

Tesla, Inc. Stocks Research

Solomiia Lenio Mykola Morhunenko

January 2020

Task description

Nowadays it is quite popular though that any person can make money on stock exchange - all you need is starter money and some knowledge.

We decided to check if it is possible to get a big revenue from some big companies without much effort.

Data description

For our research we first wanted to use the **New York Stock Exchange dataset** that consists of the historical data on New York Stock Exchange from 2012 until 2016 parced from Yahoo! Finance. But since tha data are quite outdated and we do not need the data on all the companies, we decided to get similar data. For our final dataset we downloaded the historical data on **Tesla, Inc.** stocks from Yahoo! Finanacefrom January 5, 2015 until January 4, 2020.

You can find the .csv file created [here](#).

We're going to load the data and see what we have.

```
tsla.data <- read.csv("TSLA-5.csv")
head(tsla.data)
```

##	Date	Open	High	Low	Close	Adj.Close	Volume
## 1	2015-01-05	214.55	216.50	207.16	210.09	210.09	5368500
## 2	2015-01-06	210.06	214.20	204.21	211.28	211.28	6261900
## 3	2015-01-07	213.35	214.78	209.78	210.95	210.95	2968400
## 4	2015-01-08	212.81	213.80	210.01	210.62	210.62	3442500
## 5	2015-01-09	208.92	209.98	204.96	206.66	206.66	4668300
## 6	2015-01-12	203.05	204.47	199.25	202.21	202.21	5950300

Here we have a data frame consisting of **7 different columns** and **1259** rows. What do these data mean? Each row in tha data frame represent a day, the stock was out on a market. The date of the day can be found in the **date** column, consisting of date objects.

All the follwoing columns, except for the last one, tell us about the **price** of the stock. The **open** and **close** columns tell the price of the stock when the market opened and closed on that day. The **low** and **high** columns consist of the lowest and the highest values the stock price reached during the day. **Adj.Close** column consists of the close price adjusted for both dividends and splits, although it almost always is the same as the close one.

The last column - **volume** - consists of integer values refering to the numbers of shares that have been bought and sold for the day.

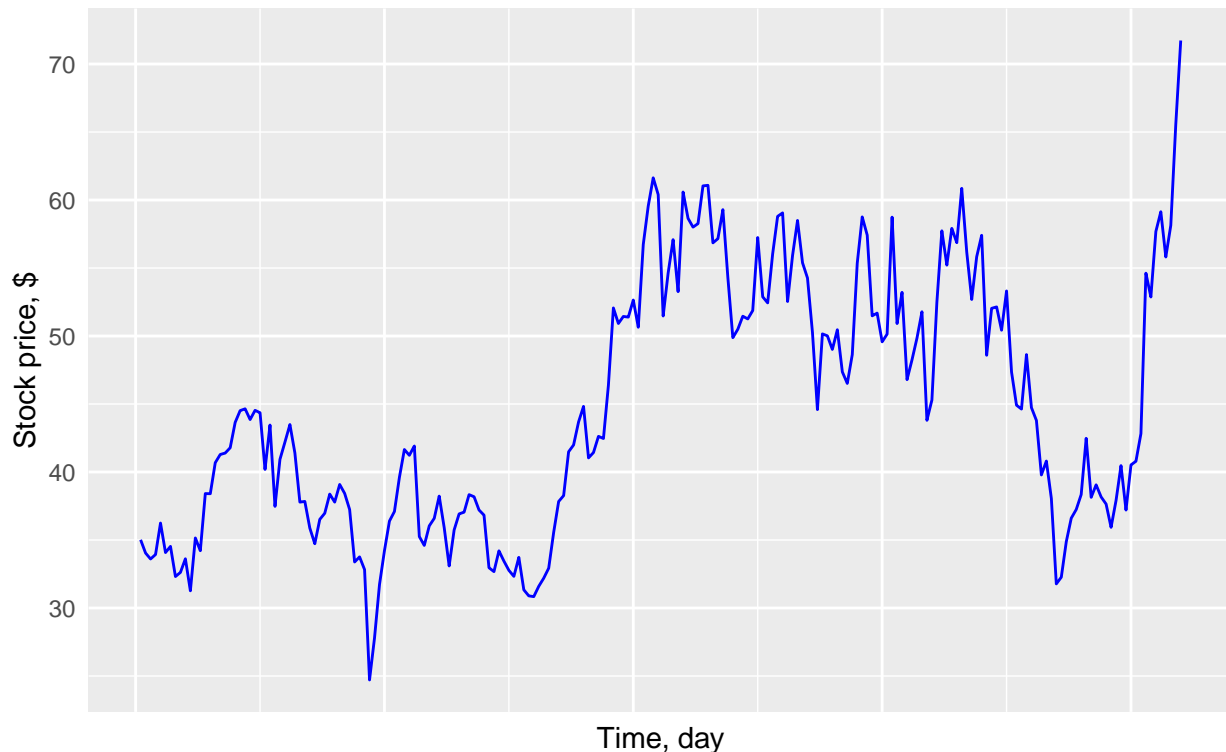
Data analysis

Let's first check how Tesla stock prices behaved themselves during the last five years.

