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 But since tha data are quite outdated and we do not need the data on all the companies, we decided to get similar data.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.

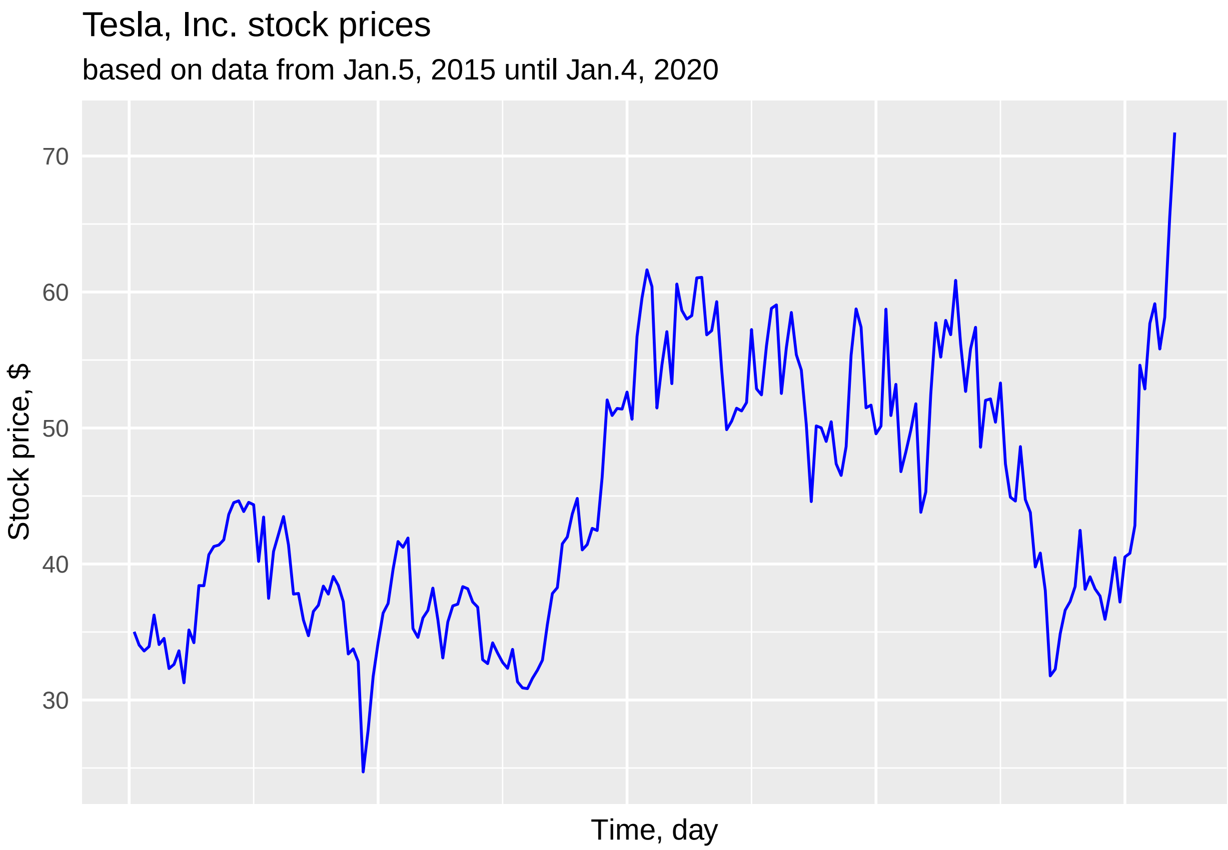
```
tsla.data <- read.csv("TSLA-5.csv")
head(tsla.data)
```

```
##         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 6^{th} day data, we'll show later why it doesn't change much.



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 envolve 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 characterics.

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

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

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

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

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

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

```r
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 conculsion 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 derrive **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.

```r
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
```

