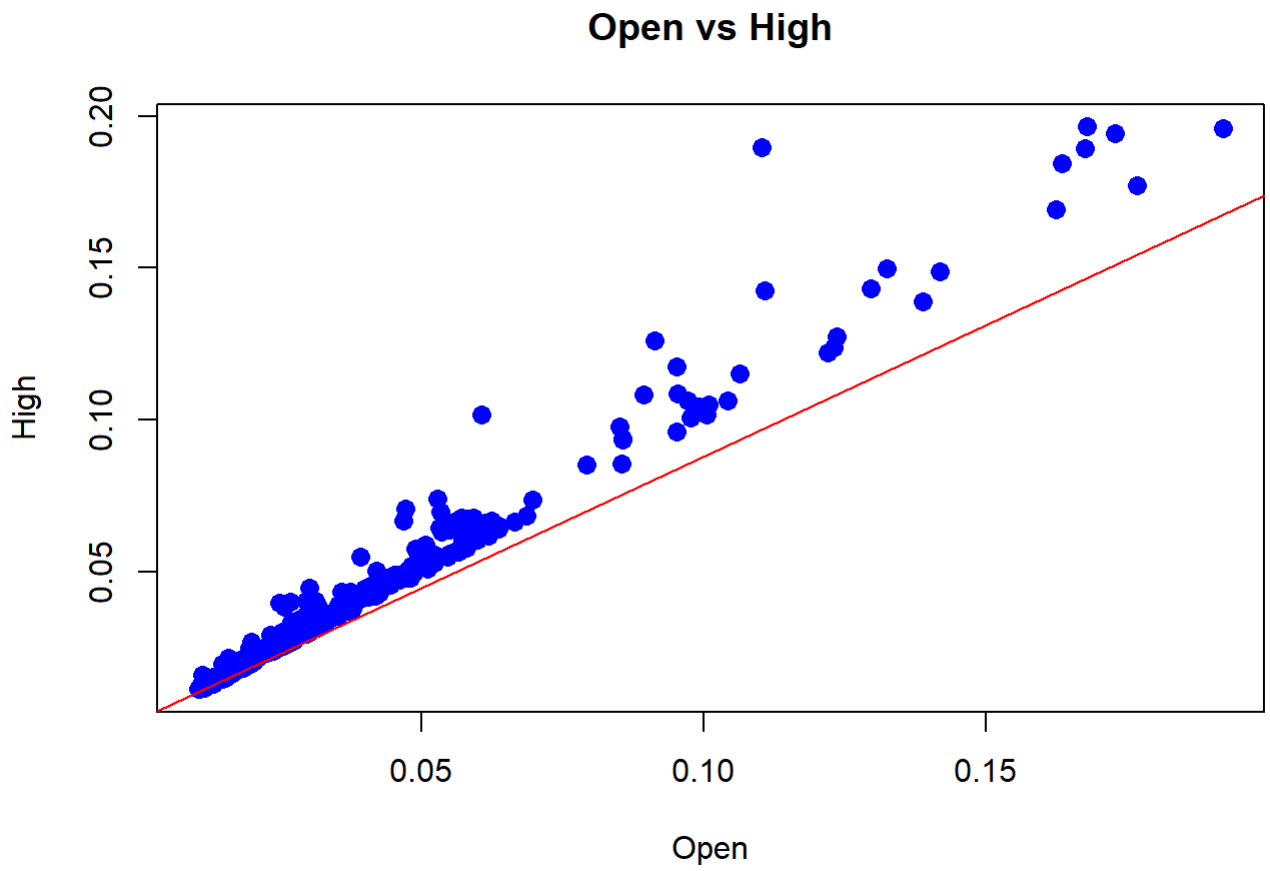
```
linearModOH <- lm(Open ~ High, data=Top_Buyers)  # build linear regression model on full data
linearModCH <- lm(Close ~ High, data=Top_Buyers)
linearModTM <- lm(Close ~ High, data=Top_Buyers)
summary(linearModOH)
```

```
##
## Call:
## lm(formula = Open ~ High, data = Top_Buyers)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.055036 -0.001672  0.000777  0.003362  0.022294
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.431e-03  3.436e-05   41.64   <2e-16 ***
## High        8.648e-01  4.042e-04 2139.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009899 on 243511 degrees of freedom
## Multiple R-squared:  0.9495, Adjusted R-squared:  0.9495
## F-statistic: 4.577e+06 on 1 and 243511 DF,  p-value: < 2.2e-16
```

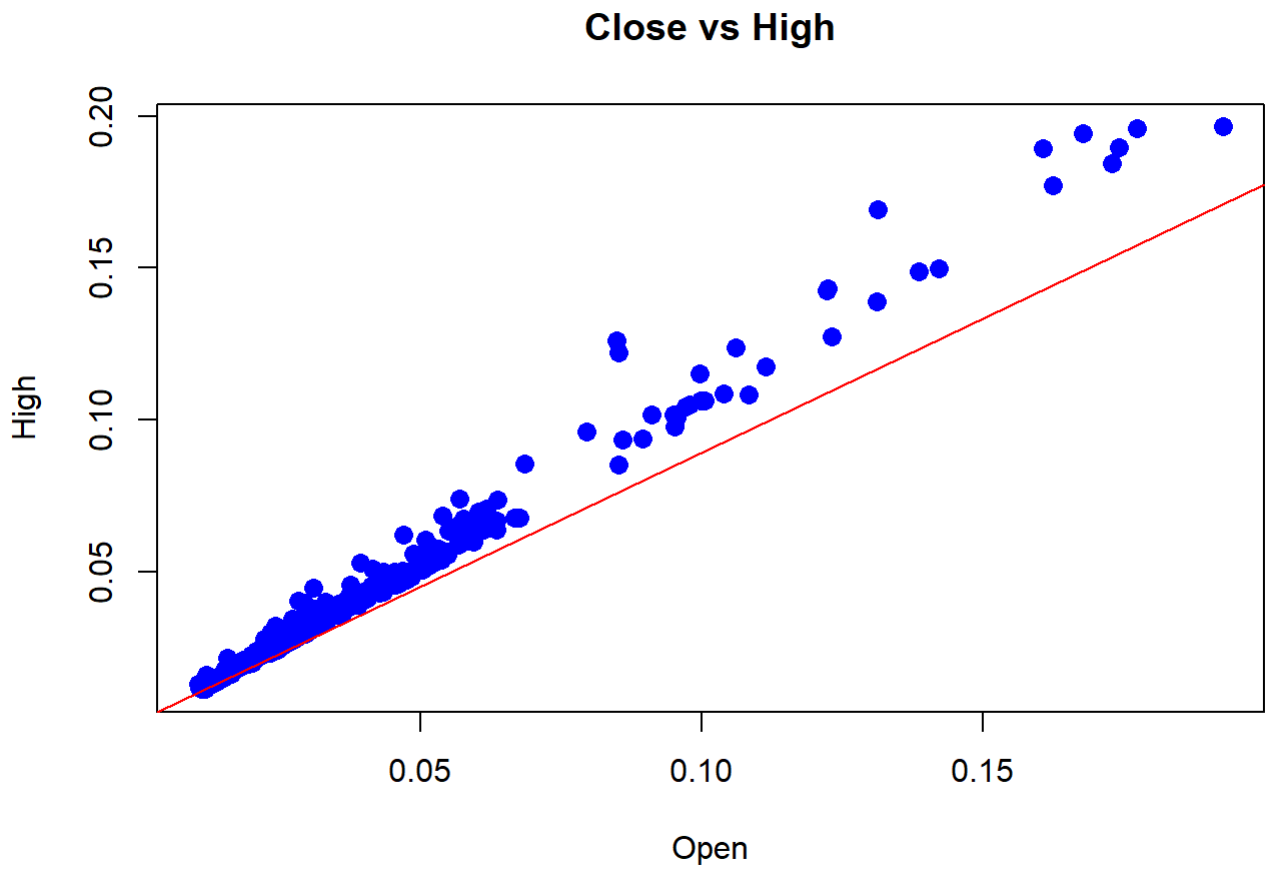
```
modelSummary <- summary(linearModOH)  # capture model summary as an object
modelCoeffs <- modelSummary$coefficients  # model coefficients
modelCoeffs
```

```
##                Estimate   Std. Error    t value Pr(>|t|)
## (Intercept) 0.001430989 3.436372e-05   41.64243        0
## High        0.864763051 4.042291e-04 2139.28921        0
```

