Future of Bitcoin

analysis of the trend of bitcoin price
relationship between oil & gas & gold price

GROUP MEMBER

YUN LI
SHIN GAO

5200 PROJECT DELIVERABLE #1

OCT. 17, 2017

tg2618@columbia.edu
yl3812@columbia.edu

TABLE OF CONTENT

❖ Part I

1. Background

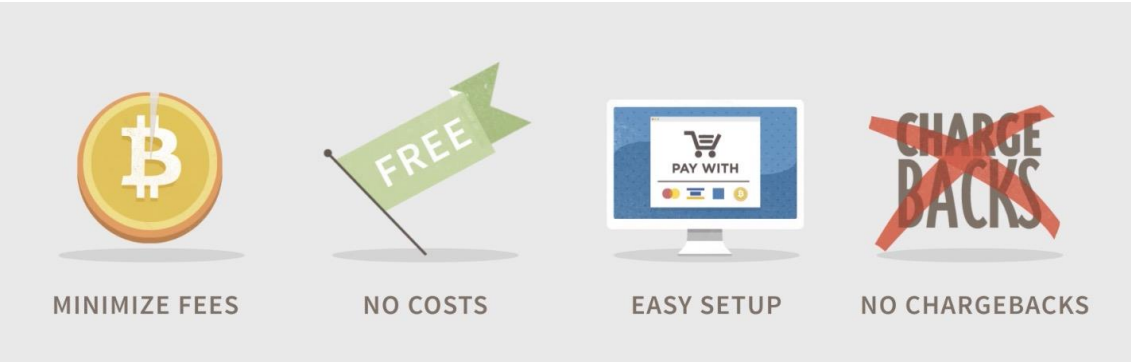
Bitcoin was invented by an unknown person or group of people under the name Satoshi Nakamoto. It's a worldwide digital payment system called the first decentralized digital currency. The system is peer-to-peer, and transactions take place between users directly, without an intermediary. These transactions 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:

- Fast peer-to-peer transactions
- Worldwide payments
- 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.

2. Introduction

A. 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*:

Type	Data Category	Details	Data Source
Part I	Bitcoin data	Bitcoin historical price data	https://bitcoincharts.com
		Bitcoin historical data(outside sources)	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

B. Meaning of columns

See *Table 2*:

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 coinbase 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

C. Evaluation of data quality

To identify data quality issues to business impacts, we need to both classify our data quality expectations as well as our business impact criteria. We analyzed key data issues according to six common data quality dimensions: completeness, conformity, consistency, accuracy, duplication as well as integrity.

1. **Completeness**
   Our goal is to figure out how bitcoin price fluctuated as time over and the relationships between bitcoin price and bulk commodities. Therefore, the following information should be included in our datasets:

   Bitcoin: date, open price, close price, high price, low price, transaction volume and currency volume;
   Bulk commodities: date, open price, close price, high price, low price.

   After reviewing our datasets, we found that all the information is included but there were some missing values on specific date. This may be because the exchange (or its API) was down, the exchange (or its API) did not exist, or some other unknown technical error in data reporting or gathering.

   Our process of dealing with missing values could be found in Part II – Data cleaning - Delete NA&'-'.

2. **Conformity**
   In order to do further analysis, our expectation is that data values conform to specified formats. Maintaining conformance to specific formats is important in data representation, presentation, aggregate reporting, search, and establishing key relationships.

   The problem in our dataset is all the variables are shown as "chr" which should be transformed into date class and numeric class depending on their specific meanings.

   Our process of dealing with missing values could be found in Part II – Data cleaning - Split 'Timestample' into Year, Month, Day

3. **Consistency**
   Bitcoin data and bulk commodities data come from different systems and applications, therefore, they have the same column names with different meanings.

We renamed columns of bulk commodities to distinguish specific column meanings. The process could be found in Part III – renamed column names of new datasets.

4. **Accuracy**
   Our data source is from official websites of Bitcoin and bulk commodities. Therefore, we assumed the accuracy of data but used boxplot to identify any outliers in order to confirm whether there were manual mistakes in our datasets.

   The process could be found in Part II – detect outliers.

5. **Duplication**
   Bitcoin data and bulk commodities data are sorted by date. We need to make sure there are no multiple, unnecessary representations of the same data objects within our datasets.

   The process could be found in Part II – detect duplication values.

6. **Integrity**
   We counted numbers of row and column and compared with original csv files to make sure that the data we loaded in RStudio is not missing.

D. Issues that are not worth fixing

1. Outliers

1.1 Outliers in Close Price

Based on our data quality analysis, we used boxplot to identify any outliers in our datasets. The boxplot printed there were many outliers in column Close price in the dataset. (see figure 3)

According to our analysis of bitcoin price and the background information we learned before, we found that the price keeps increasing exponentially, and the price reached very high in the last year. Therefore, this data issue could be ignored.

1.2 Outliers in DailyChangeRate

Also, the boxplot printed there were many outliers in the new variable "DailyChangeRate" we created. (see figure 4)

However, after further analysis of the distribution of outliers, all change rate lay out between -0.5 to 0.6. Thus, the outliers in this variable would have no significant impact on dataset and there is no need to fix it.

Please find detailed processing of dealing with outliers in Part II – Data cleaning – Detect outliers.

2. Delete Missing Values

According to our data quality analysis, there are some missing values on specific date. In the data cleaning process, we figured out that 21 rows out of 2197 rows (0.95%) containing missing data. Since the proportion is so small, we decided to delete these rows instead of analyzing whether it is better to replace these missing values by the median or mean value.

❖ Part II

1 Research Question

Using the data from 2011 to now, we tried to figure out how bitcoin price fluctuated 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.

