**Assignment on Properties of Stock Returns**

1. *Check for the missing data and remove the respective entries from the dataset, if any. If you did find the missing data, make a comment on this in your report.*

I used daily data for TWTR. I checked the dataset for N/A data, nothing found. The length of N/A omitted dataset = length of started dataset = 5856.

1. *Compute and plot the simple returns on these securities against time. Use the adjusted price. Comment on the plot (how volatile the data are, volatility clustering, outliers etc).*

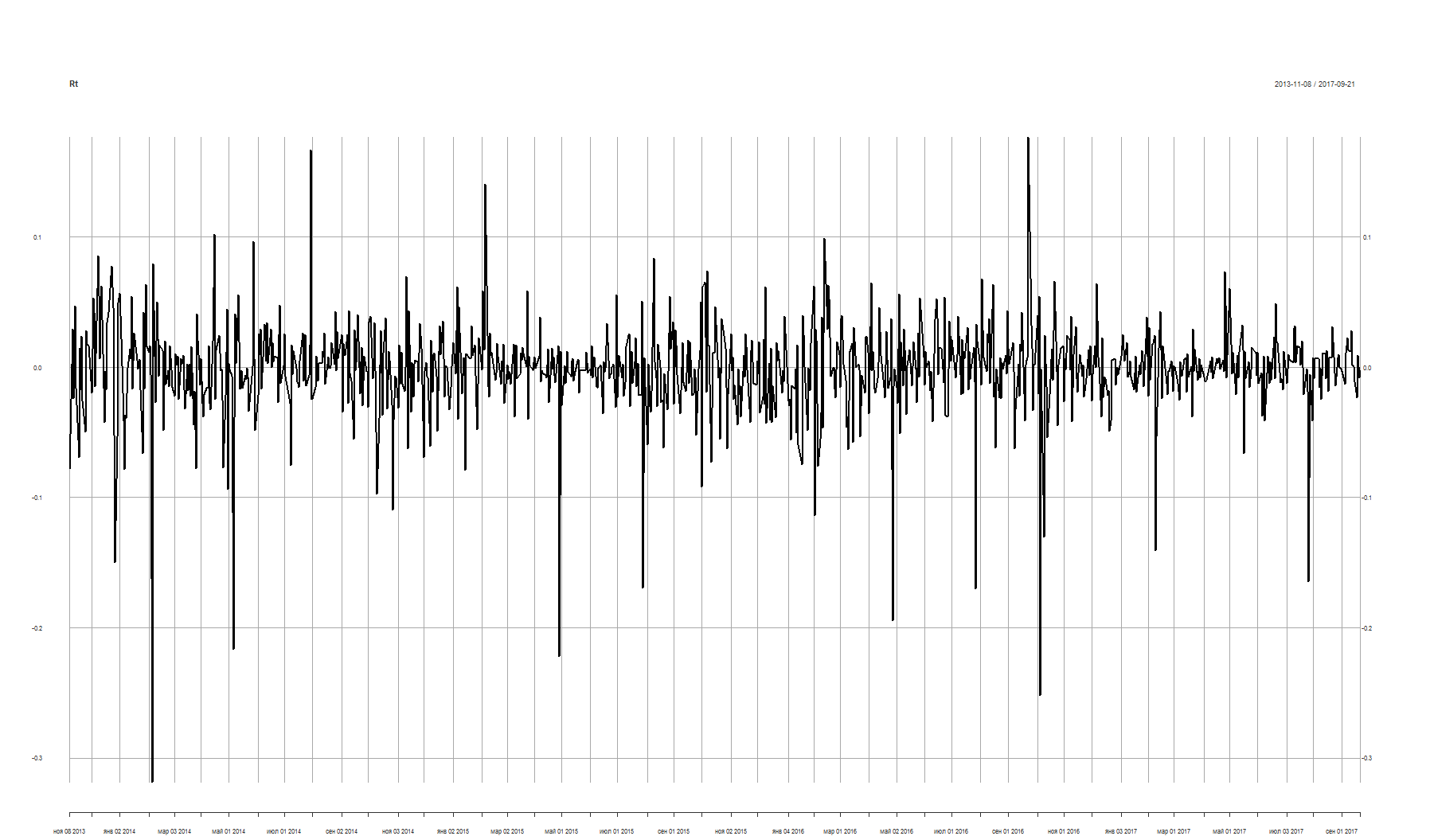
After slicing the dataset to one-dimension time series object (it contains only Adjusted price), the length of dataset decreased to 976 samples.