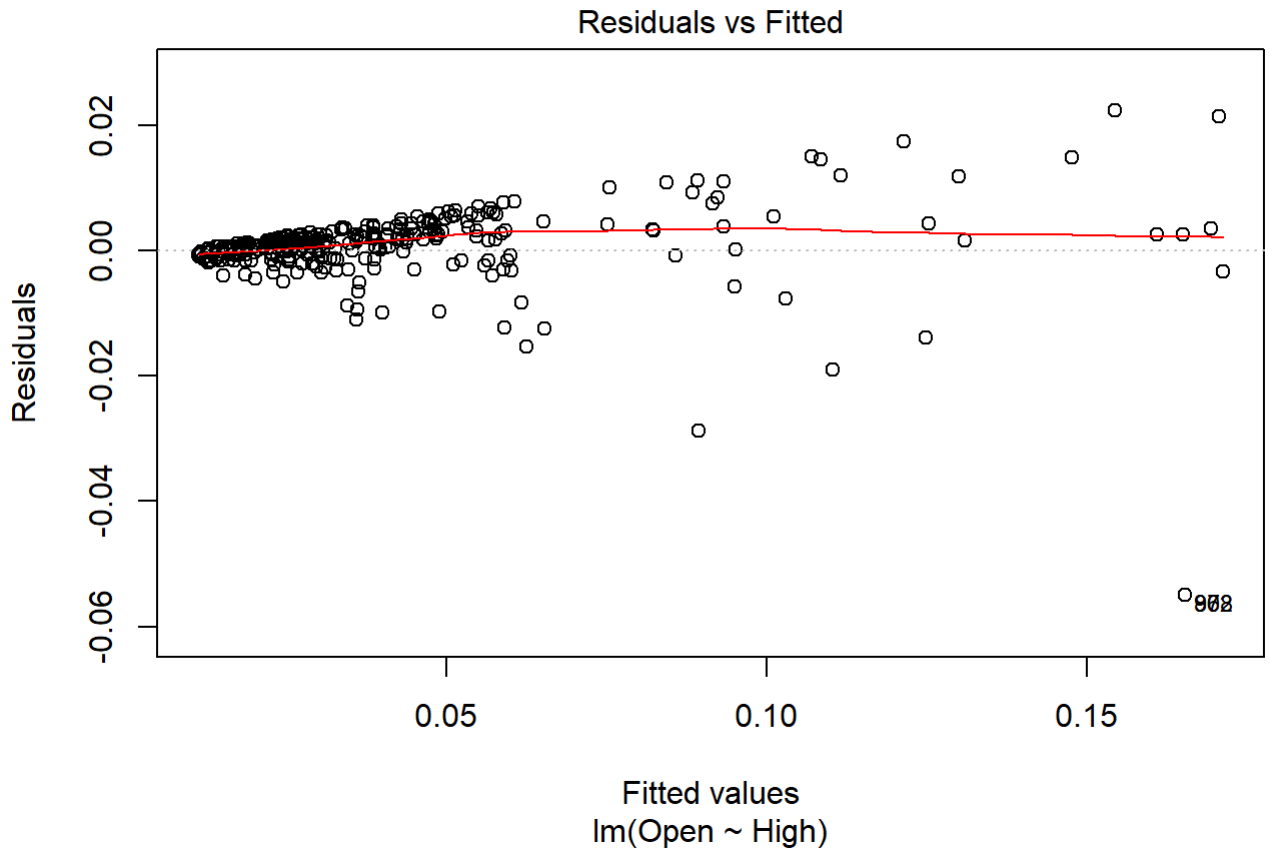
```
plot(Top_Buyers$Open, Top_Buyers$High, pch = 16, cex = 1.3, col = "blue", main = "Open vs High",
xlab = "Open", ylab = "High")
abline(lm(Top_Buyers$Open ~Top_Buyers$High), col = 'red')
```