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

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

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

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

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

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

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

```r
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.

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

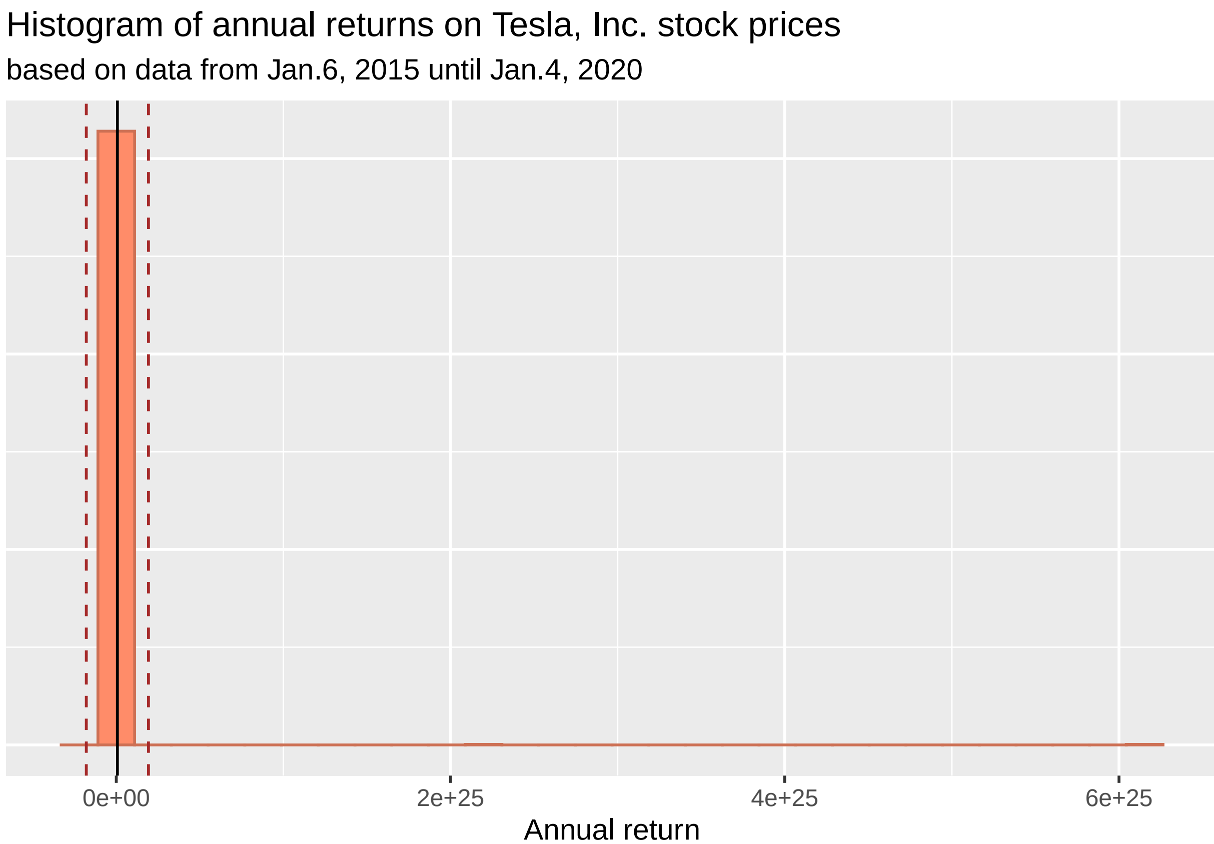
```
## skewness:  30.73026
```

```r
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, altough there is a small probability to get vaues musg 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.

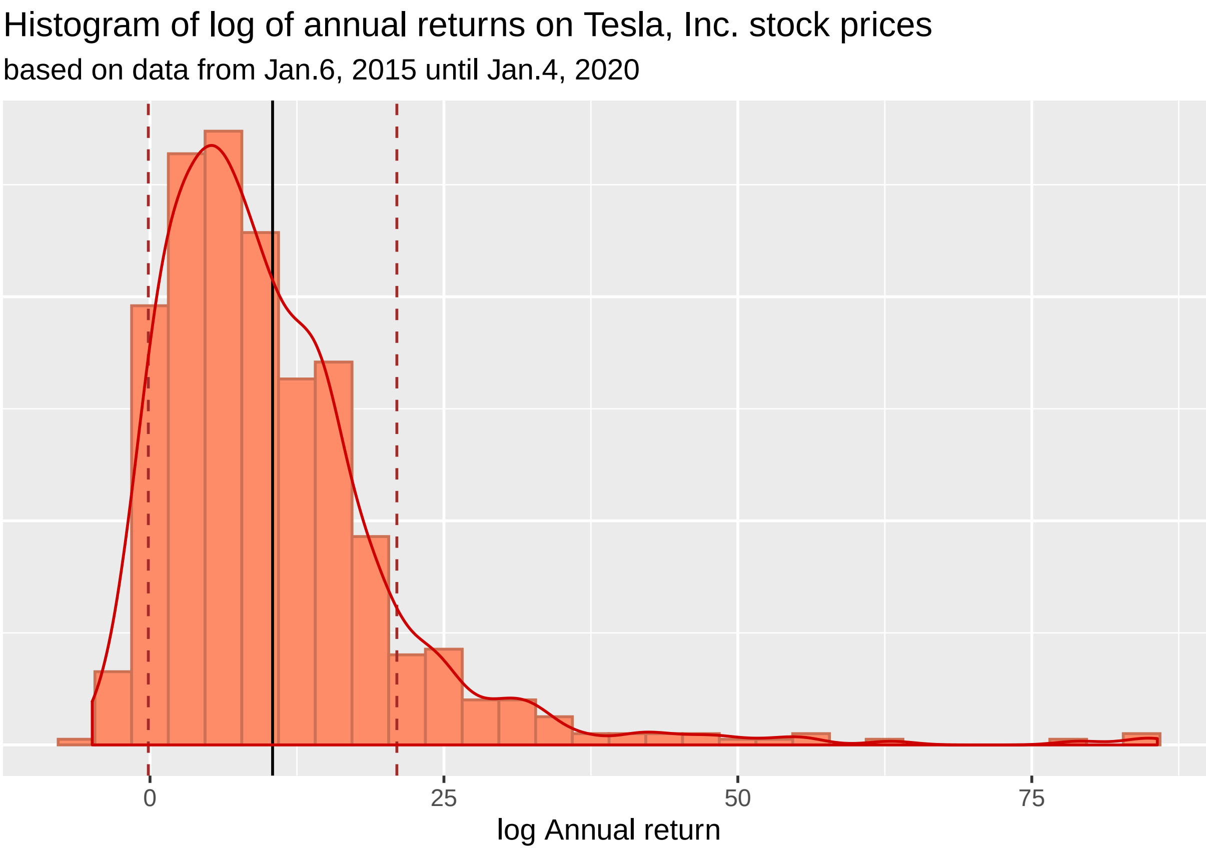
For further analysis we'll visualize all the data we got on annual return.