According to the plot 1, there are some outliers in the dataset. The smallest return was at 2014-02-06, when Twitter Stock Drops 31.8% after the first financial report after IPO. Twitter had a worth growth than expected. The highest return was at 2016-09-23 - +17.6% after @CNBC reported it may receive a bid for sale. Suitors reportedly included Google and Salesforce. In addition, a Stock history has more outliers, which can be seen at plot 1.

According plot, we can see 4 clusters of volatility:

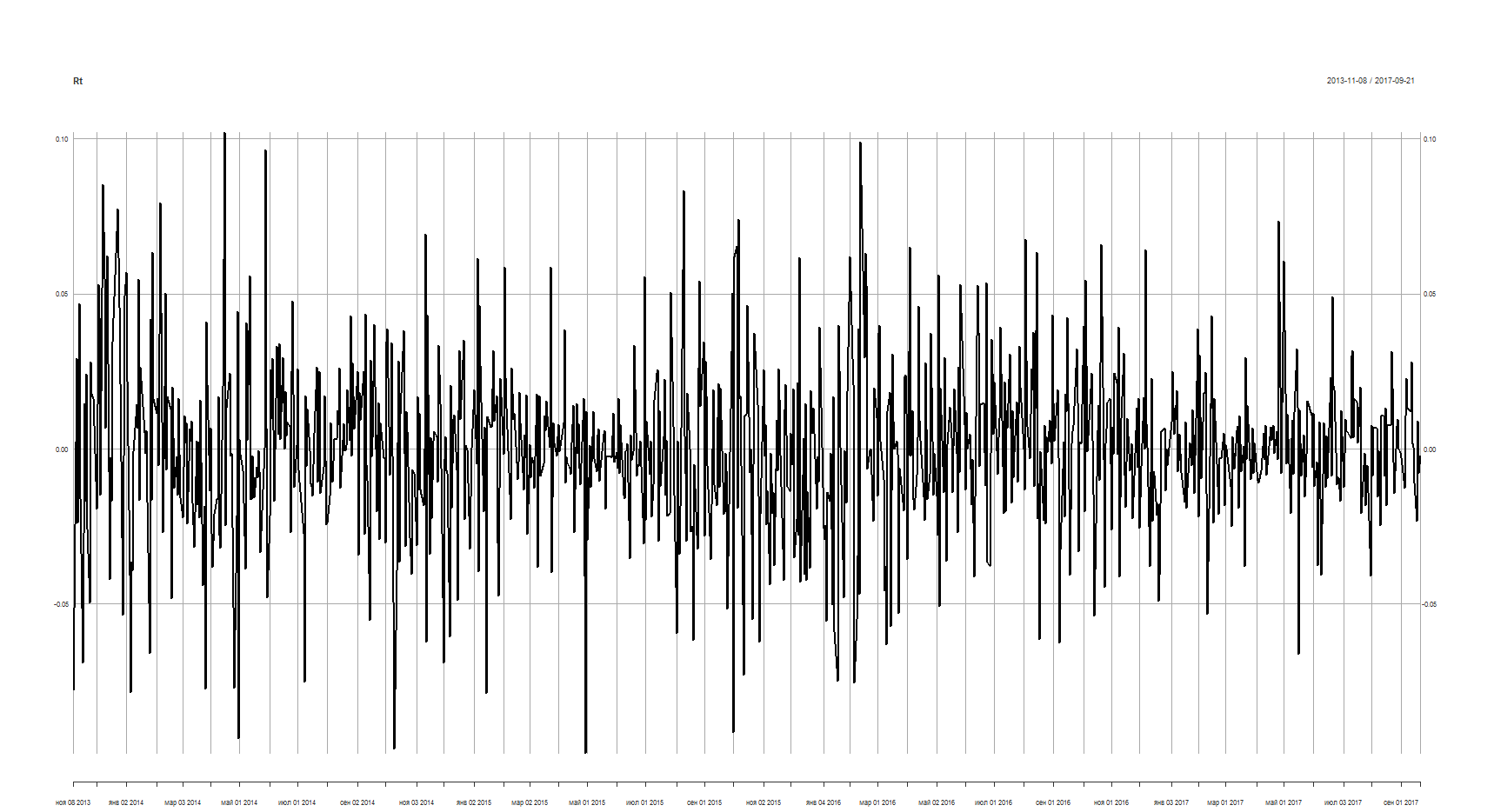
* the first – with high volatility from the start of observation up to 03-2015;
* the second – with low volatility: 03-2015 – 06-2015;
* the third – with high volatility 06-2015 – 01.2017;
* the fourth – with low volatility: 01.2017 – now.

