|  |  |  |
| --- | --- | --- |
| /Users/tinggao/Downloads/bitcoin_logo.pngYN Group Member YUN LI  SHIN GAO  [tg2618@columbia.edu](mailto:tg2618@columbia.edu)  yl3812@columbia.edu | |  | | --- | | future of Bitcoinanalysis of the trend of bitcoin pricethe relationship between oil & gas & gold price |  **5200 PROJECT DELIVERABLE #1** Oct. 17, 2017 |

* Part I

1. **Background**

Bitcoin was invented by an unknown person or group of people under the name Satoshi Nakamoto. It’s a worldwide digital payment system called the first decentralized digital currency. The system is peer-to-peer, and transactions take place between users directly, without an intermediary. These transactions are verified by network nodes and recorded in a public distributed ledger called a blockchain. The ledger is transparent, everyone can search the ledger, but the ID are totally anonymous.

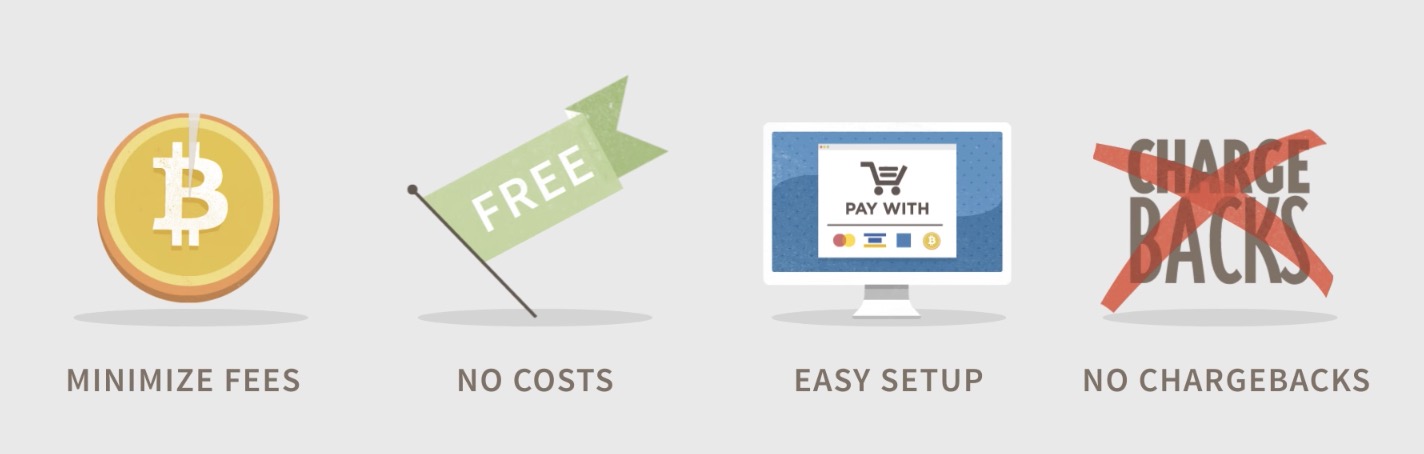
Bitcoins are created as a reward for mining.

Mining is a record-keeping service done through the use of computer processing power. You can consider miners as big servers, they keep the blockchain consistent, complete, and unalterable by repeatedly verifying and collecting newly broadcast transactions.

They can be exchanged for other currencies, products, and services. As of February 2015, over 100,000 merchants and vendors accepted bitcoin as payment.

Advantages:

|  |  |
| --- | --- |
| /Users/tinggao/Library/Containers/com.tencent.xinWeChat/Data/Library/Application Support/com.tencent.xinWeChat/2.0b4.0.9/6a44cdaf070d661399cac6858b3d4ebe/Message/MessageTemp/16e0a6ea1428554d18f7a6969a8591a6/Image/3751508283312_.pic.jpg | Fast peer-to-peer transactions |
| /Users/tinggao/Library/Containers/com.tencent.xinWeChat/Data/Library/Application Support/com.tencent.xinWeChat/2.0b4.0.9/6a44cdaf070d661399cac6858b3d4ebe/Message/MessageTemp/16e0a6ea1428554d18f7a6969a8591a6/Image/3781508283378_.pic.jpg | Worldwide payments |
| /Users/tinggao/Library/Containers/com.tencent.xinWeChat/Data/Library/Application Support/com.tencent.xinWeChat/2.0b4.0.9/6a44cdaf070d661399cac6858b3d4ebe/Message/MessageTemp/16e0a6ea1428554d18f7a6969a8591a6/Image/3791508283397_.pic.jpg | Low processing fees |



Disadvantages:

1. Difficult to exchange.
2. Used for illegal activities (governments cannot track).
3. Mining or solving blocks uses large amount of energy.
4. **Introduction**
5. The source of data and size of the dataset

Our datasets came from two parts, first part is bitcoin data from its official website, second part is date of bulk commodities from other outsource websites.

***Part I:*** CSV files for selecting bitcoin exchanges for the time period from Sep.13 2011 to Oct.8 2017, with day to day updates of OHLC (Open, High, Low, Close), Volume in BTC and indicated currency, and weighted bitcoin price. Timestamps are in Unix time. Timestamps without any trades or activity have their data fields populated with NaNs. If a timestamp is missing, or if there are jumps, this may be because the exchange (or its API) was down, the exchange (or its API) did not exist, or some other unforseen technical error in data reporting or gathering.

***Part II:*** CSV files for selecting Oil price, Gold price as well as Gas price for the time period from Jan.1 2011 to Oct 10, 2017, with day to day updates of OHLC (Open, High, Low, Close), Volume in Oil, Gas and Gold.

See [*Table 1*](#table1):

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Data Category** | **Details** | **Data Source** |
| **Part I** | Bitcoin data | Bitcoin historical price data | https://bitcoincharts.com |
| Bitcoin historical data | https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory/data |
| **Part II** | Bulk Commodities | Oil | https://cn.investing.com/commodities/crude-oil-historical-data |
| Gas | https://cn.investing.com/commodities/natural-gas |
| Gold | https://cn.investing.com/commodities/gold |