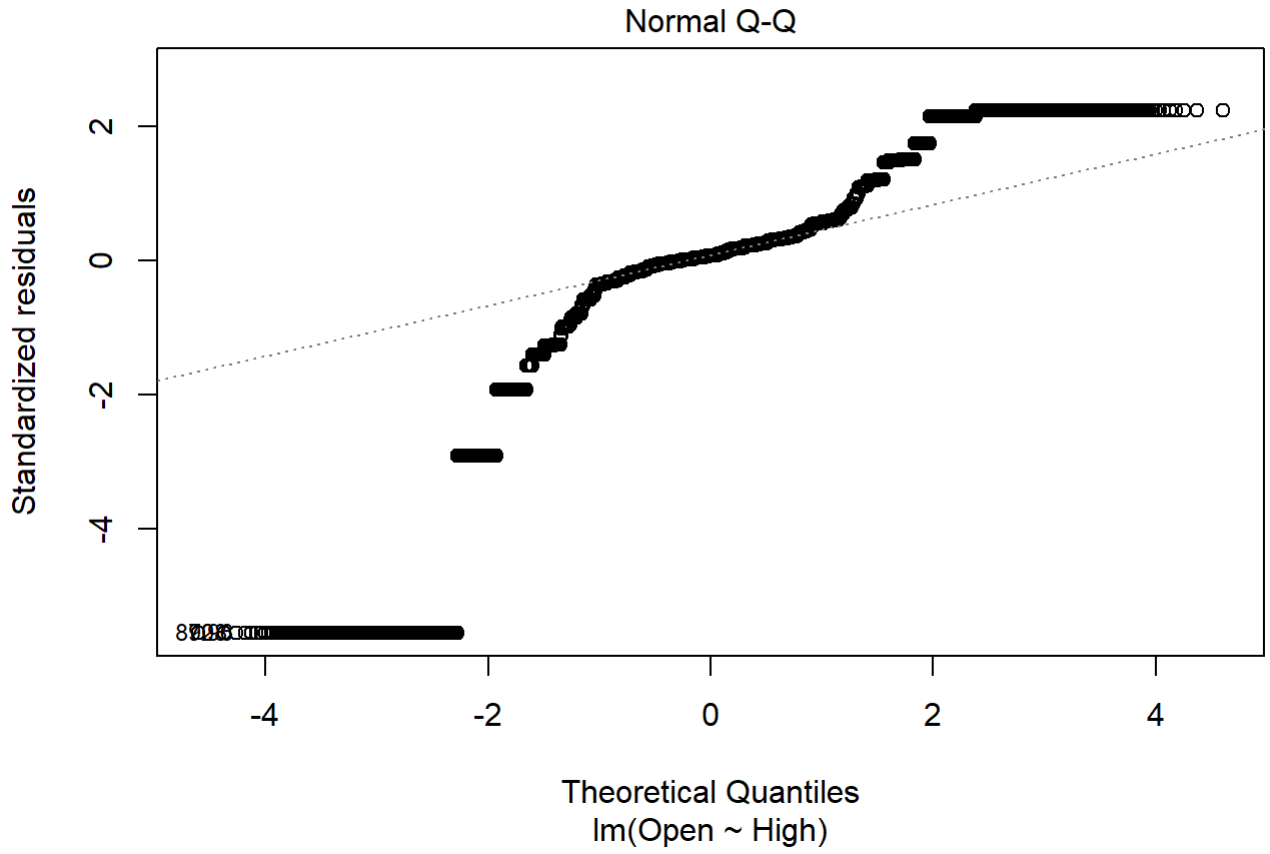


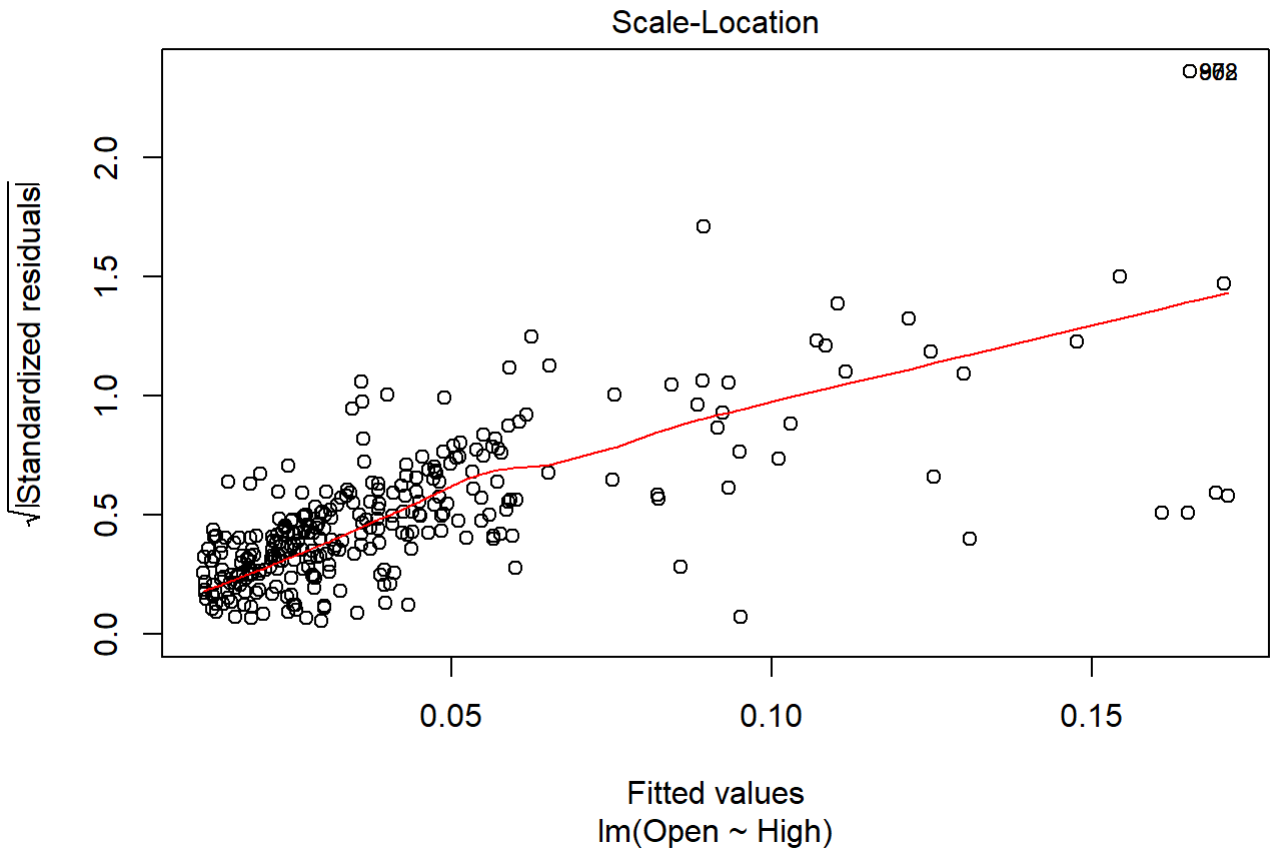

```
plot(Top_Buyers$Close, Top_Buyers$High, pch = 16, cex = 1.3, col = "blue", main = "Close vs Hig
h", xlab = "Open", ylab = "High")
abline(lm(Top_Buyers$Close ~Top_Buyers$High), col = 'red')
```