*Plot 1. Simple returns*



Because this dataset contains many outliers, which can add distortion in the future analysis, I rejected some observation, according rule of 3 sigma. New dataset shown on plot 2. Deleted 18 observations.

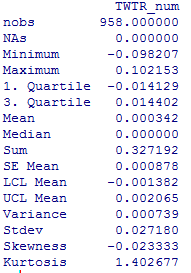
*Plot 2. Simple returns without outliers*



1. *Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, each simple return series.*

For Twtiter: mean = 0.003, sd = 0.027, skewness = -0.023, kurtosis = 1.402, minimum = -0.098 (outliers deleted, history minimum = -0.318), maximum = 0.1022 (outliers deleted, history maximum = 0.176).

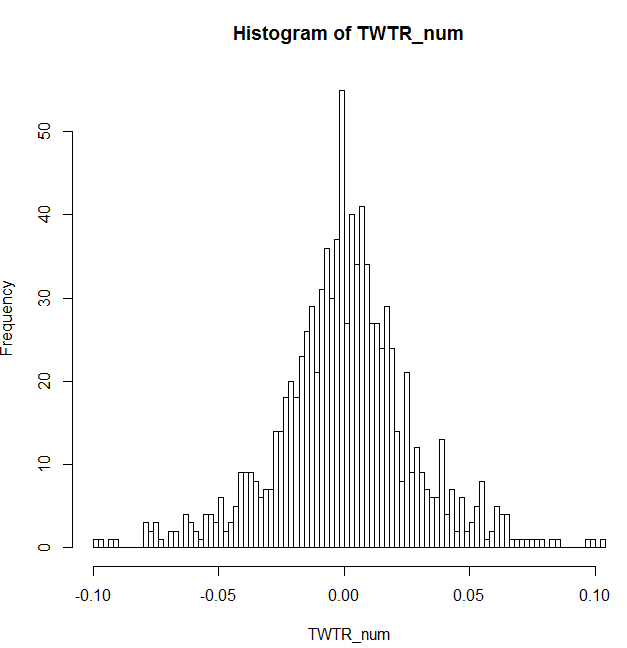
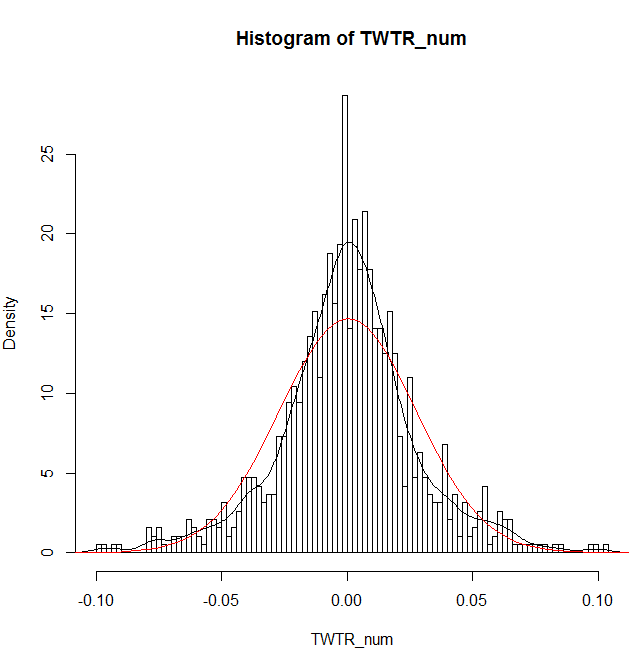
*Image 1. Twtiter summary*



1. *Obtain the histogram and the estimate of density of the simple returns for all time series. Impose the theoretical normal density on your histogram. Do the returns seem to be normally distributed?*

According plot 3, frequency histogram may seem as normal distribution, but after adding the theoretical line to the observable density histogram, can be seen that distribution is not normal.