*Table 1 Source of Data*

1. Meaning of columns

See [*Table 2*](#table2):

|  |  |  |
| --- | --- | --- |
| **Sheet name** | **Column** | **Description** |
| 2011.9.13  -2017.10.8 | Date | Date of observation |
| Open | Opening price on the given day |
| High | Highest price on the given day |
| Low | Lowest price on the given day |
| Close | Closing price on the given day |
| Volume (BTC) | Volume of transactions on the given day |
| Volume (Currency) | Volume of currency on the given day |
| Weighted Price | Market capitalization in USD |
| bitcoin\_dataset | Date | Date of observation |
| btc\_market\_price | Average USD market price across major bitcoin exchanges. |
| btc\_total\_bitcoins | The total number of bitcoins that have already been mined. |
| btc\_market\_cap | The total USD value of bitcoin supply in circulation. |
| btc\_trade\_volume | The total USD value of trading volume on major bitcoin exchanges. |
| btc\_blocks\_size | The total size of all block headers and transactions. |
| btc\_avg\_block\_size | The average block size in MB. |
| btc\_n\_orphaned\_blocks | The total number of blocks mined but ultimately not attached to the main Bitcoin blockchain. |
| btc\_n\_transactions\_per\_block | The average number of transactions per block. |
| btc\_median\_confirmation\_  time | The median time for a transaction to be accepted into a mined block. |
| btc\_hash\_rate | The estimated number of tera hashes per second the Bitcoin network is performing. |
| btc\_difficulty | A relative measure of how difficult it is to find a new block. |
| btc\_miners\_revenue | Total value of coin base block rewards and transaction fees paid to miners. |
| btc\_transaction\_fees | The total value of all transaction fees paid to miners. |
| btc\_cost\_per\_transaction\_  percent | miners revenue as percentage of the transaction volume. |
| btc\_cost\_per\_transaction | miners revenue divided by the number of transactions. |
| btc\_n\_unique\_addresses | The total number of unique addresses used on the Bitcoin blockchain. |
| btc\_n\_transactions | The number of daily confirmed Bitcoin transactions. |
| btc\_n\_transactions\_total | Total number of transactions. |
| btc\_n\_transactions\_excluding\_popular | The total number of Bitcoin transactions, excluding the 100 most popular addresses. |
| btc\_n\_transactions\_excluding\_chains\_longer\_than\_100 | The total number of Bitcoin transactions per day excluding long transaction chains. |
| btc\_output\_volume | The total value of all transaction outputs per day. |
| btc\_estimated\_transaction\_  volume | The total estimated value of transactions on the Bitcoin blockchain. |
| btc\_estimated\_transaction\_  volume\_usd | The estimated transaction value in USD value. |
| Oil/Gas/Gold | Date | date of observation |
| Open | Opening price on the given day |
| High | Highest price on the given day |
| Low | Lowest price on the given day |
| Close | Closing price on the given day |
| Volume | Volume of transactions on the given day |
| Ratio | (Close Price-last day's close price)/last day's close price |

*Table 2 Meaning of Columns*

* Part II

1. Research Question

Using the data from 2004 to now, We tried to figure out how bitcoin price fluctuate based on USD currency.

After we got basic conception of price fluctuation of bitcoin, we added price data of gas & oil & gold, to analyze if there is any relationship between bitcoin price and bulk commodities. We tried to apply linear regression, log-linear regression, kernel regression to match the dataset, to track which model fits the data best.

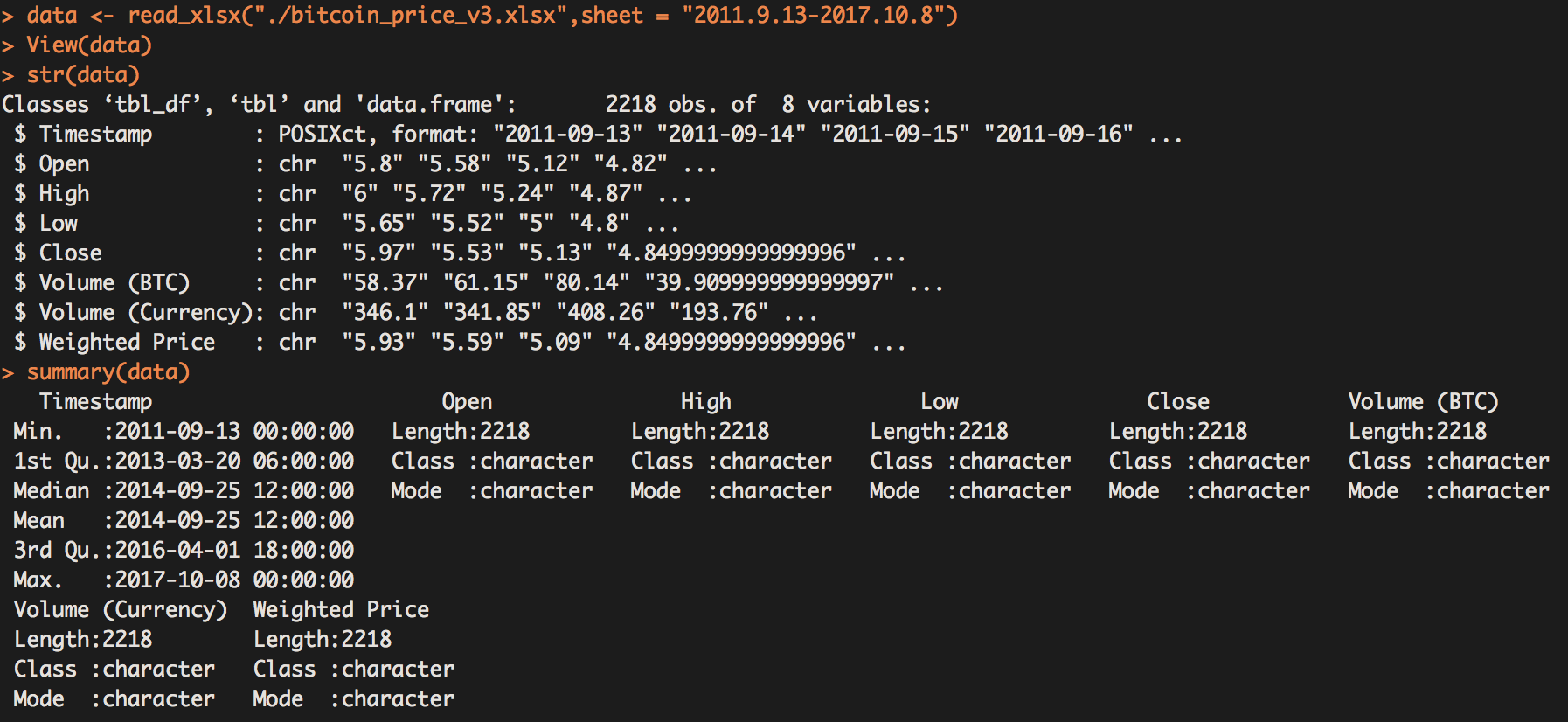
We compared the standard errors, ranges, coefficients from different models to interpret the data from visualization results.

1. Data Cleaning
2. **Load data & Glimpse data structure**

Firstly, we load data into Rstudio to make initial diagnosis of the original dataset.

The date frame contains 8 variables, which are all chr class except ‘Timestamp’ is POSAIXct class. The POSIXct is a date class, we can transform it to standard date class later on. (see [*figure 1*](#figure1))

The other variables are characters, but actually they are all numbers, thus when we got use them, we need to convert them to numeric.