For a better look of the plot we're going to plot every 6th day data, we'll show later why it doesn't change much.

Tesla, Inc. stock prices

based on data from Jan.5, 2015 until Jan.4, 2020



As we can see here, the stock prices have significantly increased during last months. Let's then check if we can make some good money here :)

Daily return

Since we wanted to check if we can make easy money on Tesla we'll test one of the easiest way - buying the shares one day and selling them the next day. Even easier way that does not involve any tracking - buying and selling at the same time.

For our case let's take the buying and selling time to be just before the market closes.

For this we'll need a **daily return** notion - the amount of stock price daily growth. Daily return is calculated by a simple intuitive formula

$$R_t = \frac{C_t - C_{t-1}}{C_{t-1}} = \frac{C_t}{C_{t-1}} - 1$$

```
tsla.data$PrevClose <- tsla.data$Close
# since there is no previous day for the first day, we'll get NA value
# in order not to get NA in further calculations we'll get rid of the first day
tsla.data <- na.omit(transform(tsla.data, PrevClose = c(NA, PrevClose[-nrow(tsla.data)])))
tsla.data$DailyRet <- with(tsla.data, Close / PrevClose - 1)
```

```
head(tsla.data)
```

##	Date	Open	High	Low	Close	Adj.Close	Volume	PrevClose
## 2	2015-01-06	210.06	214.20	204.21	211.28	211.28	6261900	210.09
## 3	2015-01-07	213.35	214.78	209.78	210.95	210.95	2968400	211.28
## 4	2015-01-08	212.81	213.80	210.01	210.62	210.62	3442500	210.95
## 5	2015-01-09	208.92	209.98	204.96	206.66	206.66	4668300	210.62
## 6	2015-01-12	203.05	204.47	199.25	202.21	202.21	5950300	206.66
## 7	2015-01-13	203.32	207.61	200.91	204.25	204.25	4477300	202.21

```
##      DailyRet
## 2  0.005664254
## 3 -0.001561918
## 4 -0.001564361
## 5 -0.018801591
## 6 -0.021532938
## 7  0.010088487
```

Now we'll see what we can say on the daily return values we got.
We can start off with some simple numeric characteristics.

```
cat("mean value: ", mean(tsla.data$DailyRet), " ")
```

```
## mean value:  0.000991882
```

```
cat("max value: ", max(tsla.data$DailyRet), " ")
```

```
## max value:  0.1766923
```

```
cat("min value: ", min(tsla.data$DailyRet), "\n")
```

```
## min value:  -0.1390154
```

```
cat("standart deviation: ", sd(tsla.data$DailyRet))
```

```
## standart deviation:  0.02829682
```

Looking at these numbers we already can say that day-to-day changes in the stock prices are not that significant. Small standart deviation tells us that variety of all the values is quite small meaning that day-to-day buying-selling is not a risky strategy.

It drives us to a conclusion that the strategy of selling the shares just the next day is **not the best strategy for a good money income, although it's a safe one** - you will not loose much if something goes wrong.

Annual return

Still, daily return can give even more information on some company stock prices - daily return can be used to derrieve **annual return**.

Annual return is the approxiamation of finance return in a year. It can be calucated with a single daily return value by the following formula

$$Y_t = (R_t + 1)^{365} - 1$$

We're now going to add one more column with calculated annual return.

```
tsla.data$YearRet <- with(tsla.data, power(DailyRet + 1, 365) - 1)
head(tsla.data)
```

```
##      Date   Open   High   Low  Close Adj.Close  Volume PrevClose
## 2 2015-01-06 210.06 214.20 204.21 211.28   211.28 6261900   210.09
## 3 2015-01-07 213.35 214.78 209.78 210.95   210.95 2968400   211.28
## 4 2015-01-08 212.81 213.80 210.01 210.62   210.62 3442500   210.95
## 5 2015-01-09 208.92 209.98 204.96 206.66   206.66 4668300   210.62
## 6 2015-01-12 203.05 204.47 199.25 202.21   202.21 5950300   206.66
## 7 2015-01-13 203.32 207.61 200.91 204.25   204.25 4477300   202.21
##      DailyRet   YearRet
## 2  0.005664254  6.8586848
## 3 -0.001561918 -0.4347831
## 4 -0.001564361 -0.4352877
## 5 -0.018801591 -0.9990200
## 6 -0.021532938 -0.9996457
## 7  0.010088487 38.0111416
```