2 Data Cleaning

1. 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*)
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.

```
> data <- read_xlsx("./bitcoin_price_v3.xlsx",sheet = "2011.9.13-2017.10.8")
> View(data)
> str(data)
Classes 'tbl_df', 'tbl' and 'data.frame':       2218 obs. of  8 variables:
 $ Timestamp        : POSIXct, format: "2011-09-13" "2011-09-14" "2011-09-15" "2011-09-16" ...
 $ Open             : chr  "5.8" "5.58" "5.12" "4.82" ...
 $ High             : chr  "6" "5.72" "5.24" "4.87" ...
 $ Low              : chr  "5.65" "5.52" "5" "4.8" ...
 $ Close            : chr  "5.97" "5.53" "5.13" "4.8499999999999996" ...
 $ Volume (BTC)     : chr  "58.37" "61.15" "80.14" "39.909999999999997" ...
 $ Volume (Currency): chr  "346.1" "341.85" "408.26" "193.76" ...
 $ Weighted Price   : chr  "5.93" "5.59" "5.09" "4.8499999999999996" ...
> summary(data)
   Timestamp                      Open              High              Low               Close           Volume (BTC)
 Min.   :2011-09-13 00:00:00   Length:2218        Length:2218        Length:2218        Length:2218        Length:2218
 1st Qu.:2013-03-20 06:00:00   Class :character   Class :character   Class :character   Class :character   Class :character
 Median :2014-09-25 12:00:00   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :2014-09-25 12:00:00
 3rd Qu.:2016-04-01 18:00:00
 Max.   :2017-10-08 00:00:00
 Volume (Currency)  Weighted Price
 Length:2218        Length:2218
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Figure 1: Original data frame

2. Delete NA & '-'