****Since they are all character class, the ‘summary’ function cannot compute their Mode value.

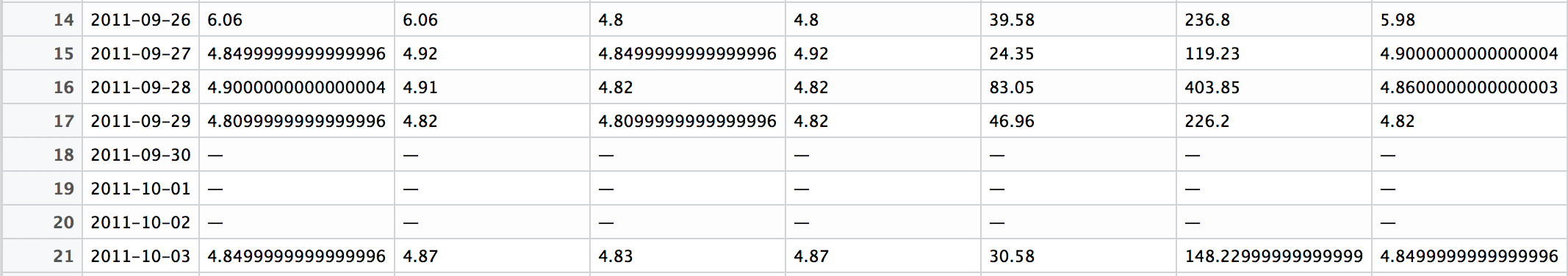
*figure 1: Original data frame*

1. **Delete NA&’-’**

> sum(which(is.na(data)))

[1] 0

us the code upon, we found there is no NA in the dataset, but from easily check the dataset, we found there are missing values represented as ‘-’. (see [*pic. 1*](#pic1))

**

*pic. 1: missing values as ‘-’*

Thus, we used the code below to find out the row indices of rows containing missing values.

> library(stringr)

> str\_count(data[,2:ncol(data)], fixed("—"))

[1] 21 21 21 21 21 21 21

> x<-which(data[ , 2:ncol(data)]== '—', arr.ind=TRUE)

> y<-unique( x[,1] )

> data <- data[-y,]

> y

[1] 18 19 20 33 34 36 37 40 41 45 51 52 56 72 76 83 88 96 1212 1213 1214

Here, we figured out that 21 rows out of 2197 rows (0.95%) containing missing data. ‘y’ is a variable listing the indices of the rows containing ‘-’. Since the proportion is small, we decide to delete those rows. We may have another choice to replace the missing value with the median or mean. But we cannot make sure if the mean or median is representative. Therefore, we delete those rows eventually.

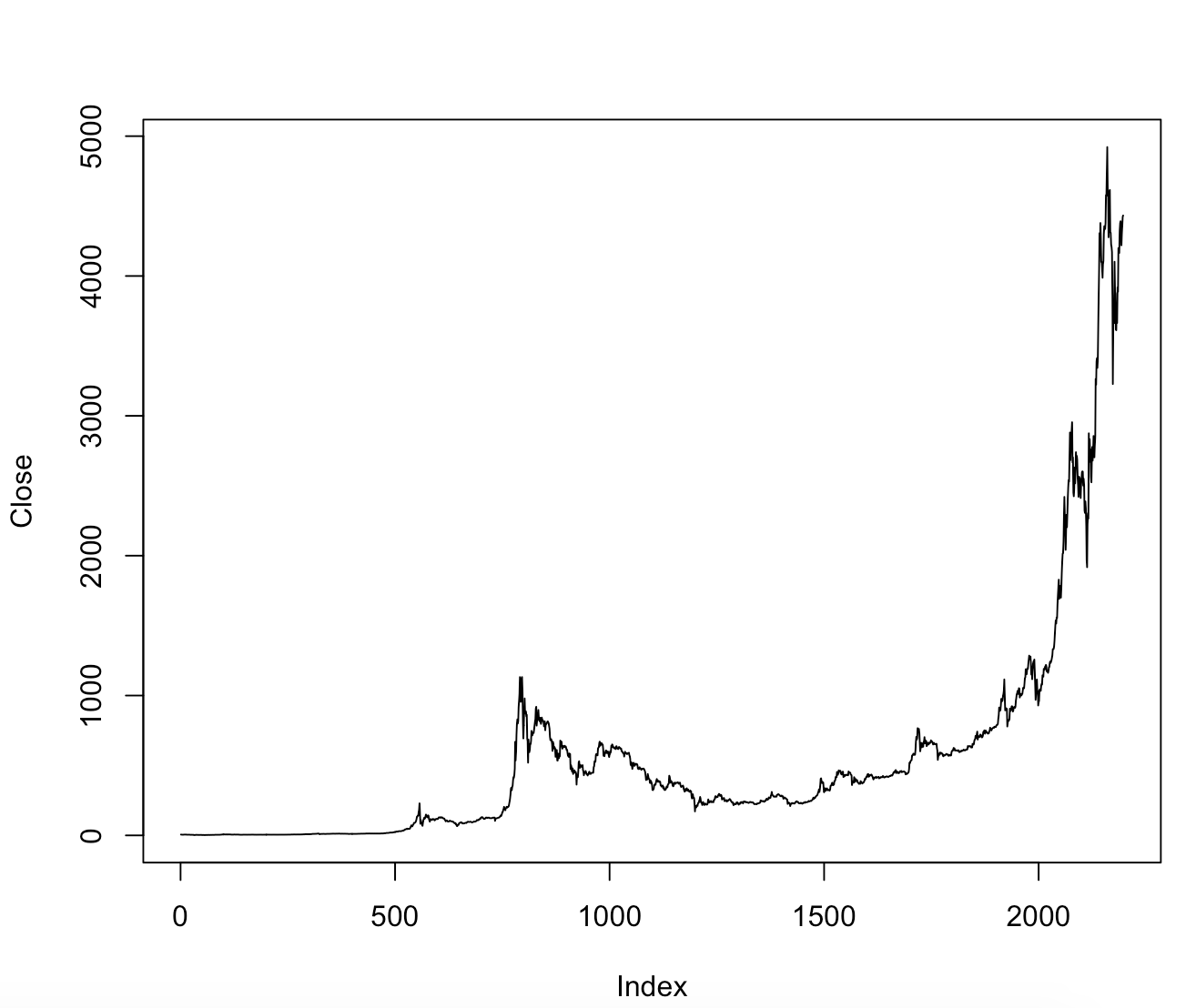
> str\_count(data[,2:8], fixed("—"))

[1] 0 0 0 0 0 0 0

after we delete the rows, we detect the dataset and find all ‘-’ are gone.