Now we can analyze what data we get.
Again we'll start with some simple characteristics.

```
## [1] -1
```

```
cat("mean value: ", mean(tsla.data$YearRet), " ")
```

```
## mean value: 6.782408e+22
```

```
cat("max value: ", max(tsla.data$YearRet), " ")
```

```
## max value: 6.193926e+25
```

```
cat("min value: ", min(tsla.data$YearRet))
```

```
## min value: -1
```

The data we got seem quite interesting and not that self-explanatory.

```
cat("standart deviation: ", sd(tsla.data$YearRet))
```

```
## standart deviation: 1.860227e+24
```

Standart deviation here is not that small which tells us that there is a variety in the annual return values.

```
cat("skewness: ", skewness(tsla.data$YearRet), " ")
```

```
## skewness: 30.73026
```

```
cat("kurtosis: ", kurtosis(tsla.data$YearRet))
```

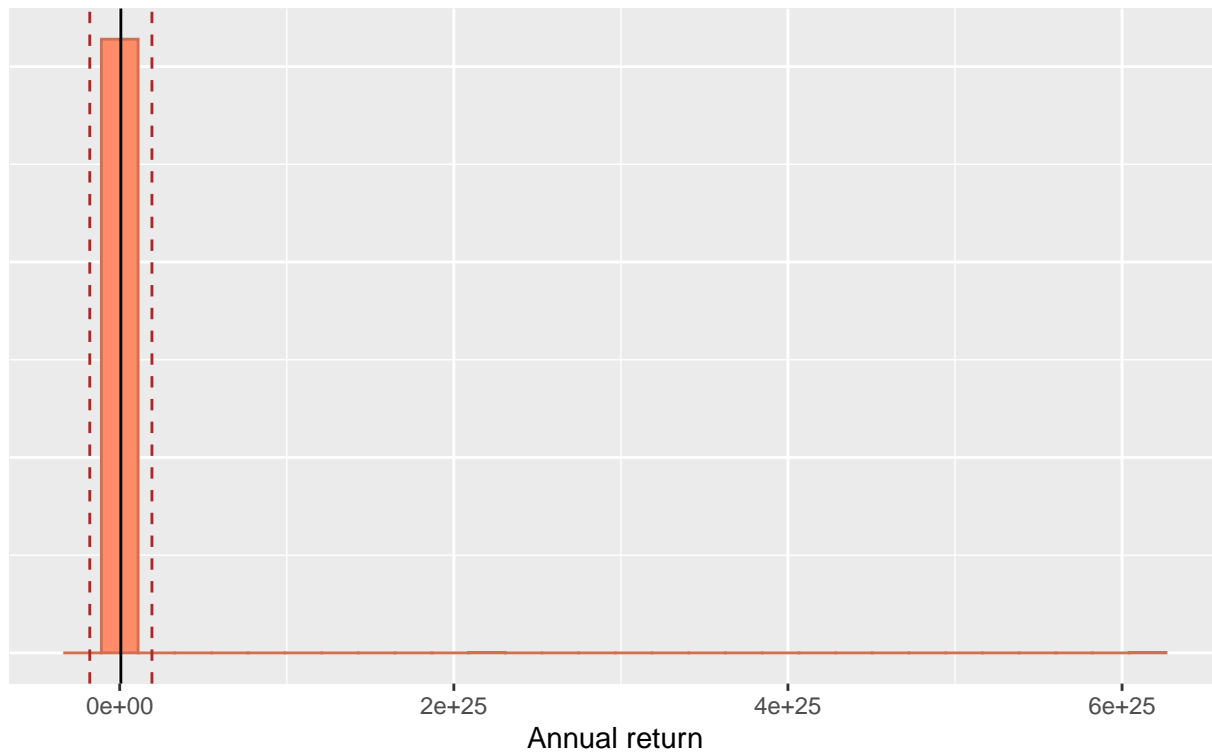
```
## kurtosis: 992.0115
```

Positive skewness tells us that the distribution of annual return has a heavy right-tail. In terms of the return itself it means that the values are mostly concentrated around the expected value, although there is a small probability to get values much bigger than the expected one.