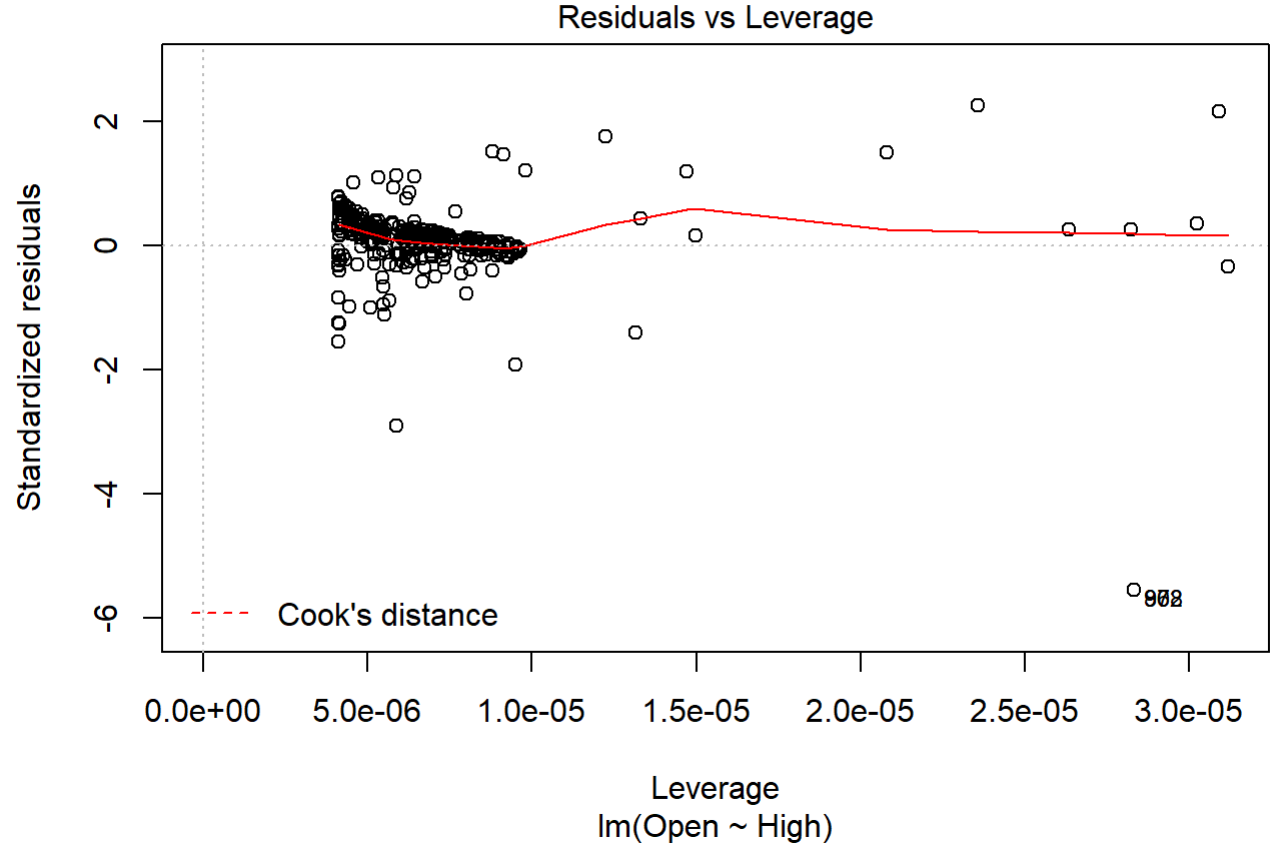



```
plot(linearModOH)
```

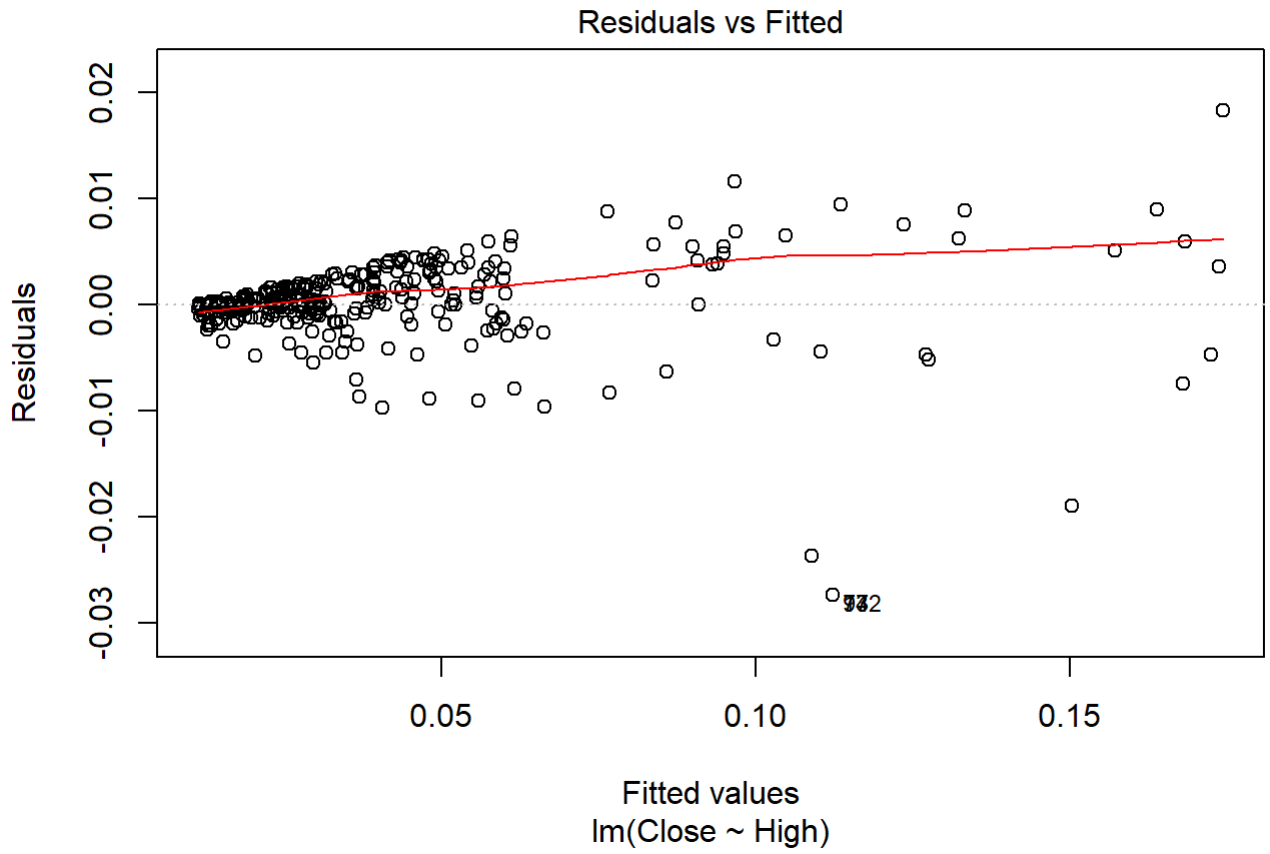

Residuals vs Fitted
Residuals
0.02
0.00
-0.02
-0.04
-0.06
0.05
0.10
0.15
Fitted values
lm(Open ~ High)