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)
```

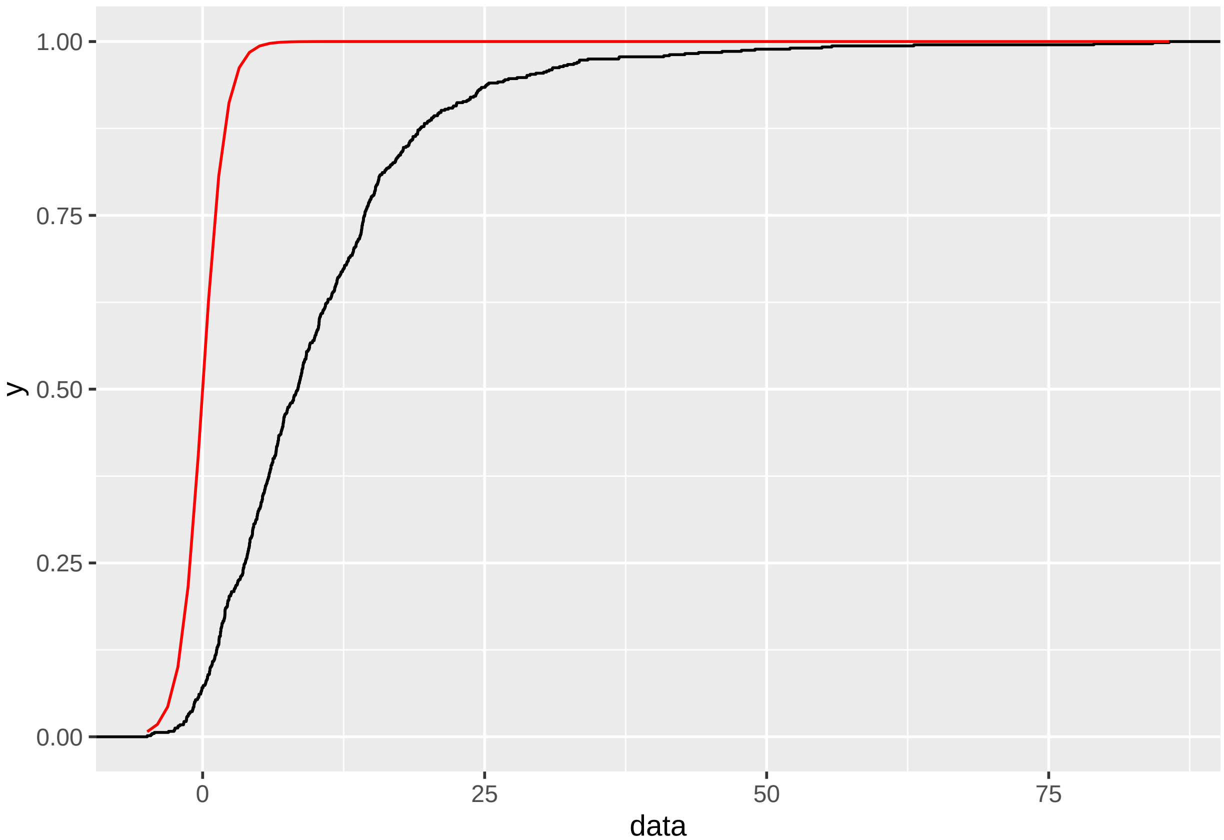
```
## [1]  2.777932  5.248350  3.126778 12.578770 13.357224 13.583587
```



It would be expected to suggest that annual reurn follows a lognormal ditribution 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 altough it has heabier tail - meaning higher kurtosis.