Positive kurtosis means that the distribution has so-called thick tails. The high value of kurtosis tells about frequent extreme values in the distribution. In terms of the data we have it again means that there are a lot of extreme values for the annual return.

For further analysis we'll visualize all the data we got on annual return.

Histogram of annual returns on Tesla, Inc. stock prices based on data from Jan.6, 2015 until Jan.4, 2020

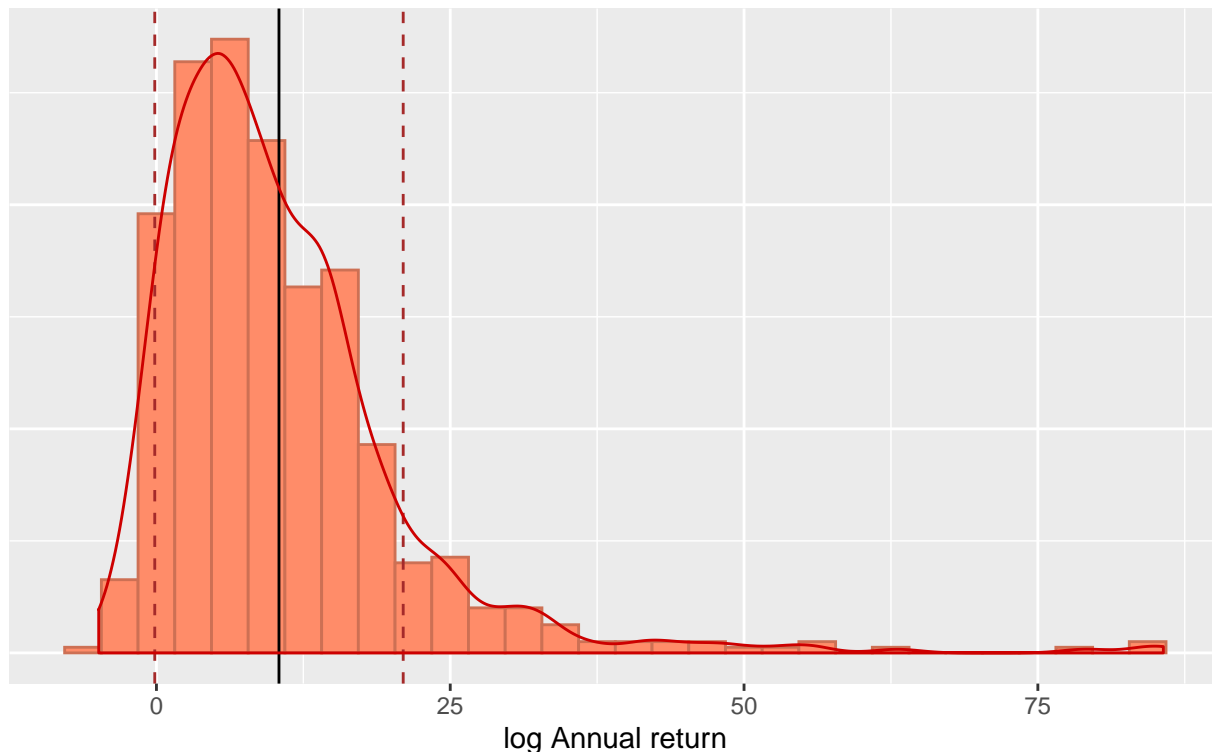


As we can see the data turned to have some quite extreme values and the plot does not help much. In order to fix this we're going to take the logarithm of the annual return, since we're working with ratios, and plot the data we got.

```
log.ret <- with(tsla.data, log2(YearRet))
log.ret <- log.ret[!is.nan(log.ret) & !is.infinite(log.ret)]
head(log.ret)

## [1]  2.777932  5.248350  3.126778 12.578770 13.357224 13.583587
```

Histogram of log of annual returns on Tesla, Inc. stock prices based on data from Jan.6, 2015 until Jan.4, 2020



It would be expected to suggest that annual return follows a lognormal distribution but since it assumes both positive and negative values lognormal distribution is not suitable here.

Our next suggestion here is **logistic distribution**. By its shape it may resemble a normal distribution although it has heavier tail - meaning higher kurtosis.

In order to check it we're going to run **Kolmogorov-Smirnov test for goodness-of-fit**.