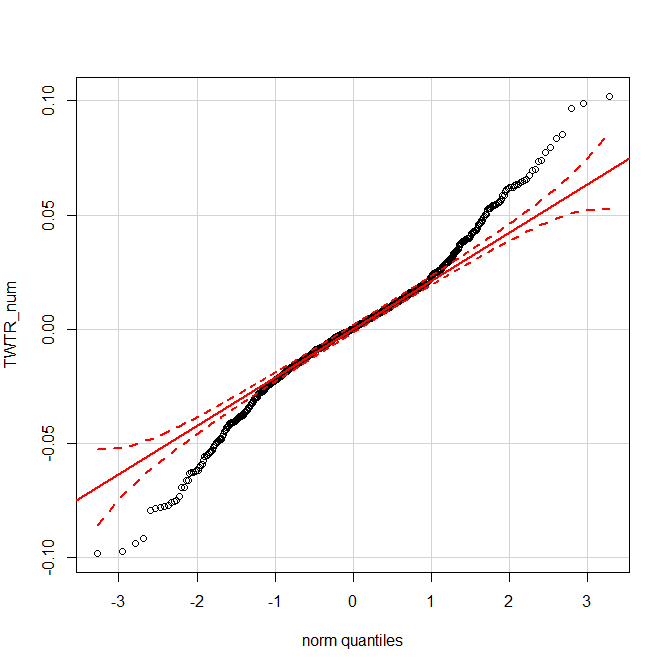
*Plot 3. Density and frequency histograms*

1. *Obtain a qq-plot using command qq.plot() from package car and comment on that*

According qq-plot (see plot 4), our observation is not normal (observation not in confidence interval).

*Plot 4. QQ-plot*



1. *Perform the Jarque-Bera normality test.*

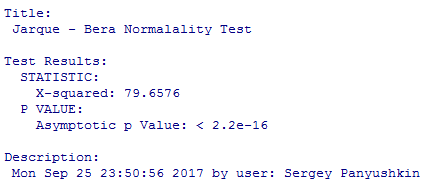
H0: S = 0, K = 3

H1: S != 0, K != 0

Jarque-Bera testing skewness and kurtosis equal to normal distribution. It is goodness-of-fit test that observations are normal.

p-value < 0.05 (image 2) and we should **reject** H0 – distribution is not normal.

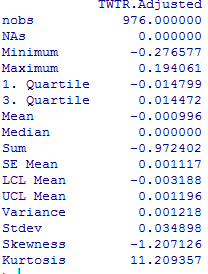
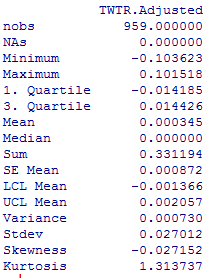
*Image 2. Jarque-Bera Normality Test results*



1. *Transform the simple returns to log returns. Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of each log return series. Comment on each statistic briefly.*

After transformation to log-returns the first observation get NaN value. It was omitted. Sample size decreased to 9 observation (in this dataset more samples, because outliers didn’t omit). As shown at image 3, the kurtosis much greater than in normal distribution (heavy tails), skewness not equal 0 (distribution mowed), also this dataset have a big stdev (0,034) – in 34 times greater, than mean (-0.001). Minimum is -0.28 and maximum is 0.194. I omitted outliers with 3-sigma rule. Sample size = 959. New sample information shown at image 3.

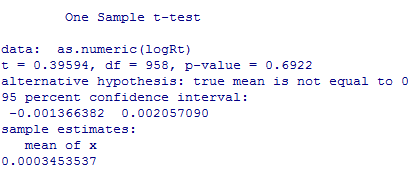
*Image 3. Log-returns info of source dataset and without outliers*

1. *Test the null hypothesis that the mean of the log returns is zero.*

One sample t-test: H0: mean is equal 0, H1: mean is not equal 0. We have p-value = 0.6922 and we can’t reject H0 – mean is equal 0.