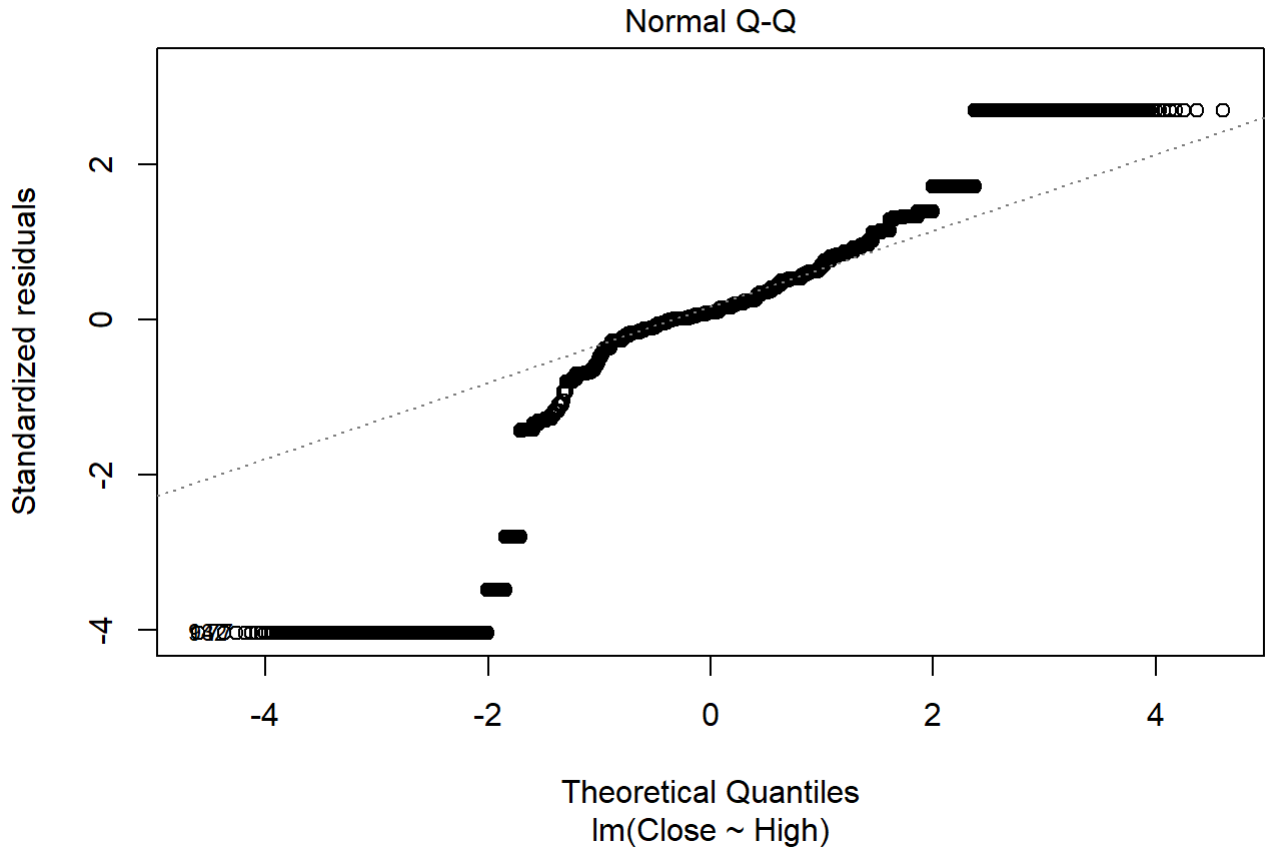

Normal Q-Q
Standardized residuals
2
0
-2
-4
-4
-2
0
2
4
Theoretical Quantiles
lm(Open ~ High)

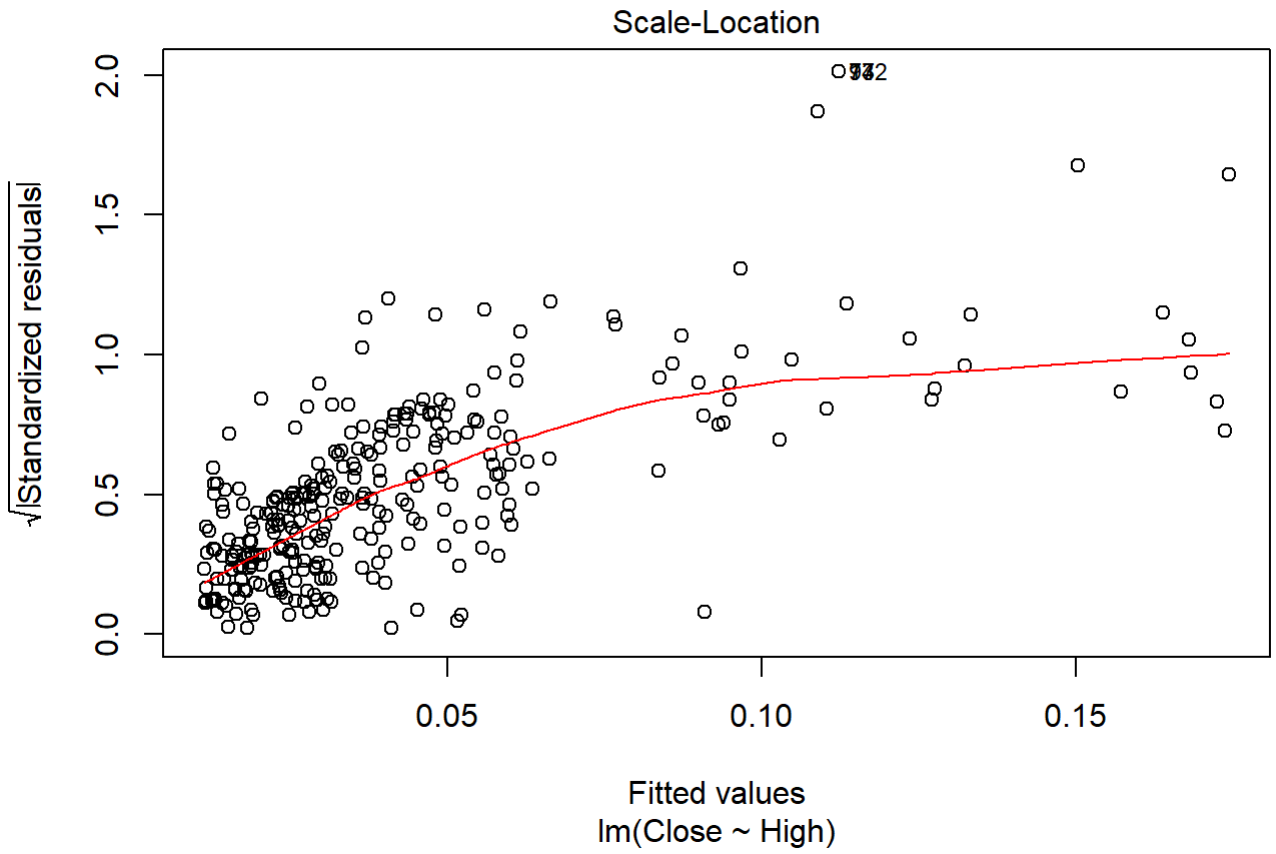

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.05
0.10
0.15
Fitted values
lm(Open ~ High)