H_0 : annual returns follow distribution F , being s logistic distribution with parameters $\mu = \hat{m}_1$ and $s = \frac{\sqrt{3}\hat{s}_d}{\pi}$
 H_1 : annual return does not follow distribution F

```
ks.test(tsla.data$YearRet, "plogis", location=mean(tsla.data$YearRet), scale=sqrt(3) * sd(tsla.data$YearRet))
```

```
## Warning in ks.test(tsla.data$YearRet, "plogis", location =
## mean(tsla.data$YearRet), : ties should not be present for the Kolmogorov-
## Smirnov test
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data:  tsla.data$YearRet
## D = 0.51414, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

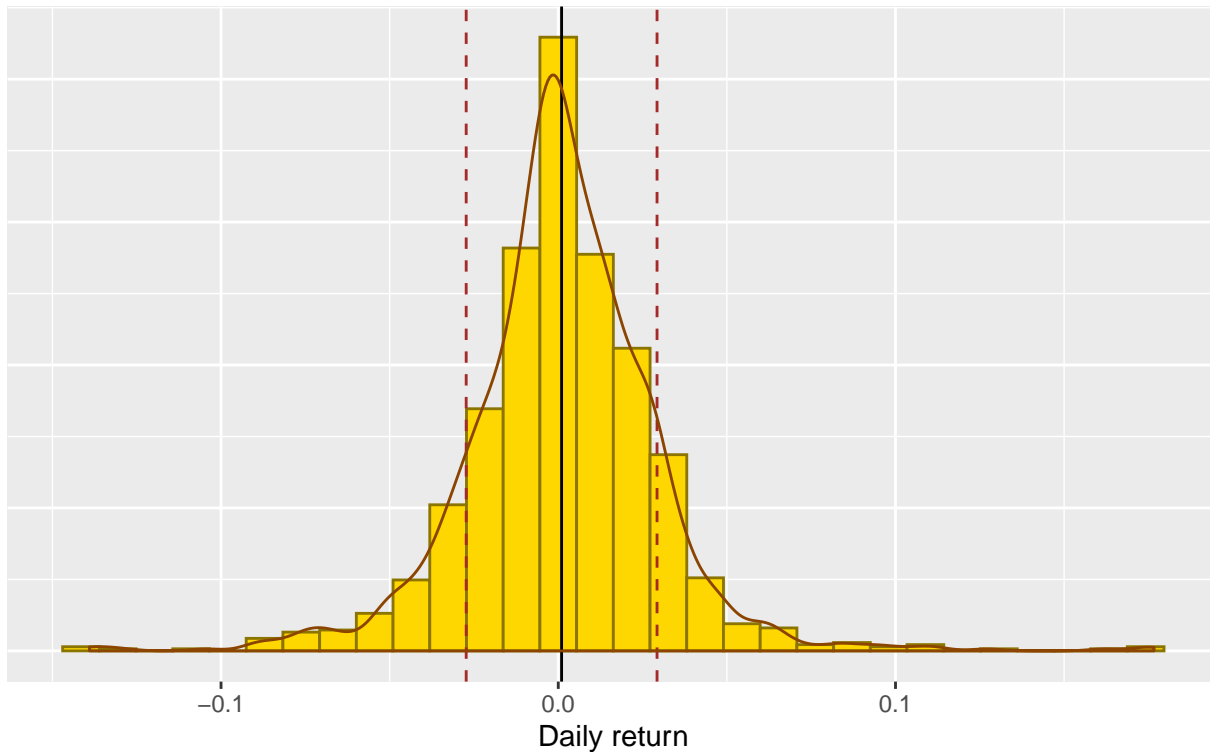
p-value of the test is almost zero which means that our guess was wrong and we reject H_0 .

In order to understand the distribution we'll go back to how we calculated it.

We used a formula $Y_t = (R_t + 1)^{365} - 1$, where R_t was the value of a daily return.

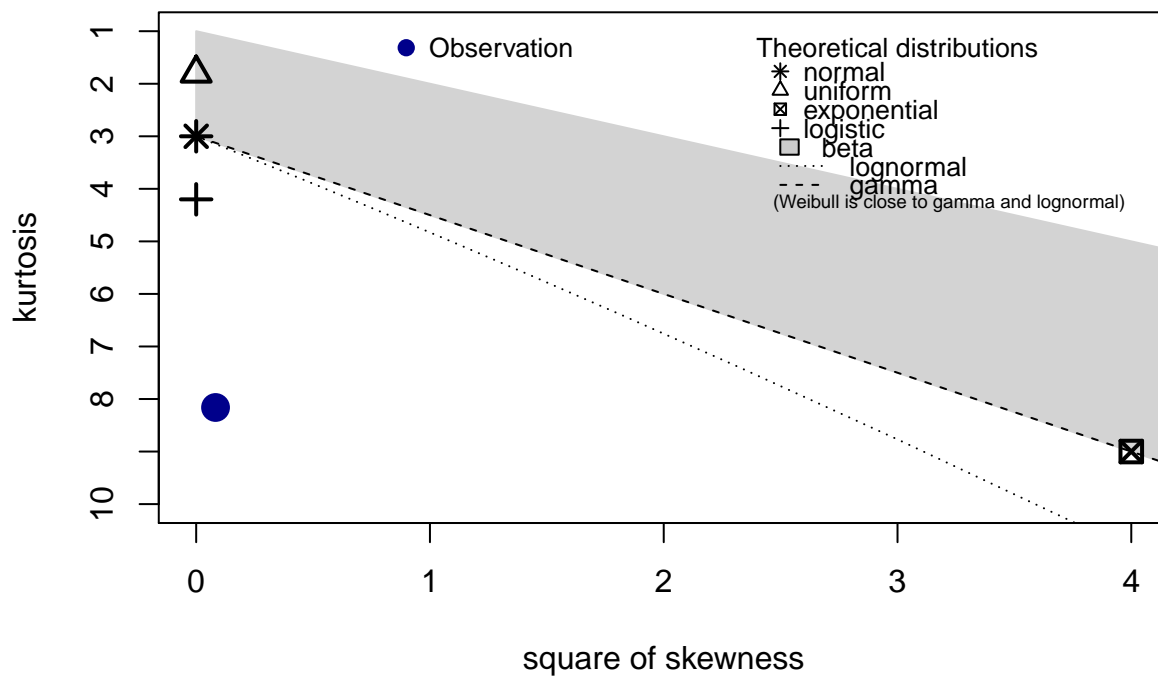
We'll plot daily return data and try to fit them to some distribution in order to get some idea of how annual return is distributed.

Histogram of daily returns on Tesla, Inc. stock prices
based on data from Jan.6, 2015 until Jan.4, 2020