*Image 4.* One sample t-test for log returns



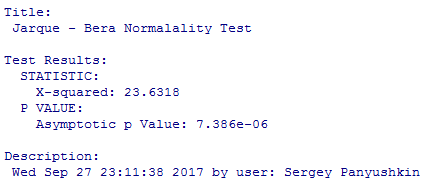
1. *Perform the Jarque-Bera normality test for log returns.*

H0: S = 0, K = 3

H1: S != 0, K != 0

p-value < 0.05 (image 5) and we should **reject** H0 – distribution, according Jarque-Bera test, is not normal.

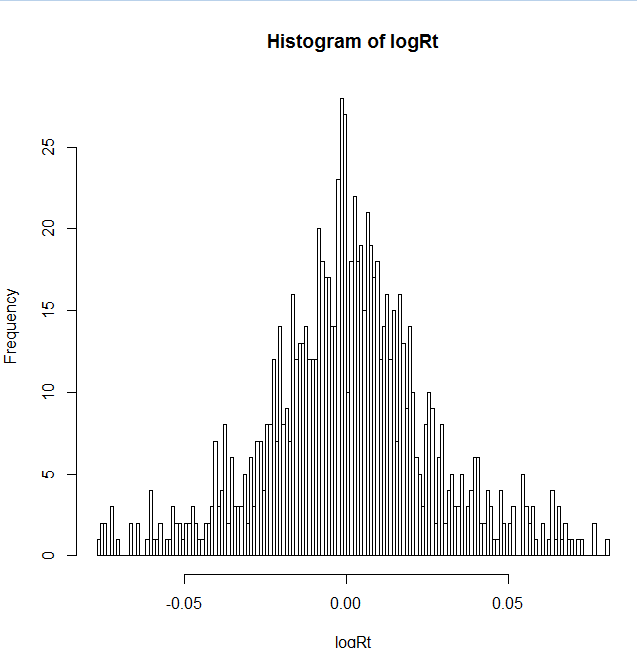
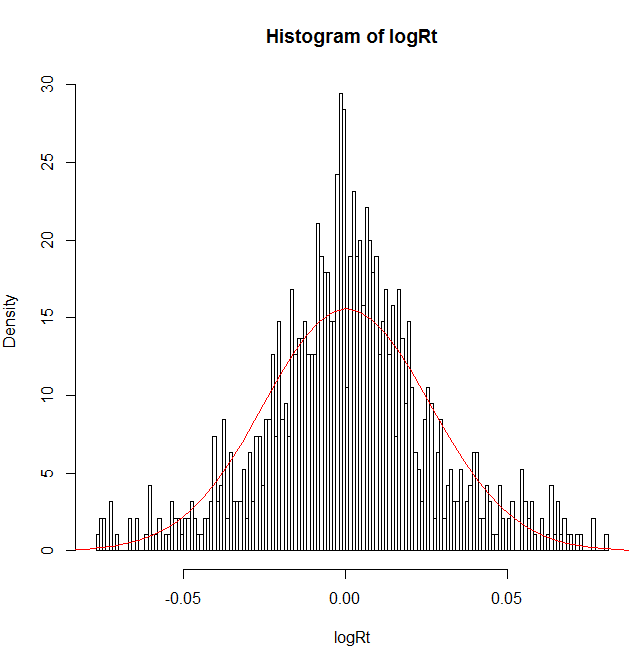
*Image 5.* Jarque-Bera test for log returns



1. *Obtain the histogram and the empirical density plot of the daily log returns.*

According plot 5, distribution far from normal, because observations near mean value are very frequently observed.

*Plot 5. Density and frequency histograms for log-returns*

1. *Compute the Pearson’s correlation, Kendall’s tau and Spearman’s rank correlations between the log returns of the three stocks and the S&P 500 index. Construct the respective scatter plots (see pp. 31-32,44 of [2]). Comment on the results.*

For correlation, I used started data (with outliers) without N/A values for TWTR, BIDU, GPRO and SPX. This companies have different IPO-date and I used data only for last 180 days – from 01.10.2017 to 09.26.2017.