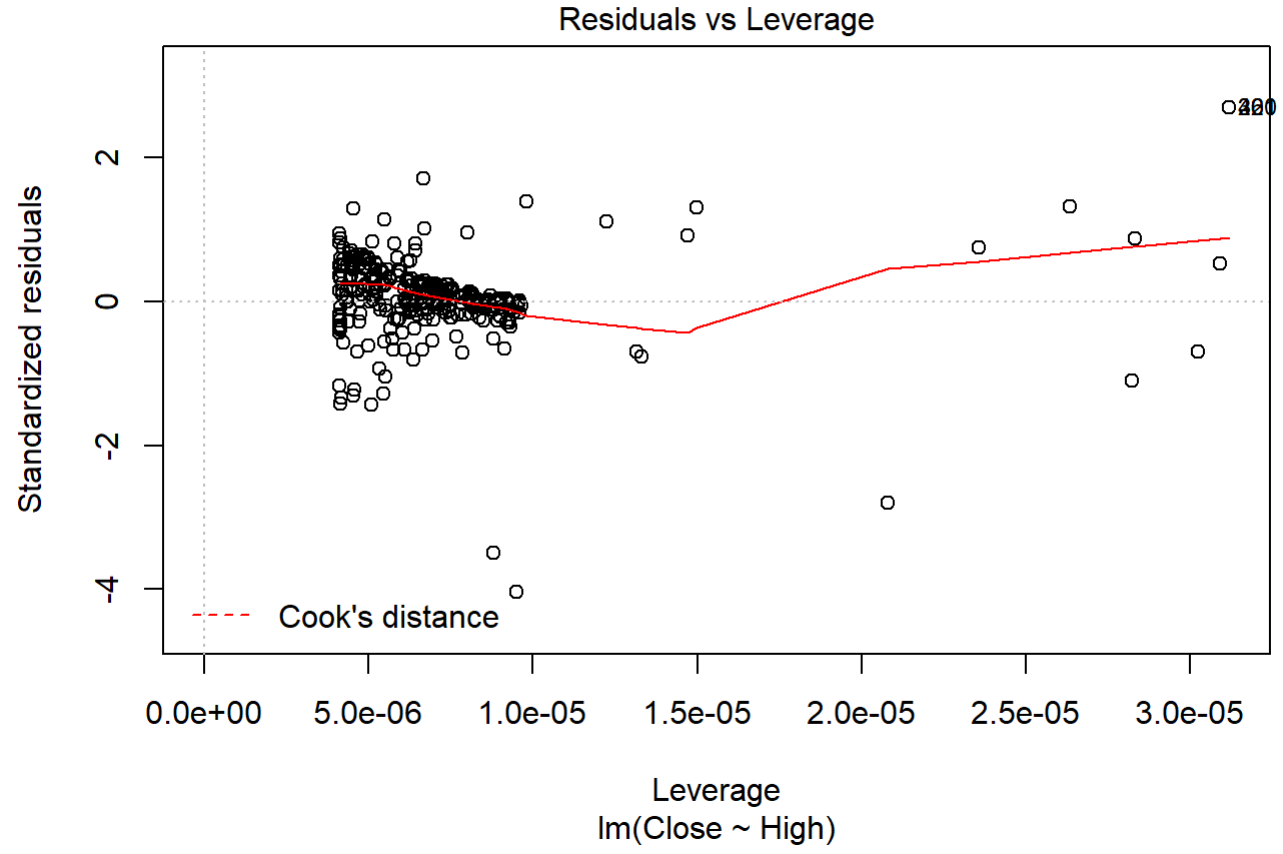

Residuals vs Leverage
Standardized residuals
2
0
-2
-4
-6
Cook's distance
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
2.5e-05
3.0e-05
Leverage
lm(Open ~ High)

```
plot(linearModCH)
```