```
descdist(tsla.data$DailyRet)
```

Cullen and Frey graph

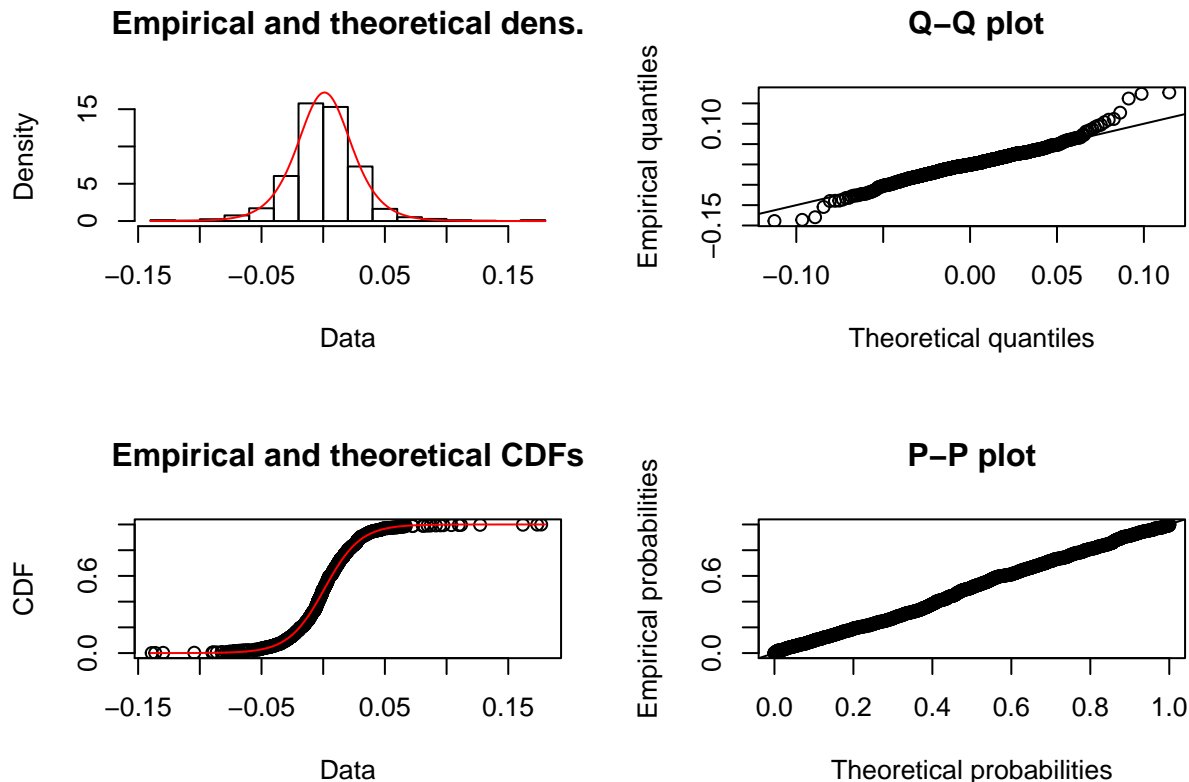