```
> sum(which(is.na(data)))
[1] 0
```

use 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*)

14	2011-09-26	6.06	6.06	4.8	4.8	39.58	236.8	5.98
15	2011-09-27	4.8499999999999996	4.92	4.8499999999999996	4.92	24.35	119.23	4.9000000000000004
16	2011-09-28	4.9000000000000004	4.91	4.82	4.82	83.05	403.85	4.8600000000000003
17	2011-09-29	4.8099999999999996	4.82	4.8099999999999996	4.82	46.96	226.2	4.82
18	2011-09-30	–	–	–	–	–	–	–
19	2011-10-01	–	–	–	–	–	–	–
20	2011-10-02	–	–	–	–	–	–	–
21	2011-10-03	4.8499999999999996	4.87	4.83	4.87	30.58	148.22999999999999	4.8499999999999996

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.

3. Plot price graph

Here, we ploted the price fluctuation graph to directly see how price of bitcoin changed over time. See *figure 2*

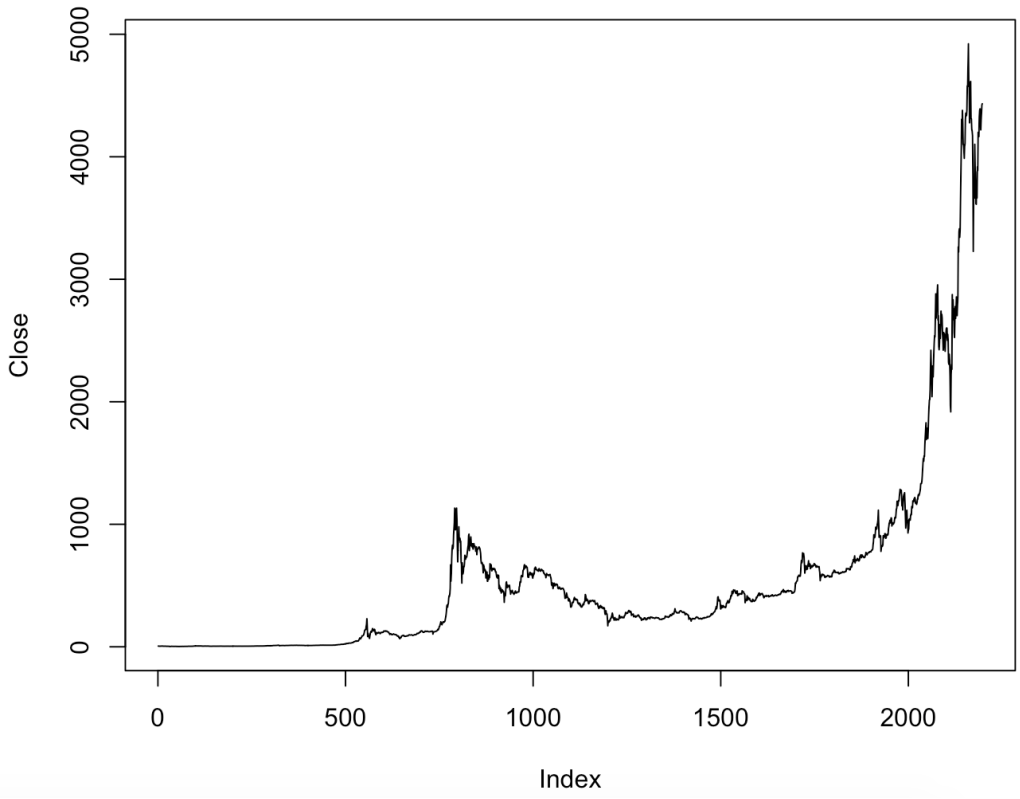


figure 2 Price of Bitcoin

4. Detect outliers

The next step is to find outliers, delete them or replace them. The toolbox we learned in class to detect outliers is to boxplot the data, and find the index of the rows which contain 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*)

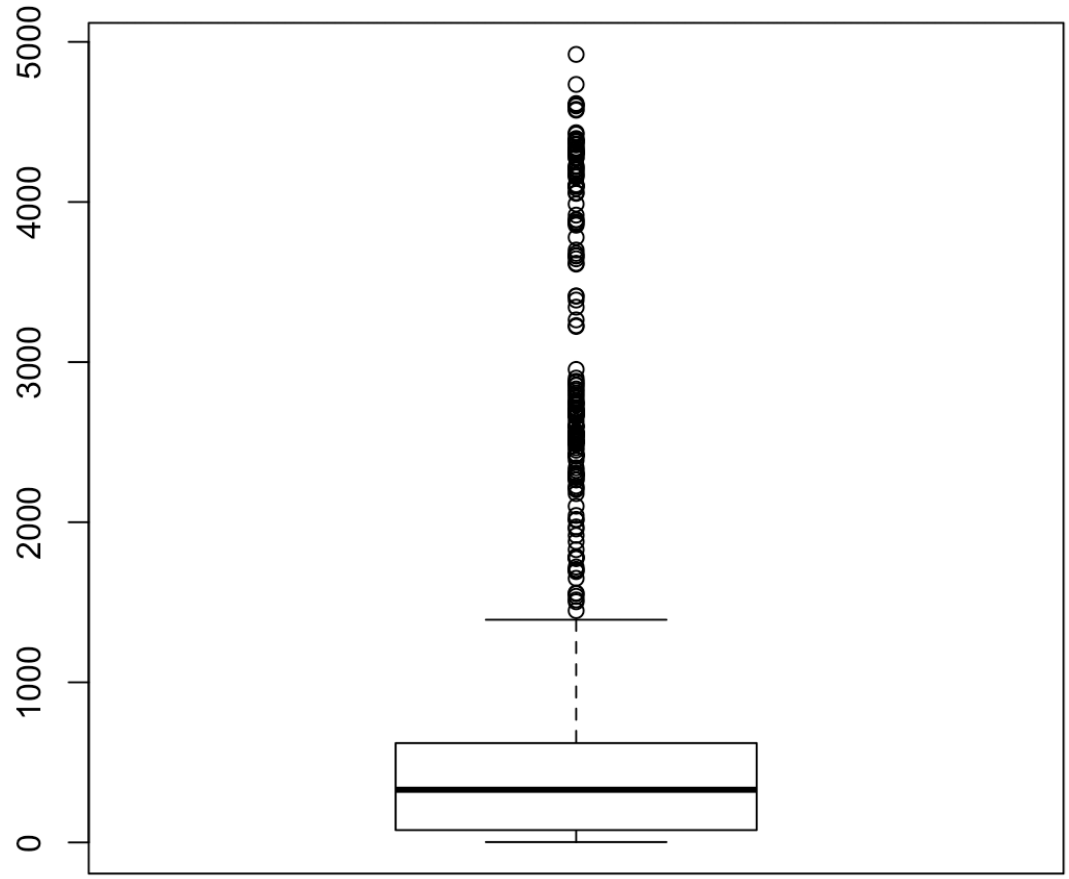


figure 3 Close price centralization

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.

5. 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*)

	Year	Month	Day	Timestamp	Open	High
1	2011	09	13	2011-09-13	5.8	6
2	2011	09	14	2011-09-14	5.58	5.72
3	2011	09	15	2011-09-15	5.12	5.24
4	2011	09	16	2011-09-16	4.82	4.87
5	2011	09	17	2011-09-17	4.87	4.87
6	2011	09	18	2011-09-18	4.87	4.92
7	2011	09	19	2011-09-19	4.9000000000000004	4.9000000000000004

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 4*)

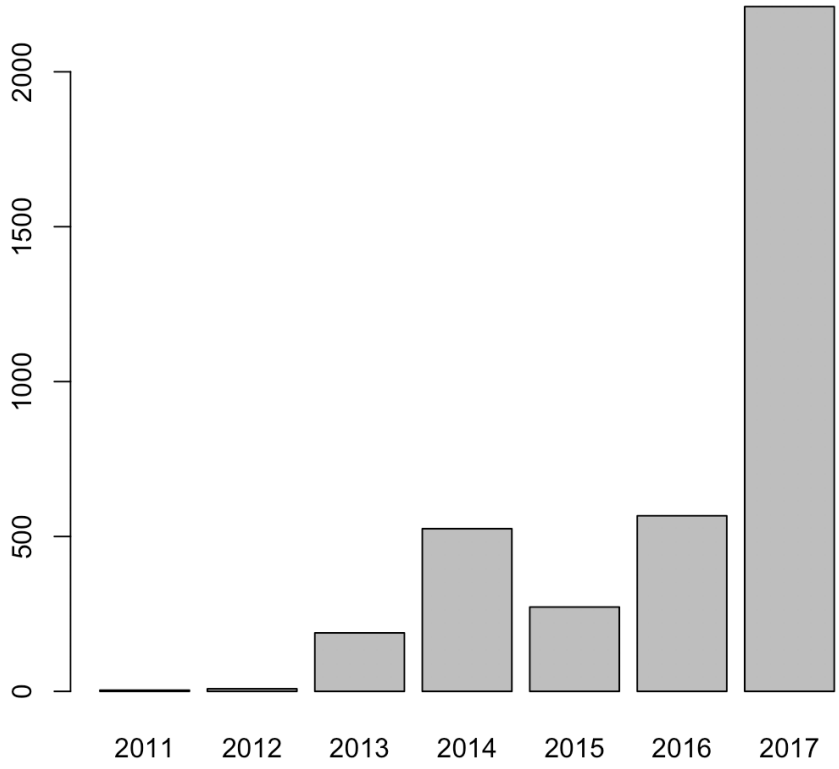


figure 4 mean price of each year

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.

6. Create new variable ----- 'DailyChangeRate'

We used the following code to create 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*:

	Open	Close	DailyChangeRate
1	5.80	5.97	0.0293
2	5.58	5.53	-0.0090
3	5.12	5.13	0.0020
4	4.82	4.85	0.0062
5	4.87	4.87	0.0000

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 5*)

```
> 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 is 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.