1. **Plot price graph**

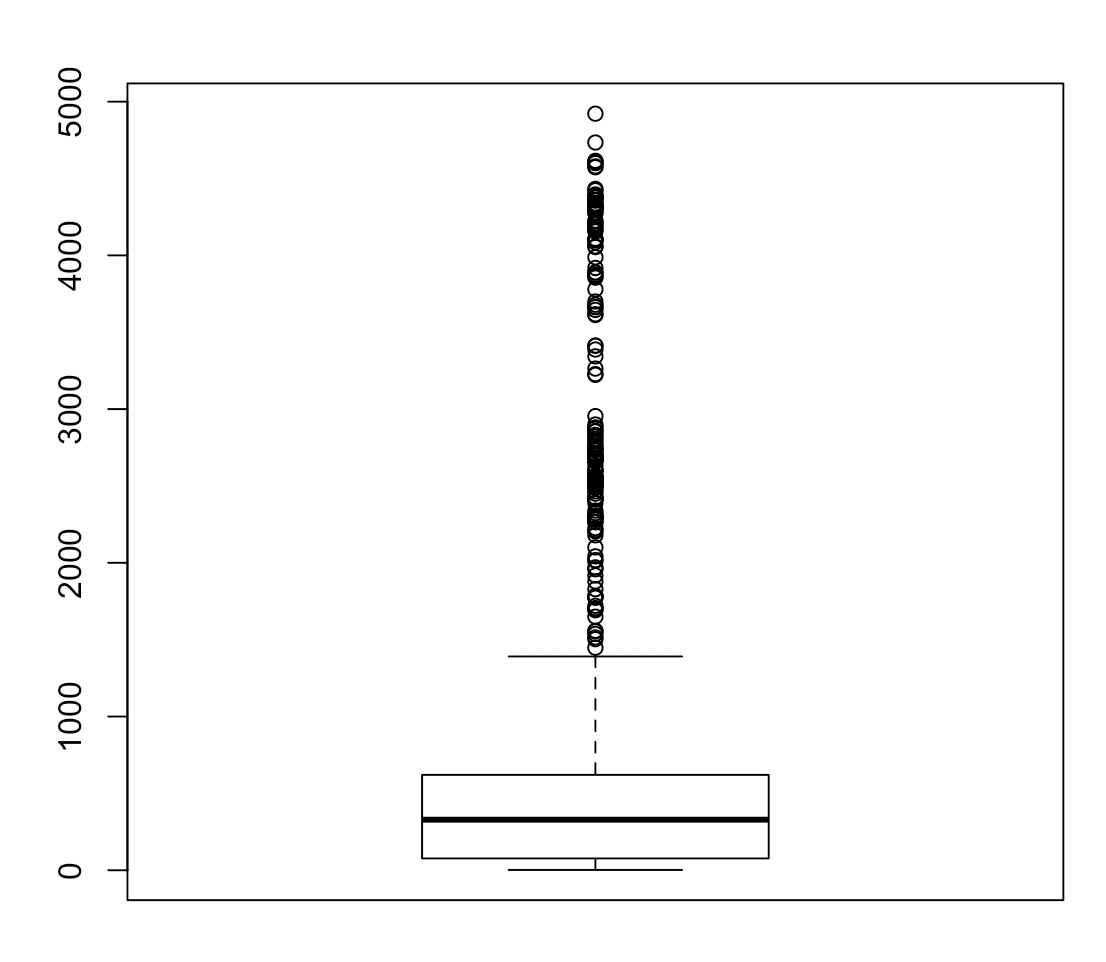
**See figure 2**

****

**figure 2**

1. **Detect outliers**

The outliers in the dataset is hard to filter, since the price keeps increasing exponentially, and the price rise very high in the last year. The boxplot print there are many outliers in the dataset. (see *figure 3*)



*figure 2 Close price centralization*

In fact, the better way to detect the outliers is to test a new variable ‘DailyChangeRate’, which mean the daily change rate of the price.

1. **Split ‘Timestample’ into Year, Month, Day**

The original Timestample data is like this :’ 2014-03-29’, it’s a standard date format. But here, we want to analyze the statistical information of each year. Thus, we want to split the Timestample into Year, Month, and Day.

> datesplit <- unlist(strsplit(as.character(data$Timestamp), "-", fixed = TRUE))

> datesplit <- split(datesplit, 1:3)