```
ggplot(data.frame(data =log.ret), aes(x = data)) +
  stat_ecdf() +
  stat_function(fun = plogis, color = "red")
```




```r
library(fitdistrplus)
```

```
## Loading required package: MASS
## Loading required package: survival
## Loading required package: npsurv
## Loading required package: lsei
```

```r
library(logspline)
```

```r
# Error here!!!
# fit.logis <- fitdistr(tsla.data$YearRet, "logistic")
```

Conclusions

Hypothesis testing

Getting back to the stock prices themselves, usually people use the historical data on them in order to analyze the pattern and be able to predict them.
A linear regressions seems quite simple for such a difficult pattern we need to start somewhere.

There is no sence in dependence between time and the price but there might be some dependence between prices ob the day and the day-before.
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 linnear regression model For this test we'll again use the closing prices.

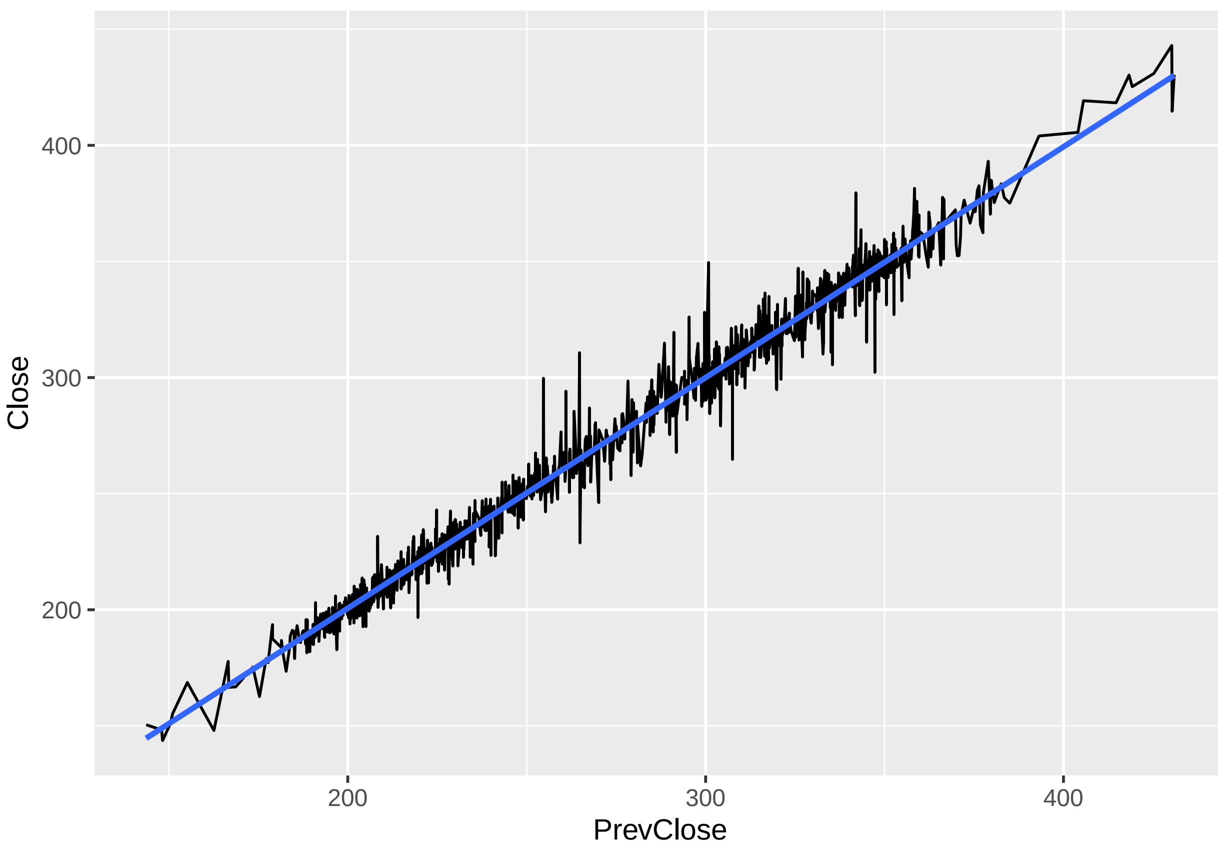
```r
price.lm <- lm(tsla.data$Close~tsla.data$PrevClose)

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
```

As

```
ggplot(tsla.data, aes(x = PrevClose, y = Close)) +
  geom_line() +
  geom_smooth(method='lm', formula= y~x)
```



Conclusions