```
## summary statistics
```

```
## -----
## min: -0.1390154 max: 0.1766923
## median: 0.0004990788
## mean: 0.000991882
## estimated sd: 0.02829682
## estimated skewness: 0.2877739
## estimated kurtosis: 8.161372
```

It seems like daily returns might follow a logistic distribution.

```
fit.logis <- fitdist(tsla.data$DailyRet, "logis")
plot(fit.logis)
```



```
fit.logis
```

```
## Fitting of the distribution ' logis ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## location 0.00102485 0.0006989643
## scale    0.01450561 0.0003408756
```

Now we'll run a test.

H_0 : daily returns follow a distribution $F = \mathcal{L}(0.001, 0.01)$ H_1 : daily returns do not follow distribution F

```
ks.test(tsla.data$DailyRet, "plogis", location=fit.logis$estimate[1], scale=fit.logis$estimate[2])
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: tsla.data$DailyRet
## D = 0.032773, p-value = 0.1341
```



```
## alternative hypothesis: two-sided
```

p-value of the test is not that big, so in most cases we need to reject H_0 . Still the plots of fitting empirical data to theoretical ones look quite good, but following Cullen and Fray graph our data are not that close to actual theoretical logisitic distribution.

Still, there are no better estimates for the distribution, so we stick to **logistic distribution**.

In order to caculate annual return we raised daily return + 1 to the power of 365, basically we did multiplying. Prodcut of two and more i.i.d. r.v.s never has the same distribution. In fact, it has a **product distribution** except for some cases like lognormal distribution.

So, based on the way we calculated annual return, we make conclusion that our data on annual returns follow a **product distrubution**.

Conclusions

Dependences

Now we're going to check on some dependences between the data we have.

```
drops <- c("Date")
no.date <- tsla.data[ , !(names(tsla.data) %in% drops)]