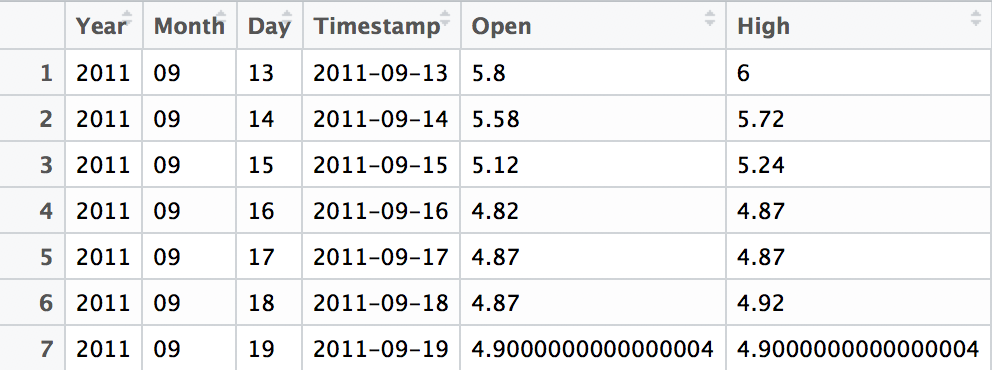
> Month <- datesplit$`2`

> Day <- datesplit$`3`

> Year <- datesplit$`1`

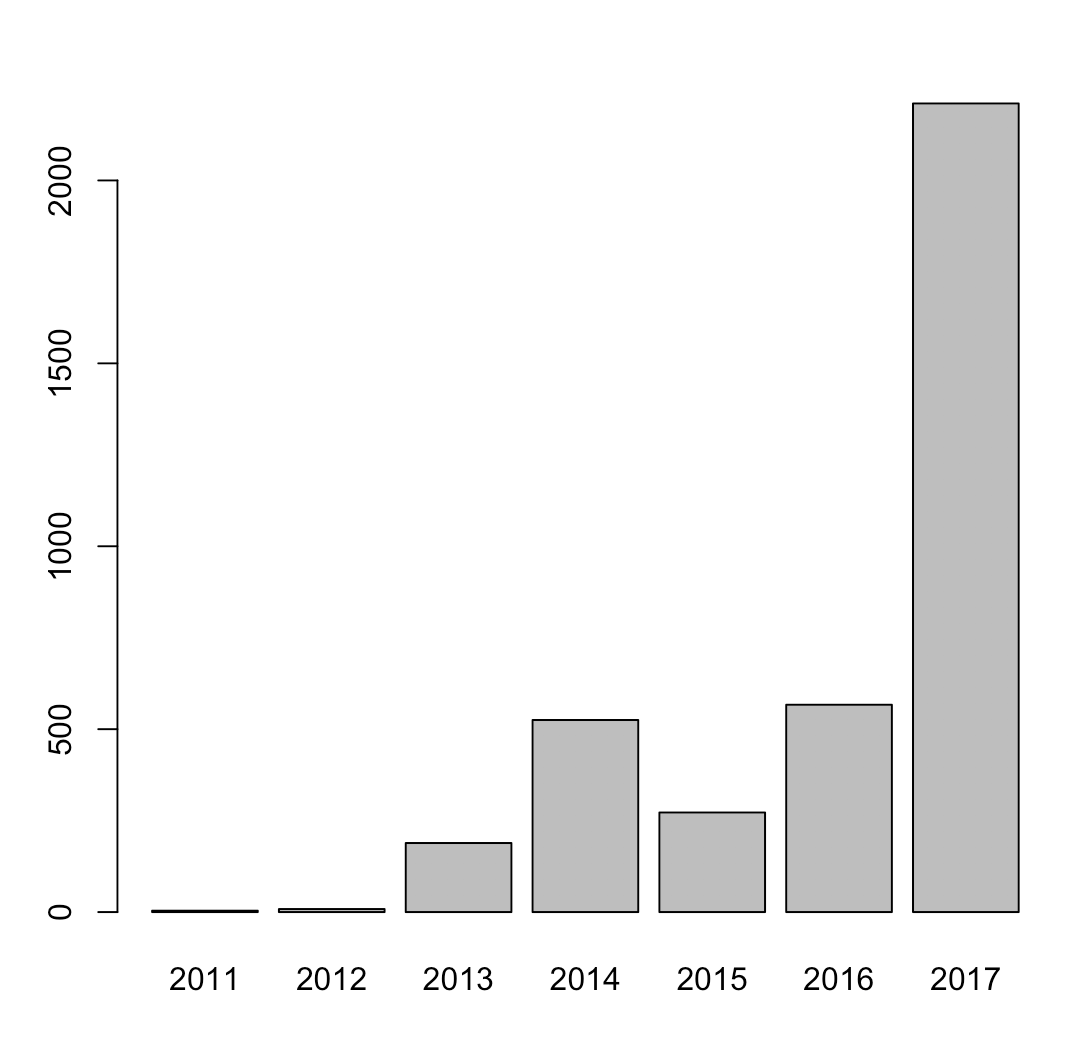
> data <- cbind(Year,Month,Day,data)

Now we split the original Timestample, and added three new variable to the dataset. (see [*pic. 2*](#pic2))



*pic. 2 New date variables added into the dataset*

Then we barplot the mean Close price of each year to see the price trend in the last few years*.* (see[*figure 3*](#figure3))

**

*figure 3 mean price of each year*

The graph shows that the price is keep rising, but in Year 2015, the mean price of decreased a little, but in 2016 the price regained to a similar level to the mean price in 2014. And in 2017, the price blown up, almost 3 times higher than the previous year.

1. **Create new variable ----- ‘DailyChangeRate’**

We used the following code to creat a new variable as ‘DailyChangeRate’’ based on the Close price and Open price of each day, to mark the change rate of each day. The DailyChangeRate can be positive or negative, it helps to define whether the price changed too sharply in one day.

> attach(data)

> Close <- as.numeric(Close)

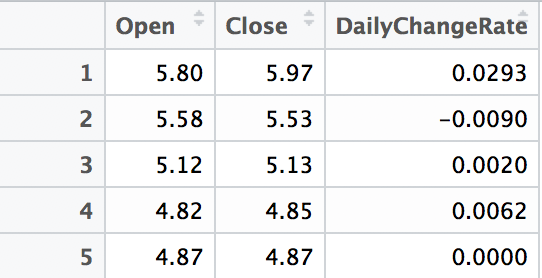
> Open <- as.numeric(Open)

> DailyChangeRate <- (Close - Open)/Open

> DailyChangeRate <- round(DailyChangeRate,digits = 4)

> data\_update <- cbind(data,DailyChangeRate)

We also round the rate to 4 digit for easy visualization. The result is shown below in [*pic. 3*](#pic3)*:*



*pic.3 Daily Change Rate*

Also, we want to testify if there is any outlier in term of the new variable. We use the code below to plot boxplot graph to visualize the distribution of the new variable. (see [*figure 4*](#figure4))

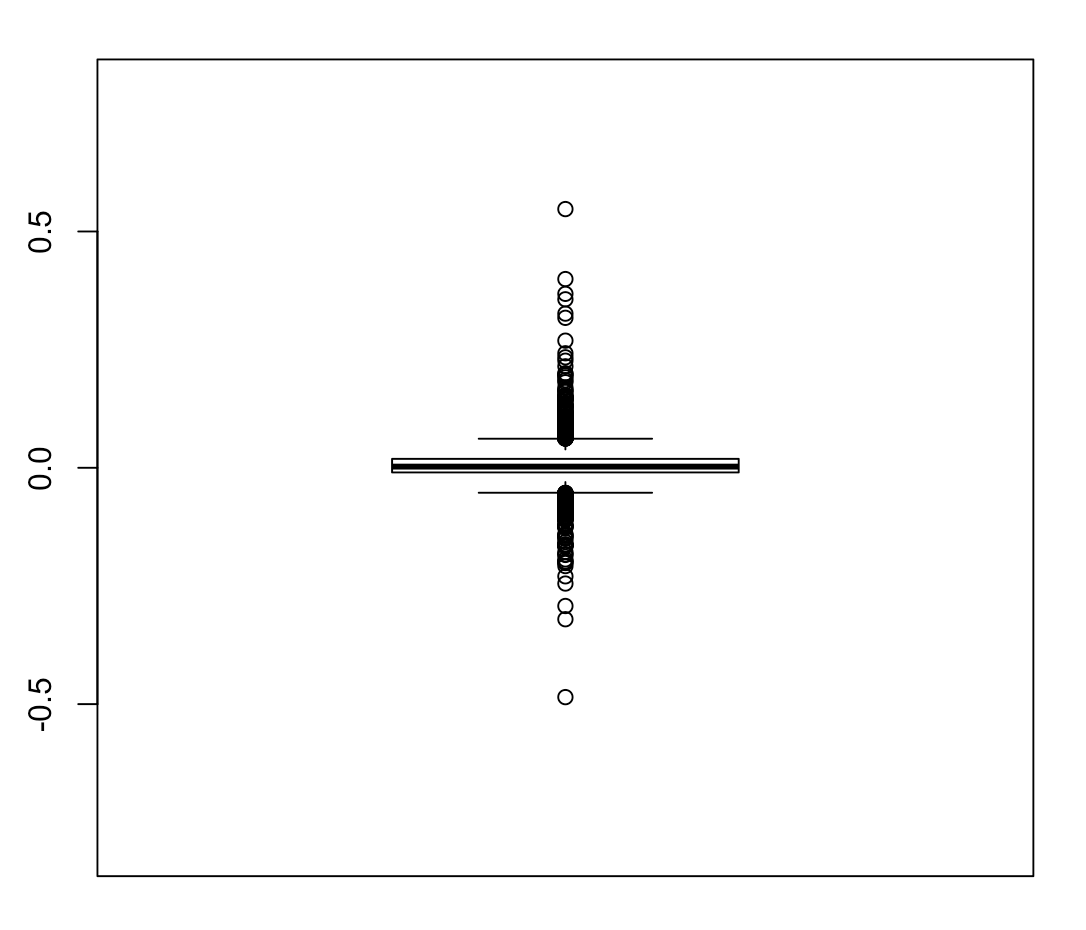
> boxplot(DailyChangeRate,ylim = c(-0.8,0.8))