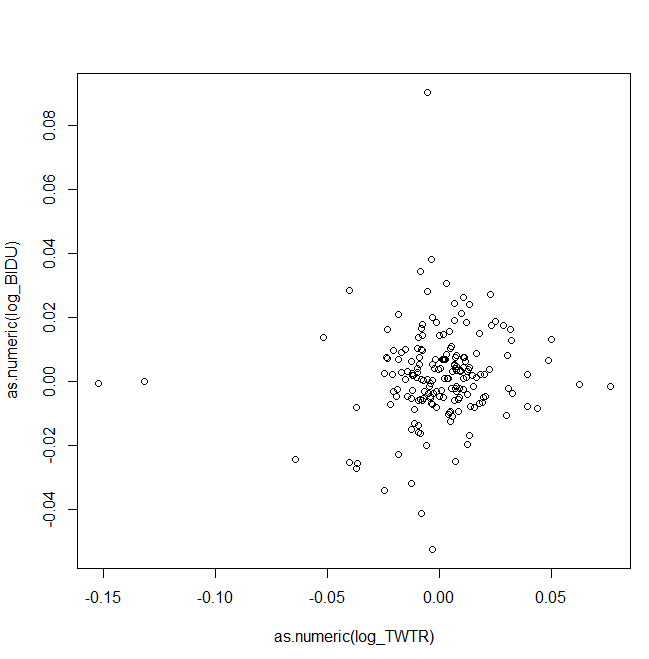
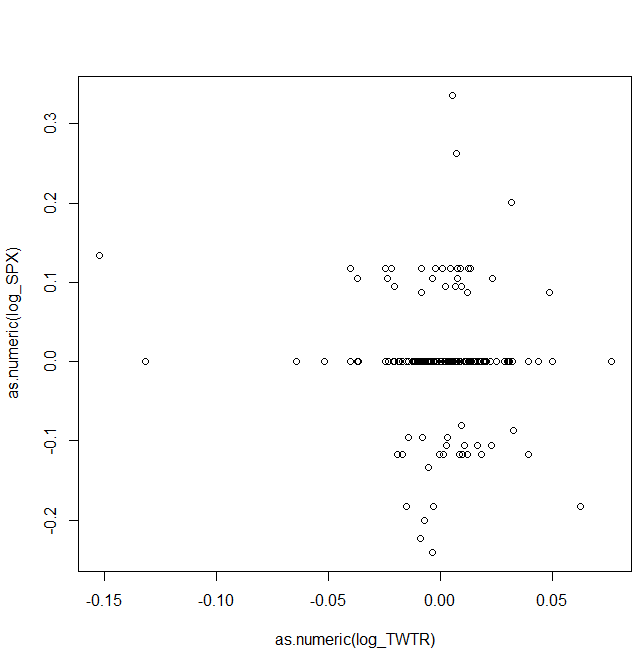
According the correlation matrix and Chaddok’s scale, all stocks have very low correlation. We should not use Pearson’s correlation, because samples are not normal and the Pearson’s criteria is parametric. We should use rang criteria – Kendall’s and Spearman’s.