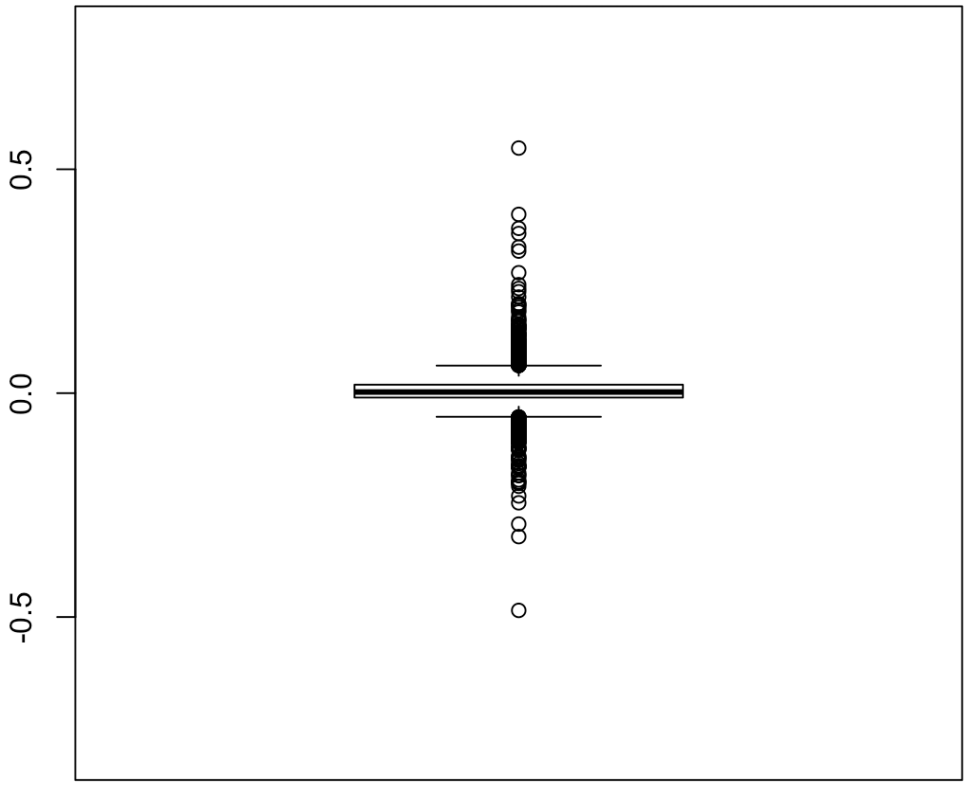


figure 5 outliers of DailyChangeRate

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 6*)

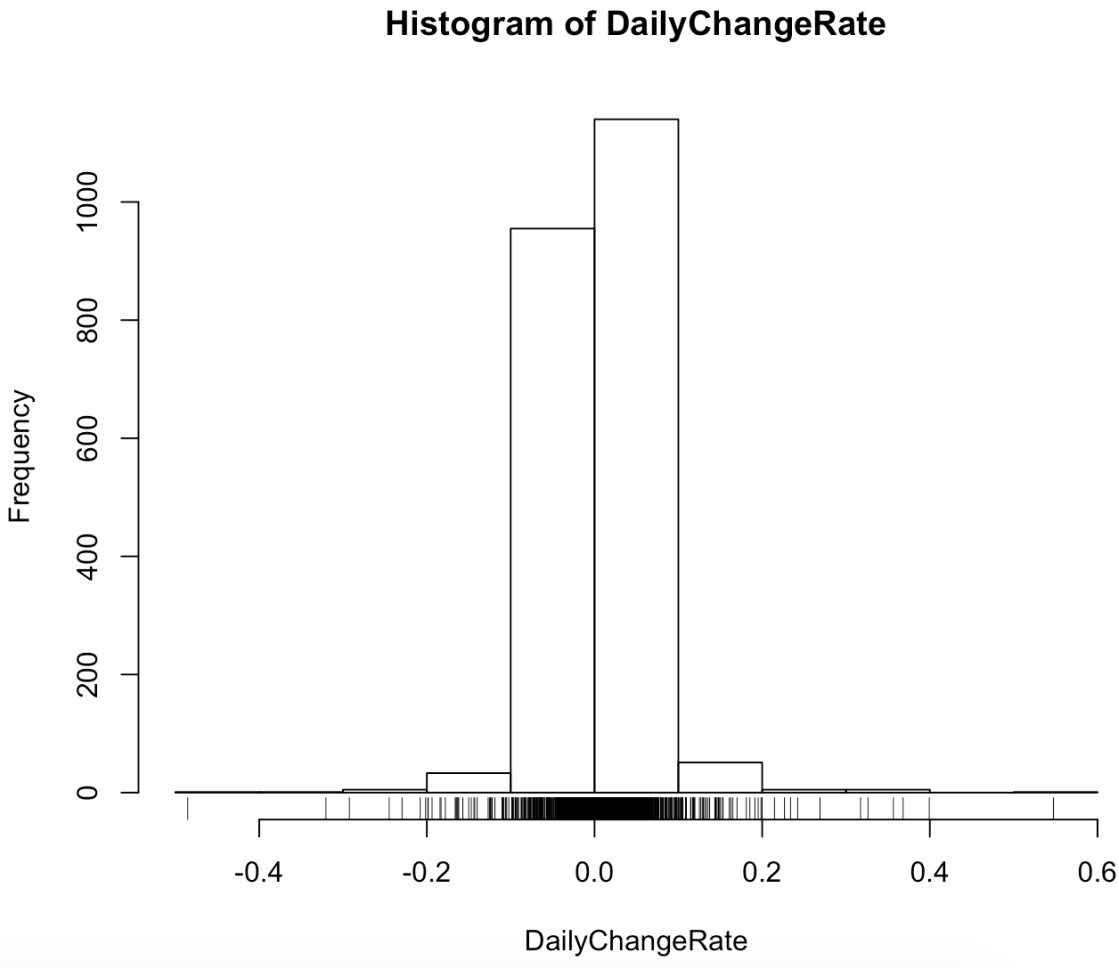


figure 6 distribution of change rates

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 7*.

7. 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”.