The result seems to contain many outliers, but all data points are within the ylim range, which is to say, all change rate are between -0.8 to 0.8. To further prove it, we ran the code below to see the consequence.

> sum(which(DailyChangeRate < -0.8 | DailyChangeRate > 0.8))

[1] 0

The result is 0, which testify our conclusion, the absolute value of all change rates are below 0.8. More specifically, we can find that all rates are between -0.5 to 0.6. So here, we made a safely produced conclusion based on the result: the changes of price are all in control, there are no visible outliers in this dataset.



*figure 4 outliers of DailyChangeRate*

To learn more about the distribution of the change rates, we used the function cut to count the number of rates in each break between -0.5 – 0.6.

> breaks <- seq(-0.5,0.6, by = 0.1)

> dat2 <- cut(data$DailyChangeRate, breaks = breaks)

> table(dat2)

dat2

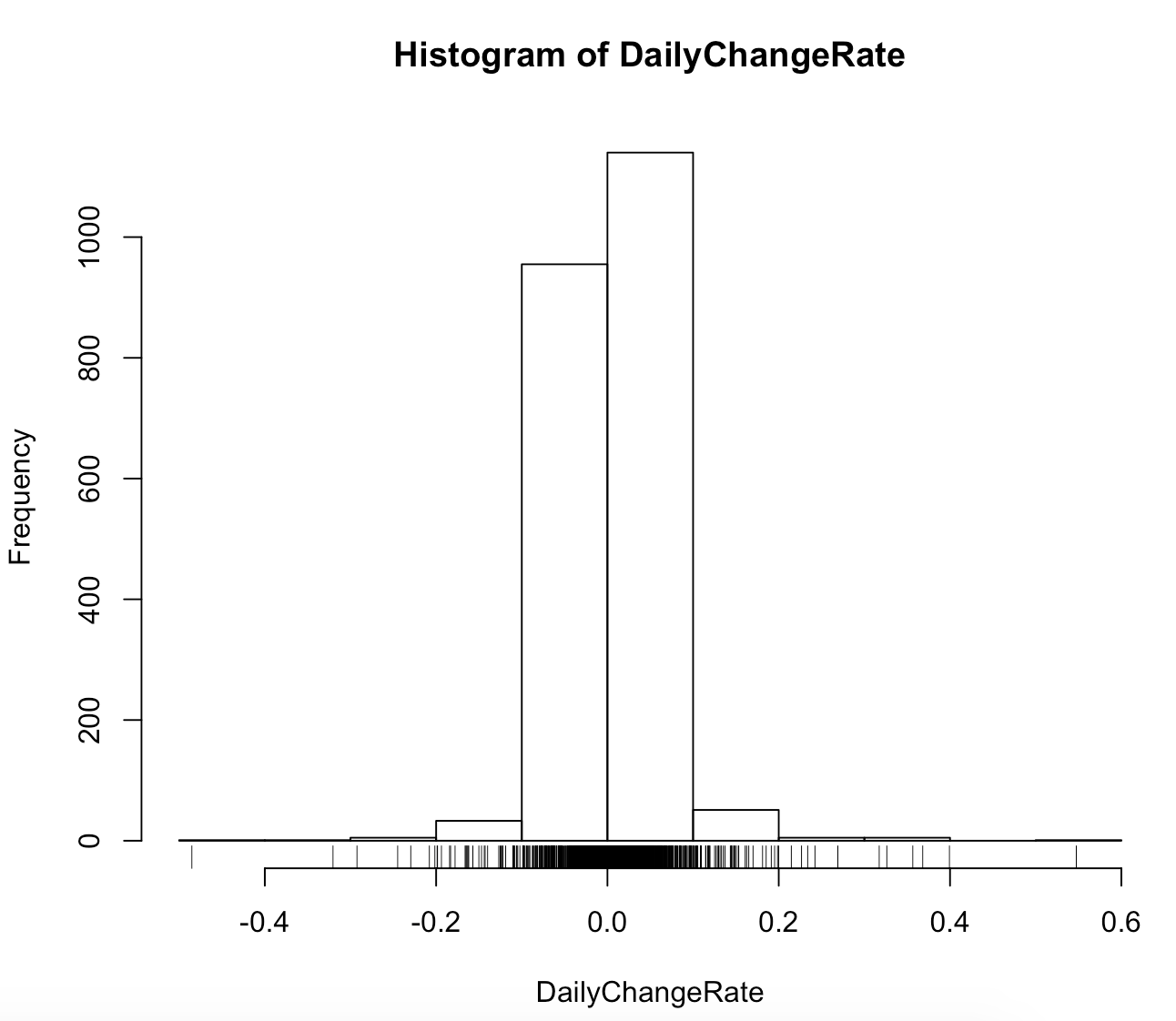
(-0.5,-0.4) (-0.4,-0.3) (-0.3,-0.2) (-0.2,-0.1) (-0.1,0) (0,0.1)

1 1 5 33 955 1140

(0.1,0.2) (0.2,0.3) (0.3,0.4) (0.4,0.5) (0.5,0.6)

51 5 5 0 1

In order to observe the distribution more directly and clearly, we plot a histogram graph. ([See *figure 5*](#figure5))



*figure 5 distribution of change rates*

It’s clear here to find out that most of change rates are centralized within -0.1 to 0.1, which totally makes sense. But still some data go beyond ±0.1, almost all of the data are within -0.2 to 0.2, those outsider samples are small enough to ignore.