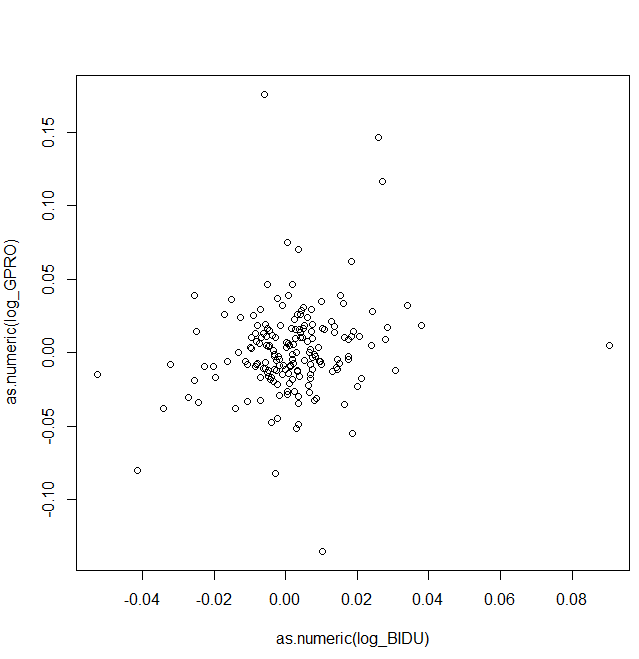
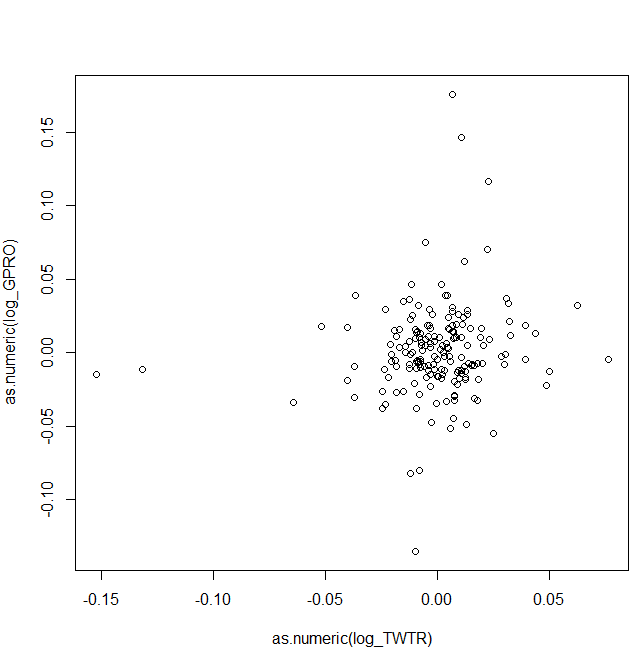
The lack or very low correlation is also noticeable on the scatter plot (see plot 6).

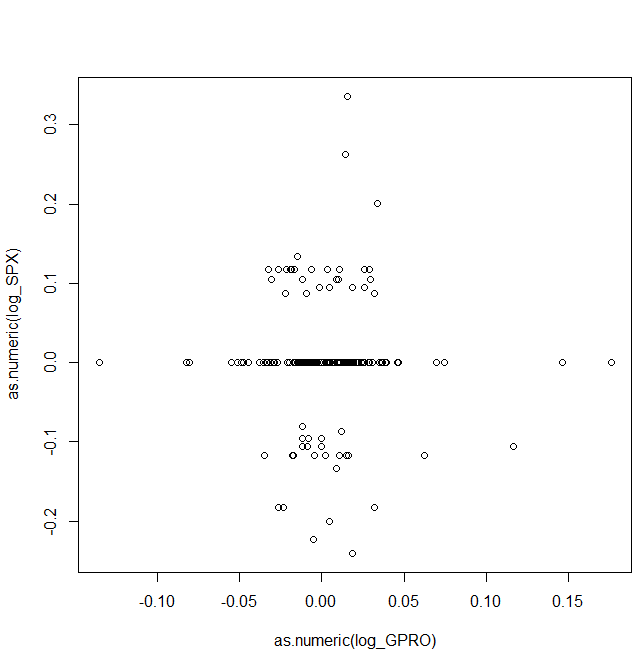
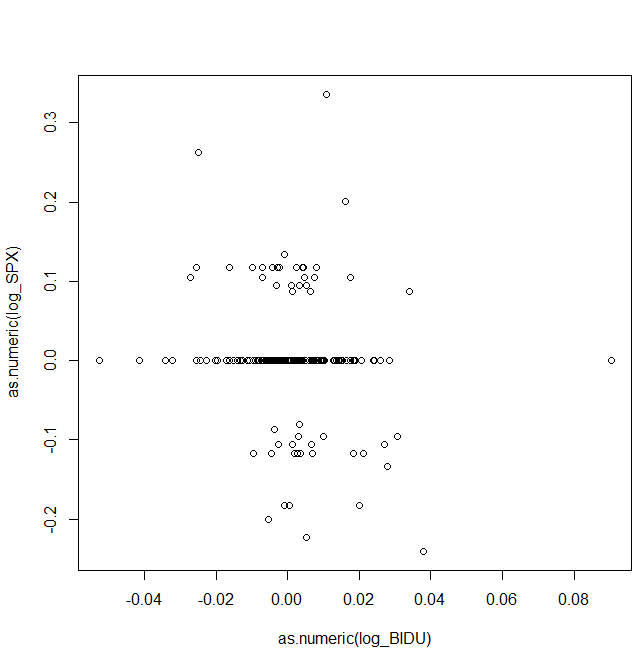
*Table 1. Correlation matrix*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Pearson’s** | | | | **Kendall’s** | | | | **Spearman’s** | | | |
|  | TWTR | BIDU | GPRO | SPX | TWTR | BIDU | GPRO | SPX | TWTR | BIDU | GPRO | SPX |
| TWTR | 1 | 0.11 | 0.13 | -0.10 | 1 | 0.08 | 0.07 | -0.04 | 1 | 0.11 | 0.09 | -0.05 |
| BIDU | 0.11 | 1 | 0.19 | -0.16 | 0.08 | 1 | 0.11 | -0.11 | 0.11 | 1 | 0.16 | -0.14 |
| GPRO | 0.13 | 0.19 | 1 | -0.01 | 0.07 | 0.11 | 1 | 0.004 | 0.09 | 0.16 | 1 | 0.003 |
| SPX | -0.10 | -0.16 | -0.01 | 1 | -0.04 | -0.11 | 0.004 | 1 | -0.05 | -0.14 | 0.003 | 1 |