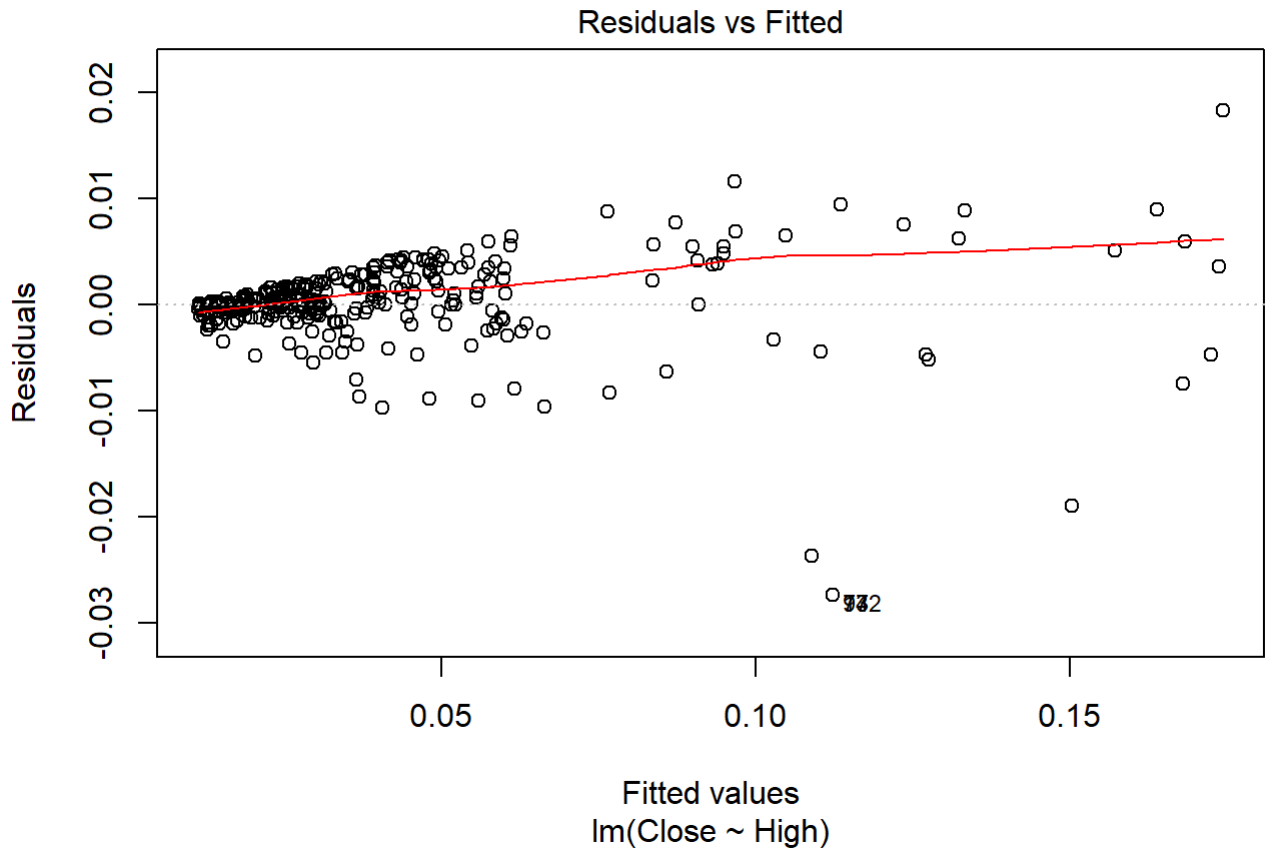

Residuals vs Fitted
Residuals
0.02
0.01
0.00
-0.01
-0.02
-0.03
0.05
0.10
0.15
Fitted values
lm(Close ~ High)

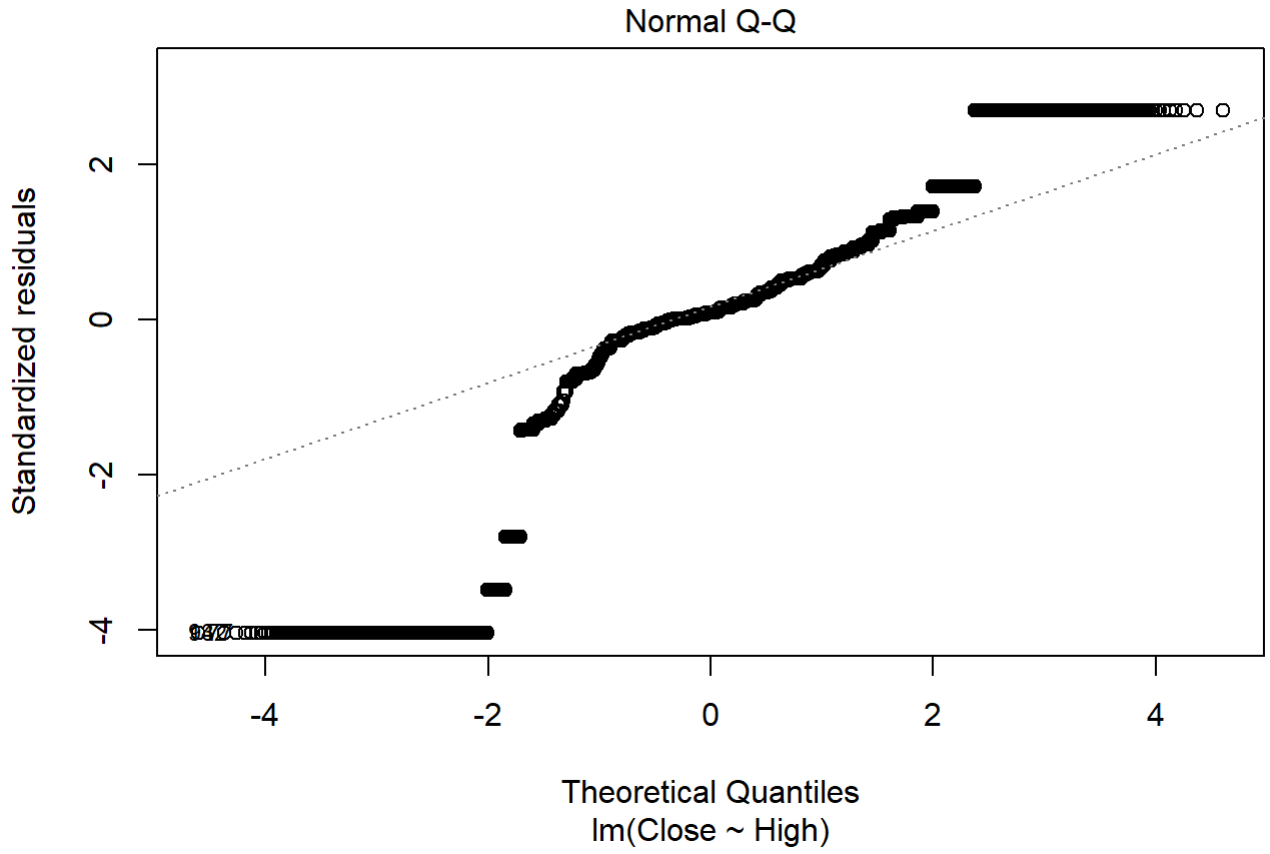

Normal Q-Q
Standardized residuals
2
0
-2
-4
-4
-2
0
2
4
Theoretical Quantiles
lm(Close ~ High)