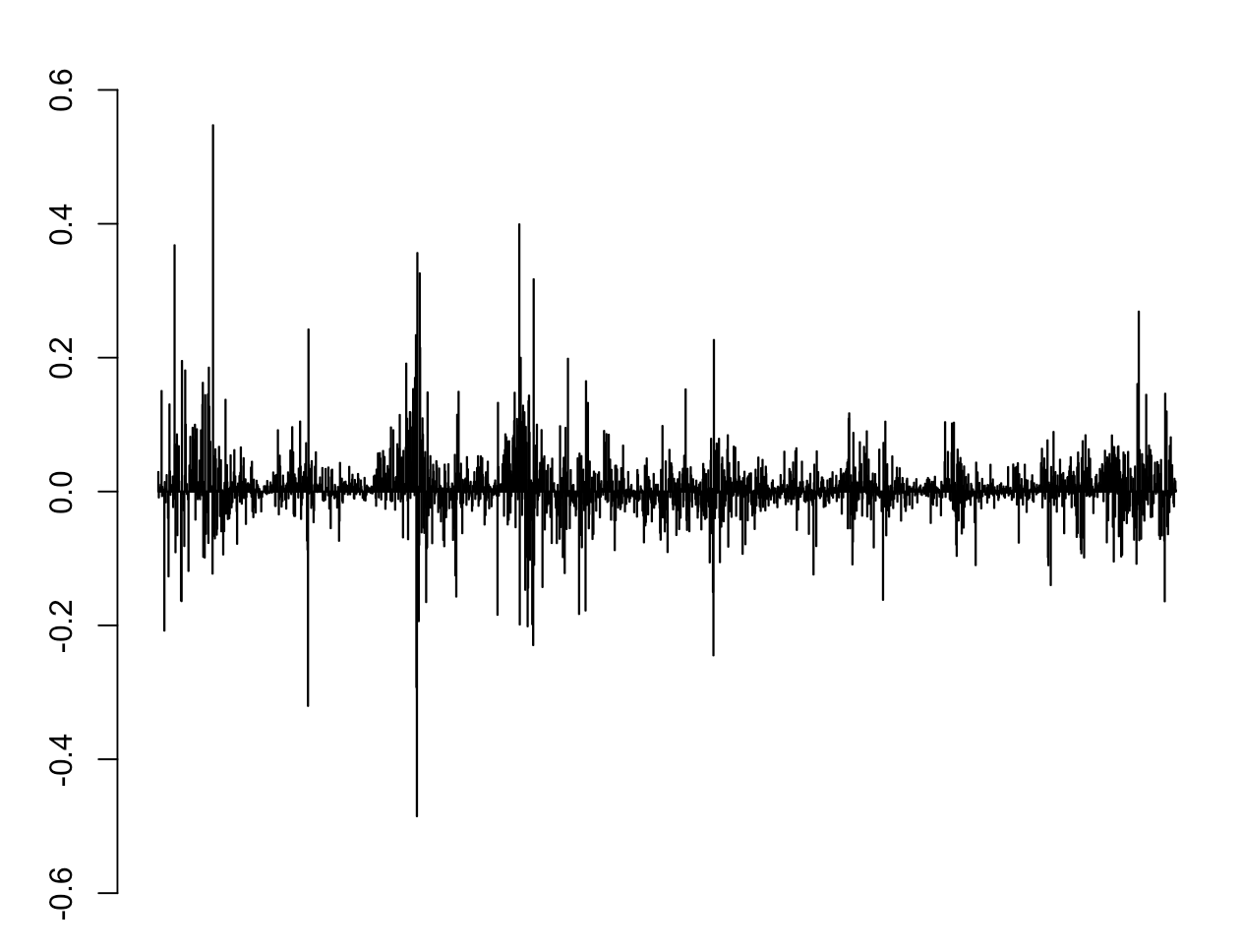
We also find that the number of positive rates is obviously greater than the negative one, which makes it reasonable that the price keeps rising.

Another way to visualize the daily change rate is to plot the rate of each day in one barplot. See [*figure 6*](#figure6)*.*

1. Do linear regression for Price

As we have seen in figure 2, it’s obvious there is no linear relationship between year and Close price. The price increase exponentially, thus we decided to test if the relation between price and the number of days counted from a start date fits the exponential function.

First, we set a start date, since the earliest date in the dataset is “9-13-2011”, so we can set a date nearer to it but before it. Here we set the start count date as “01-01-2011”.



*figure 6 Daily Change Rate*

> library("lubridate")

> start\_date <- as.POSIXct("01-01-2011", format = "%d-%m-%Y", tz = "UTC")

> span <- Timestamp - start\_date

> span <- day(days(span))

> data <- cbind(data,span)

We got a new variable, ‘span’, which represents the number of days between the date of each piece of data form the start date.

Then we did Log-Linear Regression about the ‘span’ and ‘Close’ to see if there is any correlation between time and price. See figure 8

It seems much more linear-related compared with figure 2. To further assess the relation, we built a log-linear model to test its standard error, R-squared and other statistical information.