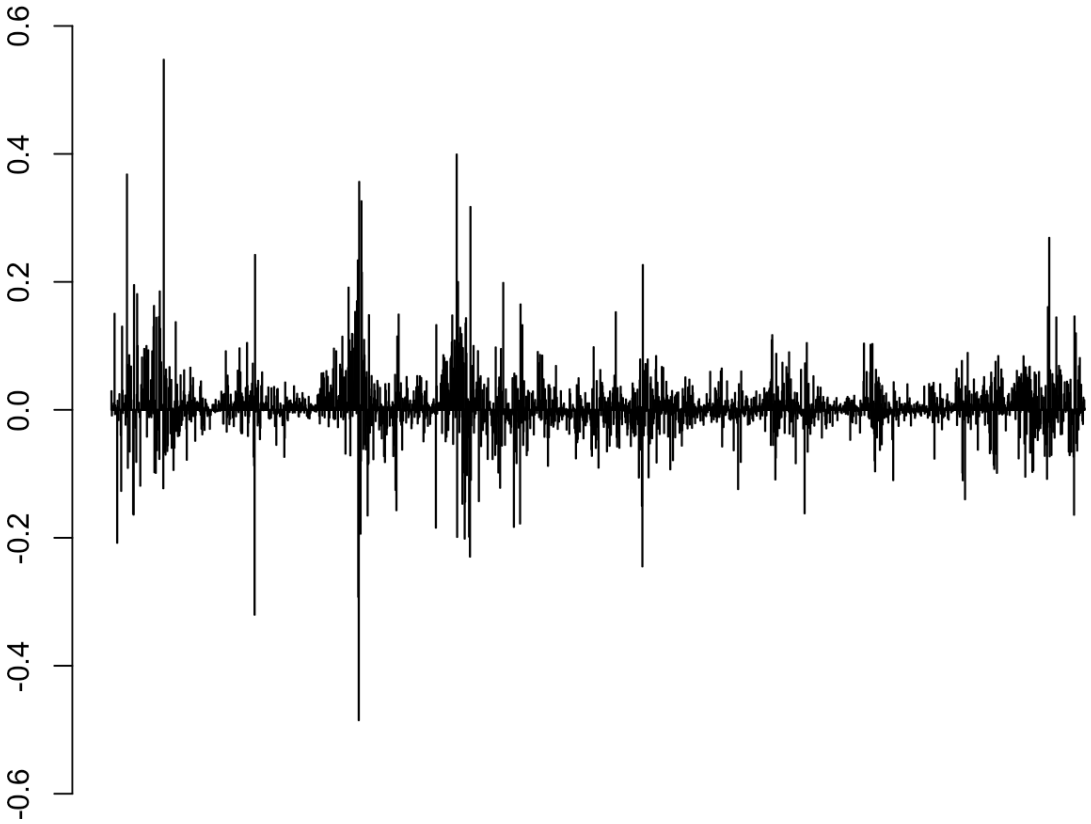


figure 7 Daily Change Rate

```
> 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.

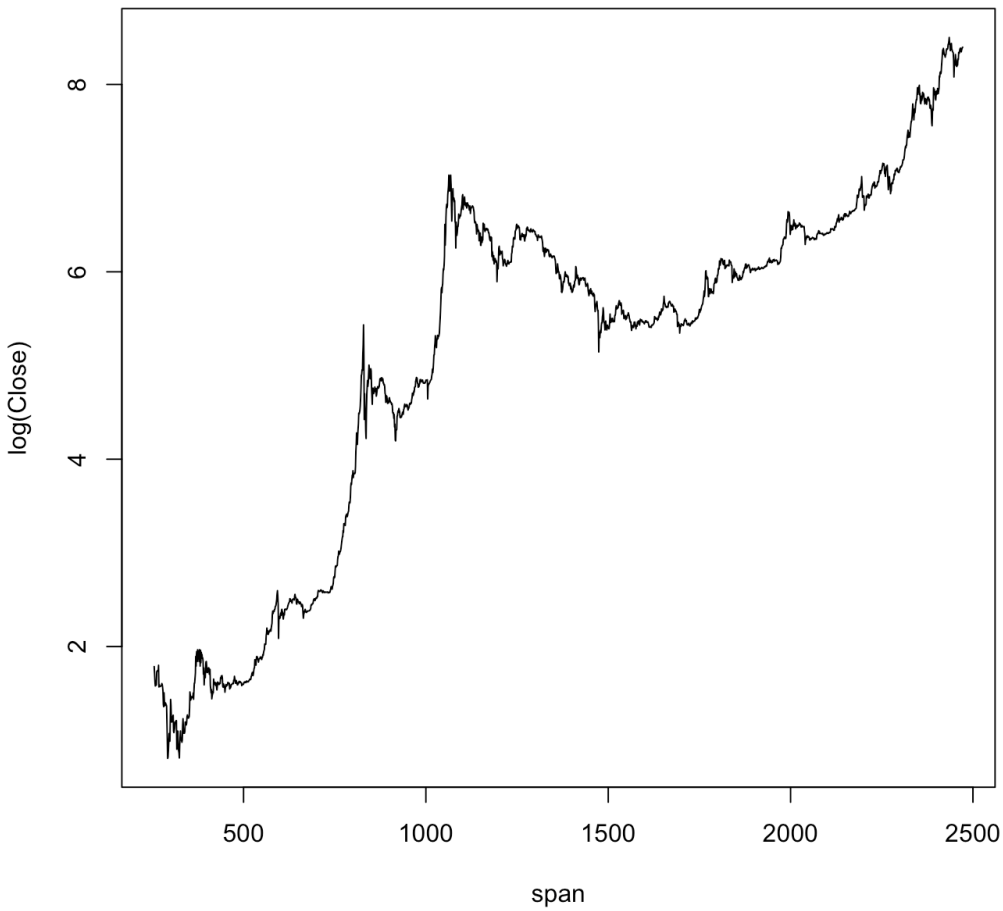


figure 8 Close Price vs. Log(span)