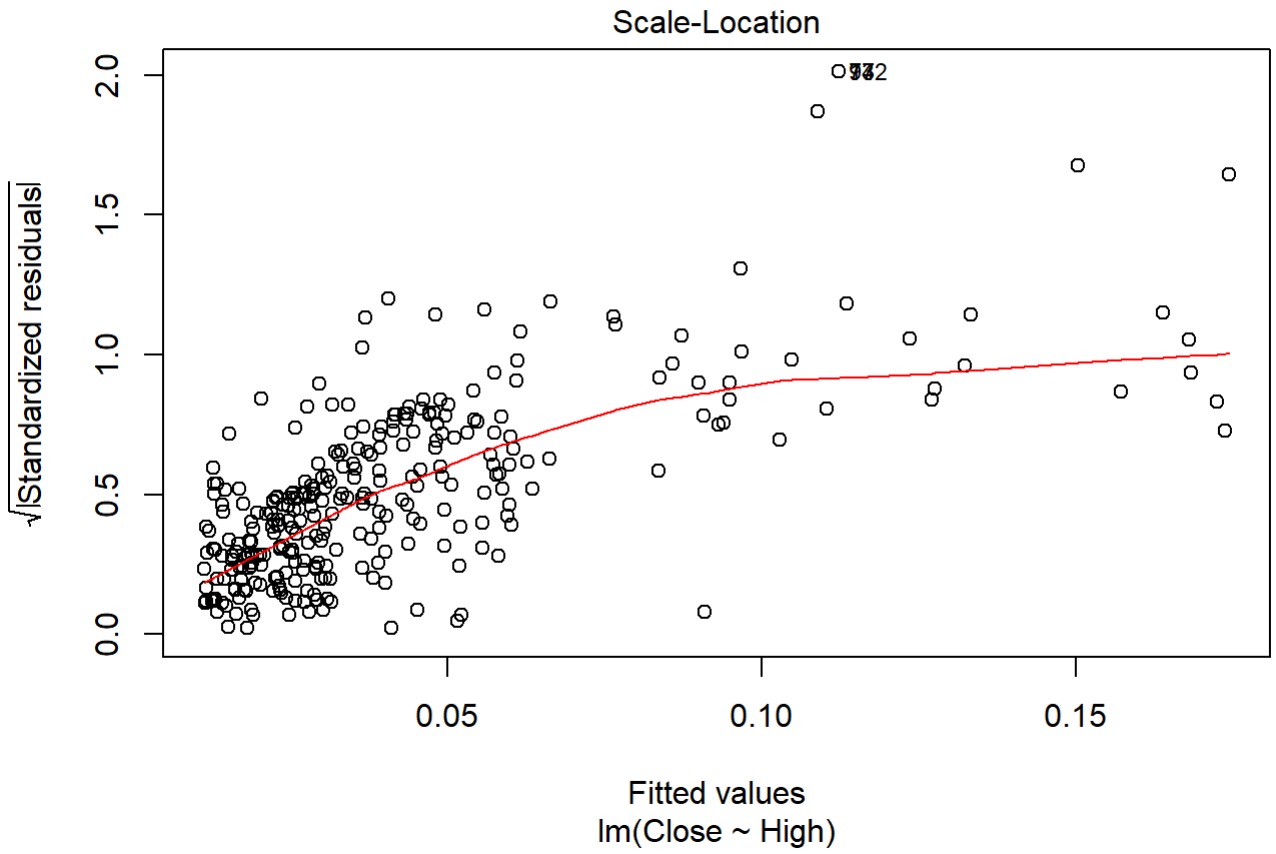

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.05
0.10
0.15
Fitted values
lm(Close ~ High)

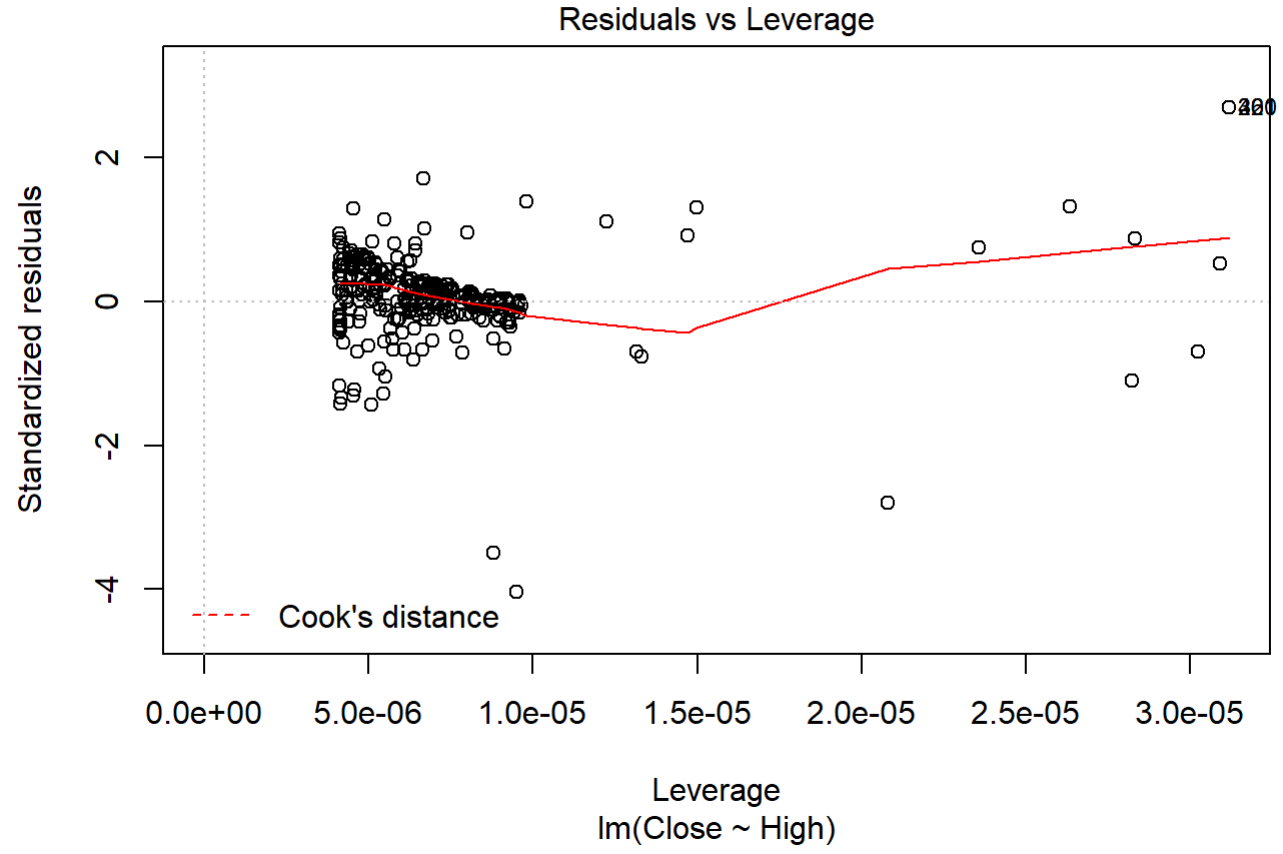

Residuals vs Leverage
Standardized residuals
2
0
-2
-4
Cook's distance
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
2.5e-05
3.0e-05
Leverage
lm(Close ~ High)

```
plot(linearModTM)
```


Residuals vs Fitted
Residuals
0.02
0.01
0.00
-0.01
-0.02
-0.03
0.05
0.10
0.15
Fitted values
lm(Close ~ High)


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


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
0.05
0.10
0.15
Fitted values
lm(Close ~ High)


Residuals vs Leverage
Standardized residuals
2
0
-2
-4
Cook's distance
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
2.5e-05
3.0e-05
Leverage
lm(Close ~ High)