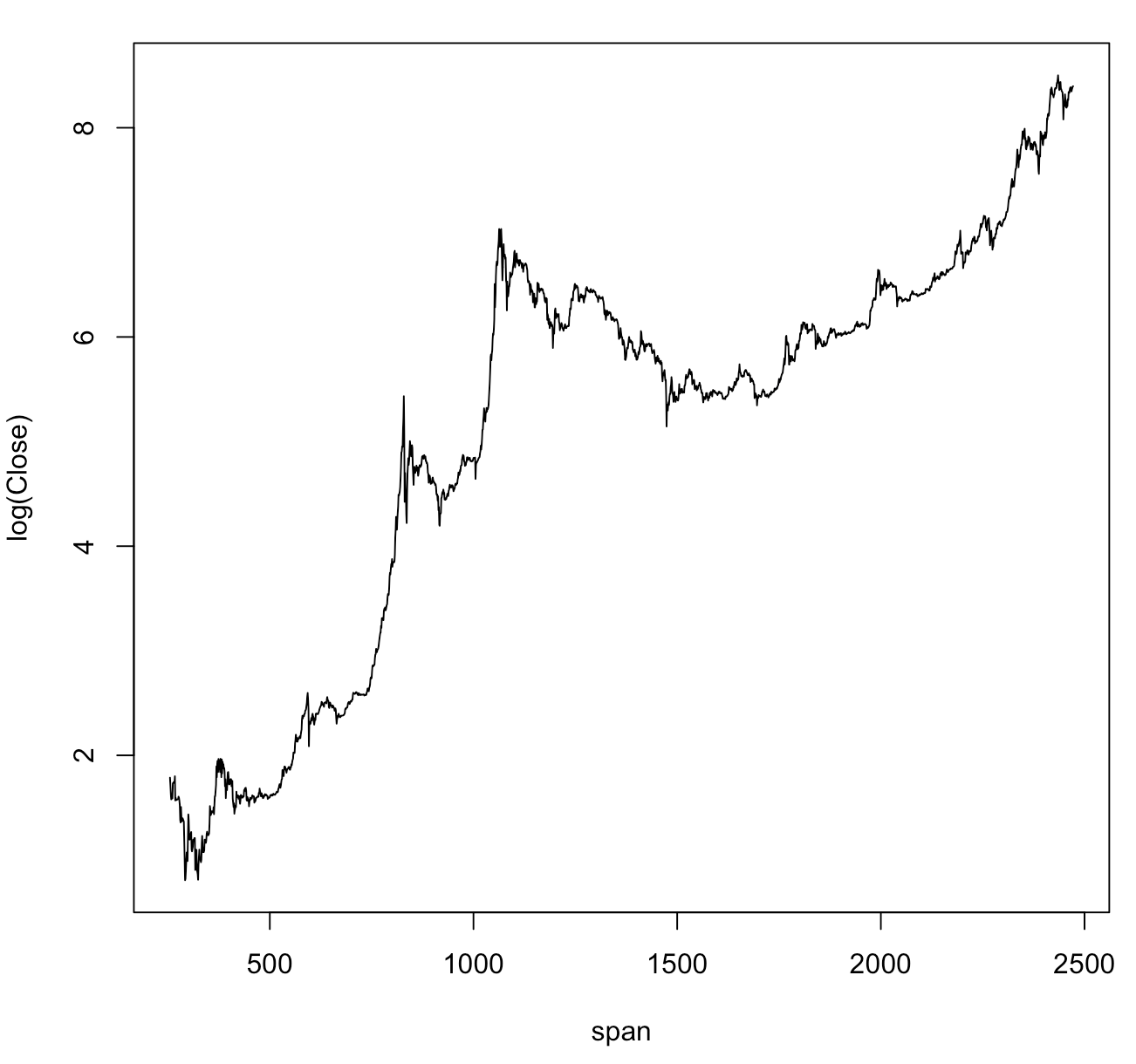


figure 8

> daymodel <- lm(Close ~ log(span))

> coef(daymodel)

(Intercept) log(span)

-5018.7354 784.9076

> plot(daymodel)

> summary(daymodel)

Call:

lm(formula = Close ~ log(span))

Residuals:

Min 1Q Median 3Q Max

-608.1 -381.2 -161.9 113.5 3820.0

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -5018.74 178.68 -28.09 <2e-16 \*\*\*

log(span) 784.91 25.14 31.22 <2e-16 \*\*\*

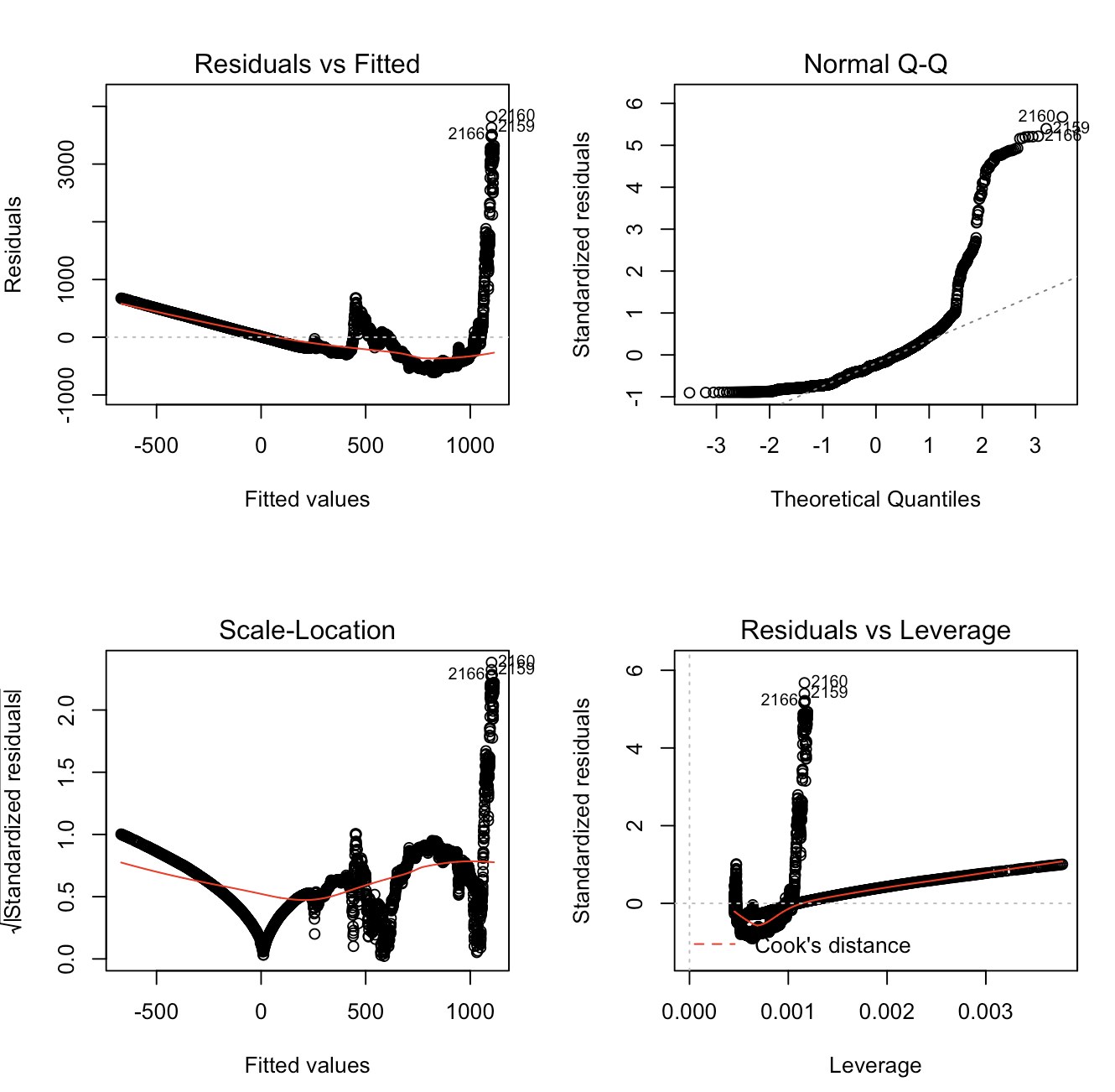
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 673.4 on 2195 degrees of freedom

Multiple R-squared: 0.3075, Adjusted R-squared: 0.3072

F-statistic: 974.8 on 1 and 2195 DF, p-value: < 2.2e-16



*figure 9*

Here, we find that the standard error of log(span) is still very large,