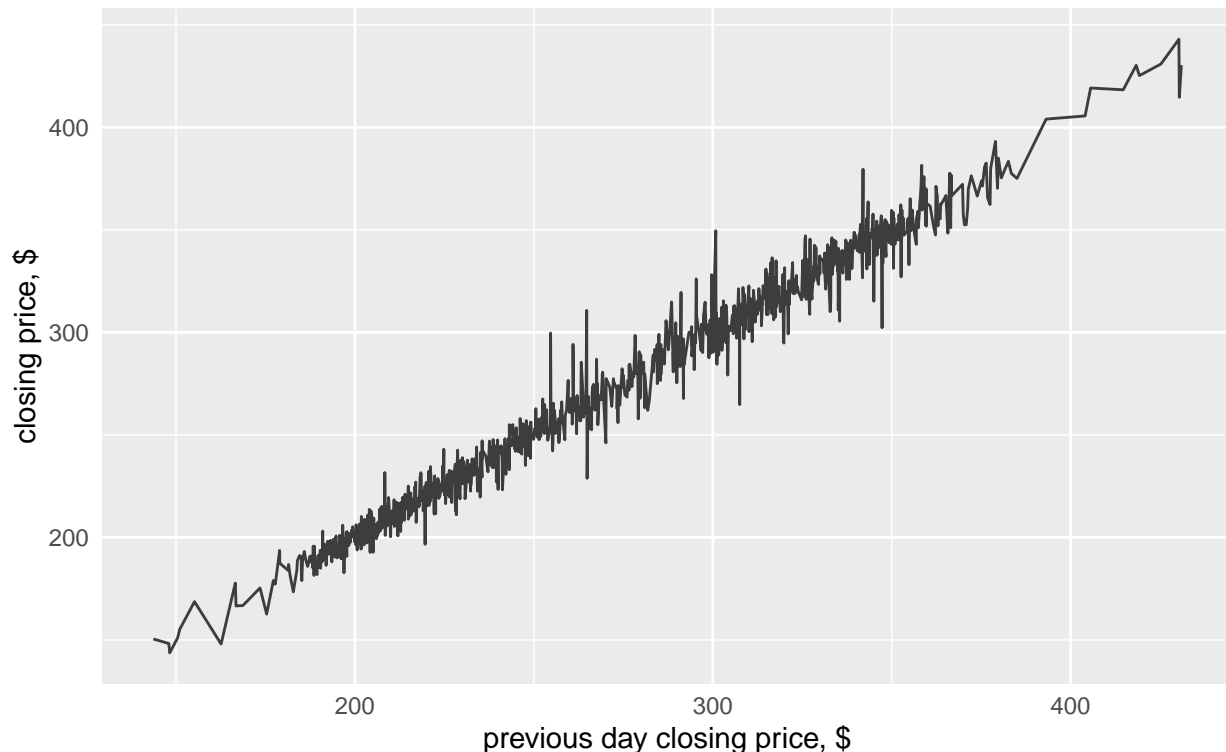
dependence <- cor(no.date)
dependence
```

##	Open	High	Low	Close	Adj.Close
## Open	1.0000000000	0.99724191	0.99725593	0.99372905	0.99372905
## High	0.9972419059	1.00000000	0.99681462	0.99761669	0.99761669
## Low	0.9972559295	0.99681462	1.00000000	0.99720342	0.99720342
## Close	0.9937290466	0.99761669	0.99720342	1.00000000	1.00000000
## Adj.Close	0.9937290466	0.99761669	0.99720342	1.00000000	1.00000000
## Volume	0.2377591753	0.26338293	0.21426069	0.24185432	0.24185432
## PrevClose	0.9963232739	0.99426028	0.99366462	0.99036152	0.99036152
## DailyRet	-0.0005577134	0.04162793	0.04185908	0.08487536	0.08487536
## YearRet	0.0203243315	0.02126841	0.01843548	0.02176967	0.02176967
##	Volume	PrevClose	DailyRet	YearRet	
## Open	0.237759175	0.996323274	-0.0005577134	0.020324332	
## High	0.263382927	0.994260277	0.0416279272	0.021268407	
## Low	0.214260688	0.993664618	0.0418590828	0.018435482	
## Close	0.241854316	0.990361518	0.0848753640	0.021769665	
## Adj.Close	0.241854316	0.990361518	0.0848753640	0.021769665	
## Volume	1.000000000	0.243700266	-0.0048235550	0.183205248	
## PrevClose	0.243700266	1.000000000	-0.0513504764	-0.007377567	
## DailyRet	-0.004823555	-0.051350476	1.0000000000	0.225323307	
## YearRet	0.183205248	-0.007377567	0.2253233069	1.000000000	

As we can see from the output the greatest correlation here can be seen between the values of low, high, close, open etc. columns. It was quite obvious to get such a result, since they all senote different values of the same, so to say, stock price function, that as we showed before do not change extremely in a short period of time. Still, we can make some use of these correlations. We can suggest a linnear relation between close and previous close price values in order to build a model for price predicions.

We're going to plot the data in order to see the actual dependency between them.

Tesla, Inc. closing stock prices dependence on previous data based on data from Jan.5, 2015 until Jan.4, 2020



It seems like there actually might be some linear relation between the values of close and previous close prices. A **linear regression** seems quite too simple model for such a case with stock prices, although we'll try it.

H_0 : there is no linear regression pattern in the stock prices behaviour H_1 : stock prices depend on the prices of the previous days and they follow a linear regression model

```
price.lm <- lm(tsla.data$Close~tsla.data$PrevClose)
```

```
summary(price.lm)
```

```
##
## Call:
## lm(formula = tsla.data$Close ~ tsla.data$PrevClose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -44.741  -3.548  -0.219   4.019  48.716
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.886647   1.077996     1.75  0.0803 .
## tsla.data$PrevClose 0.993677   0.003921  253.41 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.828 on 1256 degrees of freedom
## Multiple R-squared:  0.9808, Adjusted R-squared:  0.9808
## F-statistic: 6.422e+04 on 1 and 1256 DF, p-value: < 2.2e-16
```

p-value of the F-test is almost 0 which means that we **reject H_0 and there is linear relation**. Although residual standard error is quite large, as well as values of residuals statistics, to tell that model is good-fitted, as mentioned above - simple linear regression is still too simple model for this case.

Conclusions