```
> 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

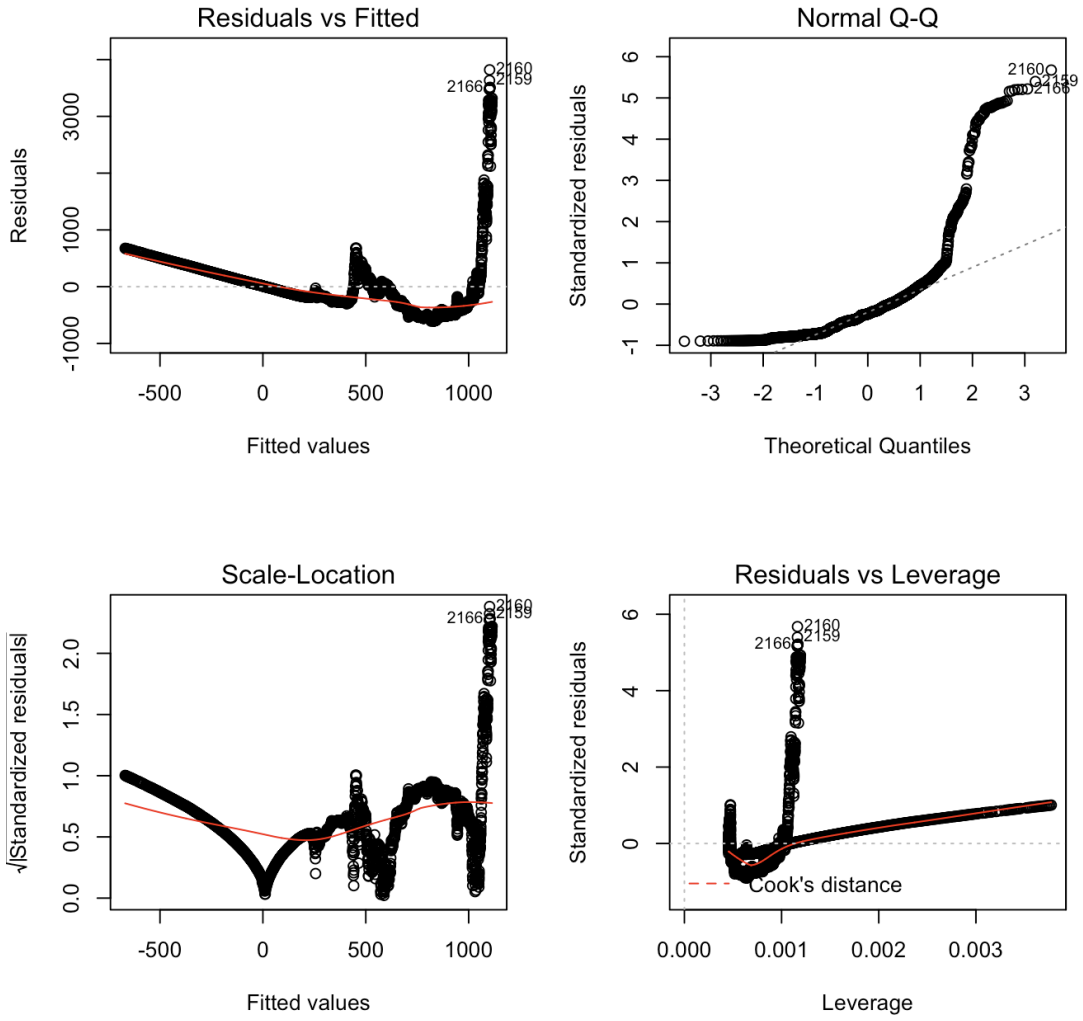


figure 9 linear analysis

Here, we find that the standard error of log(span) is so large that it's hard to say the two variables have a linear regression relation. As we further analyze the four fit graphs, we realize the relationship is weak enough to ignore.
See *figure 9*.

1) In the first graph, we found the points are combined with each other, which means non-linear relation is evident;
2) In the second graph, the point approximately gathered to build a clear line, it means the residual normality is good.
3) In the third one, the dots distribution has some regular features. Thus, based one that, we concluded the residual variance of the data is basically the same;

❖ Part III

In this part, we are trying to figure out the relationship between the prices of bitcoin and bulk commodities (gold & gas & oil).

1. Analyze Bitcoin with Gold & Oil & Gas

As we mentioned in the research questions, we tried to figure out if there is any relationship between bulk commodities and bitcoin price. The first thing we need to do is to add these outsource datasets to our original dataset.
Our outsource datasets including Gold, Gas and Oil in terms of Open Price, Close Price, High Price, Low Price as well as Volume. We combined these three independent datasets with original dataset by using 'left_join' function. The steps we followed are listed as below.

Step 1 Loaded new datasets

```
>data_Gold <- read_xlsx("/Users/yunli/Desktop/Project
1/bitcoin_price_v3.xlsx",sheet = "Gold")
>data_Oil <- read_xlsx("/Users/yunli/Desktop/Project
1/bitcoin_price_v3.xlsx",sheet = "WTI Oil")
>data_Gas <- read_xlsx("/Users/yunli/Desktop/Project
1/bitcoin_price_v3.xlsx",sheet = "Gas")
```

Step 2 Renamed column names of new datasets

Because new dataset has the same column names as the original dataset. For instance, they both use "Close" as their close price. We need to distinguish which is the closing price of the bitcoin and which is the closing price of gold/gas/oil.

Step 3 Combined new datasets with original dataset and Viewed final dataset

We used full_join function and connected three datesets by "Timestamp". Then, we are able to plot the variables together.

Step 4 Identified and Cleaned data issues

Our data covers a wide range of time, and when we use time as a field to merge data, there will be some mismatch. These "mismatch" may show as NA. So that we need to clean the data entry errors to ensure consistency of text.

```
> commodity <- na.omit(commodity)
> sum(which(is.na(commodity)))
[1] 0
```

At this stage, our final dataset is ready for further analysis.

2. Analytical Dataset Summary

Then, we tried to apply linear regression, as well as kernel regression to find which model fits better.
We compared the standard errors, ranges, coefficients from different models to interpret the data from visualization results.
The details are listed below:

1) Simple linear regression:

At first, we assumed that the price of Bitcoin and price of Gold & Gas & Oil have a linear regression relationship. So that we use the lm() function to perform linear regression with the formula Close ~ *. See *figure 10*.