*Plot 6. Correlation between log returns of TWTR, BIDU, GPRO, SPX*







**Appendix. R-commands**

getSymbols('TWTR', src = "yahoo")

length(TWTR)

TWTR = na.omit(TWTR)

length(TWTR)

daAd = Ad(TWTR)

length(daAd)

daAd = na.omit(daAd)

length(daAd)

Rt = diff(daAd)[2:length(diff(daAd))]/daAd[1:length(daAd)-1]

plot(Rt)

Rt = Rt[(Rt < mean(Rt)+3\*sd(Rt)) & (Rt> mean(Rt)-3\*sd(Rt))]

TWTR\_num = as.numeric(Rt)

basicStats(TWTR\_num)

hist(TWTR\_num, breaks = 120)

hist(TWTR\_num, breaks = 120, prob = T)

curve(dnorm(x, mean(TWTR\_num), sd(TWTR\_num)),from = -0.2, to = 0.2, add = T, col = "red")

qqPlot(TWTR\_num)

normalTest(TWTR\_num, method = "jb")

logRt = diff(log(Ad(TWTR)))

logRt = na.omit(logRt)

basicStats(logRt)

logRt = logRt[(logRt < mean(logRt)+3\*sd(logRt)) & (logRt> mean(logRt)-3\*sd(logRt))]

basicStats(logRt)

t.test(as.numeric(logRt))

normalTest(logRt, method = "jb")

hist(logRt, breaks = 120)

hist(logRt, breaks = 120, prob = T)

curve(dnorm(x, mean(logRt), sd(logRt)),from = -0.2, to = 0.2, add = T, col = "red")

log\_TWTR = diff(log(Ad(TWTR)))

getSymbols('BIDU', src = "yahoo")

log\_BIDU = diff(log(Ad(BIDU)))

getSymbols('GPRO', src = "yahoo")

log\_GPRO = diff(log(Ad(GPRO)))

getSymbols('SPX', src = "yahoo")

SPX = na.omit(SPX)

log\_SPX = diff(log(Ad(SPX)))

log\_SPX = tail(log\_SPX, 180)

log\_GPRO = tail(log\_GPRO, 180)

log\_TWTR = tail(log\_TWTR, 180)

log\_BIDU = tail(log\_BIDU, 180)

cor(as.numeric(log\_TWTR), as.numeric(log\_SPX)) – for all assets

cor(as.numeric(log\_TWTR), as.numeric(log\_SPX), method = “spearman”) – for all assets

cor(as.numeric(log\_TWTR), as.numeric(log\_SPX), method = “kendall”) – for all assets

plot(as.numeric(log\_TWTR), as.numeric(log\_SPX)) – for all assets