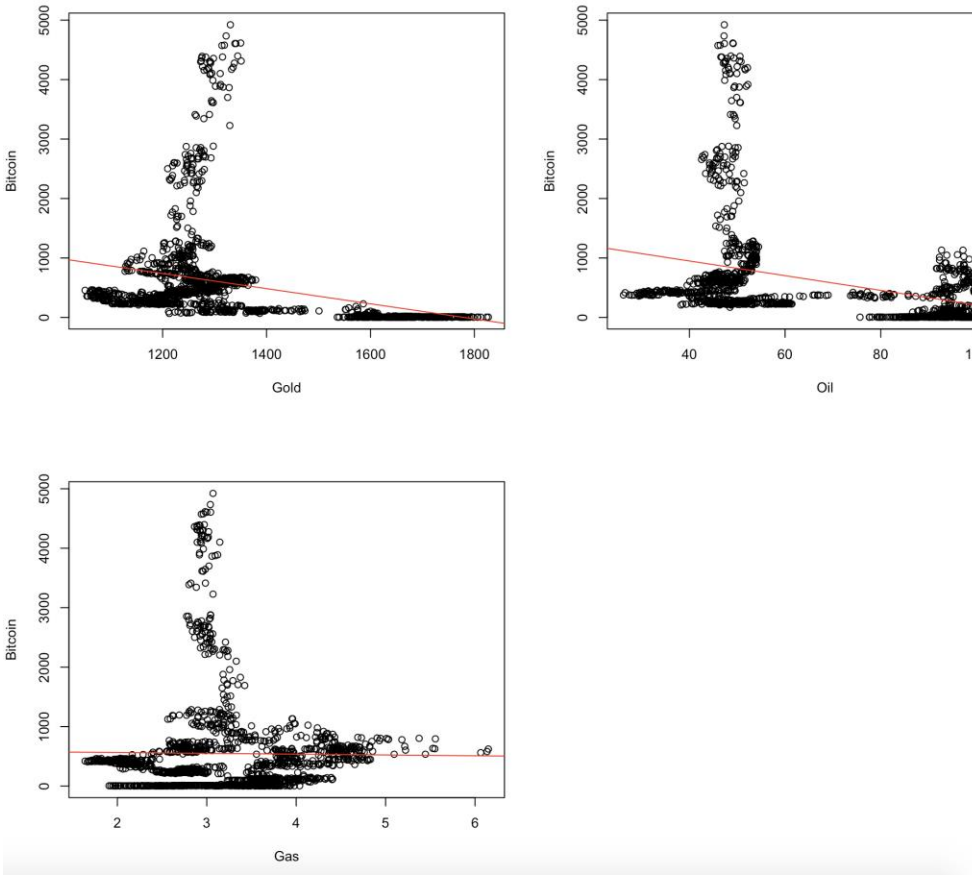


figure 10 simple linear regression

Then visualized this linear relationship between commodity price and Close price of Bitcoin as a line on the scatter plot between these two variables.

```
> summary(Modelgold)

Call:
lm(formula = commodity$Close ~ commodity$GoldClose)

Residuals:
   Min     1Q Median     3Q    Max
-656.1 -396.3 -160.8   11.5 4348.4

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        2268.21299  136.14213   16.66   <2e-16 ***
commodity$GoldClose   -1.27397    0.09977  -12.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 774.6 on 1561 degrees of freedom
Multiple R-squared:  0.09457,   Adjusted R-squared:  0.09399
F-statistic:   163 on 1 and 1561 DF,  p-value: < 2.2e-16
```

```
> summary(Modeloil)

Call:
lm(formula = commodity$Close ~ commodity$OilClose)

Residuals:
   Min     1Q Median     3Q    Max
-762.7 -410.5 -205.5  185.2 4061.1

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       1445.0251    57.9349   24.94   <2e-16 ***
commodity$OilClose  -12.3593     0.7539  -16.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 751.9 on 1561 degrees of freedom
Multiple R-squared:  0.1469,    Adjusted R-squared:  0.1463
F-statistic: 268.7 on 1 and 1561 DF,  p-value: < 2.2e-16
```

```
> summary(Modelgas)

Call:
lm(formula = commodity$Close ~ commodity$GasClose)

Residuals:
   Min     1Q Median     3Q    Max
-560.4 -453.9 -210.8   81.6 4372.1

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         591.50      90.65   6.525 9.17e-11 ***
commodity$GasClose  -13.64      27.63  -0.494    0.621
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 814 on 1561 degrees of freedom
Multiple R-squared:  0.0001562, Adjusted R-squared:  -0.0004843
F-statistic: 0.2439 on 1 and 1561 DF,  p-value: 0.6215
```

pic 4 statistical information of models

Residual standard errors are 774.6, 751.9, 814 separately, we would like to have this as low as possible. See *pic 4*.

R-squared indicates how well the predictor variable is able to capture the variation in the target variable about its mean. R2 takes values on a scale of 0–1; a score of 0 indicates that the predictor variable does not determine the target variable at all, whereas 1 indicates that the predictor variable perfectly determines the target variable. The above models have adjusted R2 of 0.09399, 0.1463, -0.0004843, indicating that Oil price is best for predicting price of bitcoin over 45% of the variance. Adjusted R2 statistic is the R2 statistic weighted by a factor involving the degrees of freedom.

Although the residual standard error of Oil price is the biggest, and the standard error of it is the least, which is 0.7539. The other two variables have too big Std. error to

get any linear relation with Bitcoin. Still, the linear relation between oil and bitcoin is so dim, since the adjusted R-squared is 0.1469 which is nearer to 0 instead of 1. Similarly, after we visualized the graphs in figure 8, we found that simple linear regression cannot well explained the relationship between bitcoin price and gold price.

2) Log-linear regression

We also applied log-linear regression as the second chance, but obviously there is weak linear relation between variables.
For example:
Residual standard error between oil and bitcoin is 756.9, which is not better than the simple linear regression model.

3) Locally Weighted Regression

We applied locally weighted regression as the third chance.

```
> ModelLOESS <- loess(commodity$Close ~ commodity$OilClose)
> summary(ModelLOESS)
Call:
loess(formula = commodity$Close ~ commodity$OilClose)

Number of Observations: 1563
Equivalent Number of Parameters: 4
Residual Standard Error: 722.3
Trace of smoother matrix: 4.33  (exact)

Control settings:
span      : 0.75
degree    : 2
family    : gaussian
surface   : interpolate   cell = 0.2
normalize   : TRUE
parametric  : FALSE
drop.square : FALSE
```

Compared with simple linear regression and log-linear regression, its residual standard error is 722.3, much larger than the previous two models.

4) Multivariate linear regression

Reviewing all these models we built between bulk commodities and bitcoin price were not well fitted. Then we use the lm() function to perform Multivariate linear regression with the formula Close ~ OilClose + GoldClose + GasClose.

```
Coefficients:
                          Estimate   Std. Error    t value     Pr(>|t|)
(Intercept)               374.6963    183.3133      2.044      0.0411 *
commodity$OilClose        -21.2605      1.2987    -16.370      <2e-16 ***
commodity$GasClose        421.0731     34.9932     12.033      <2e-16 ***
commodity$GoldClose         0.2747      0.1324      2.075      0.0382 *
```

Form the list upon, the standard errors of the three varibles varies, the standard error of gas price is extremely big, thus we delete the gas ingredient to do multi linear regression.

```
Call:
lm(formula = commodity$Close ~ commodity$OilClose +
commodity$GoldClose)

Residuals:
   Min     1Q     Median     3Q        Max
-779.6  -350.6    -154.6    117.4     4110.0

Coefficients:
                          Estimate   Std. Error  t value     Pr(>|t|)
(Intercept)               1938.3783   135.1212   14.345     < 2e-16 ***
commodity$OilClose        -10.0409     0.9448   -10.627     < 2e-16 ***
commodity$GoldClose        -0.4900     0.1214    -4.037     5.67e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 748.3 on 1560 degrees of freedom
Multiple R-squared:  0.1557,        Adjusted R-squared:  0.1546
F-statistic: 143.8 on 2 and 1560 DF,  p-value: < 2.2e-16
```

Residual standard error is still large, but the Std. Error is relatively small. And still, the Adjusted R-squared is 0.1546, quite small. Although this model seems to fit much better than single linear regression, there is weak linear relation within variables.

5) Linear regression with total volume currency of Bitcoin

At last, we analyzed the total volume currency of bitcoin in each day to discuss the linear relationship between them.

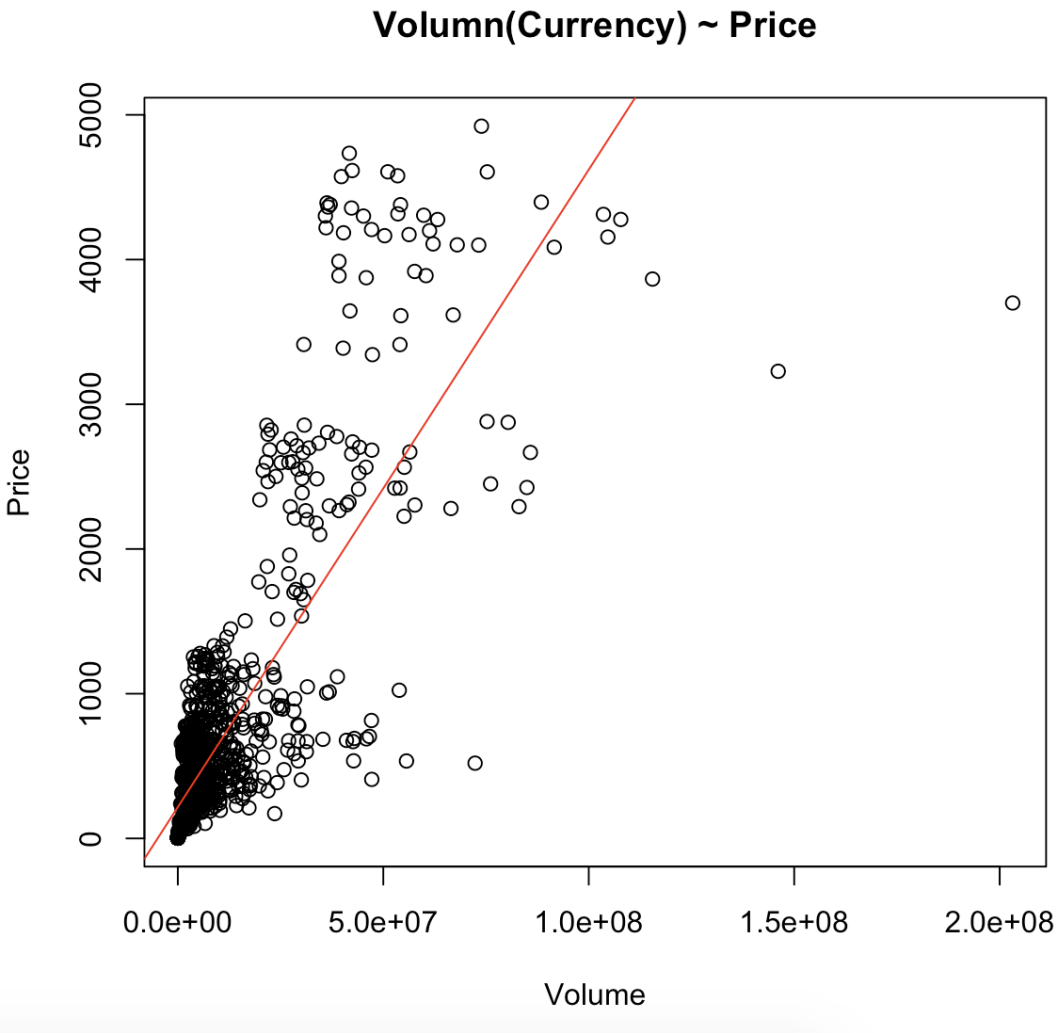


figure 11 Volumn(Currency) ~ Price

Actually, we found the two variables are quite linear-related. See *figure 11*.

```
> summary(model_currency)
Call:
lm(formula = commodity$Close ~ commodity$`Volume (Currency)`)
Residuals:
   Min     1Q  Median     3Q    Max
-5463.6  -209.0  -118.5   152.0  2680.1

Coefficients:
                        Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)           2.167e+02   1.361e+01    15.92      <2e-16 ***
```

```
commodity$`Volume (Currency)` 4.404e-05  8.150e-07   54.03  <2e-
16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 480.5 on 1561 degrees of freedom
Multiple R-squared:  0.6516,   Adjusted R-squared:  0.6514
F-statistic:  2920 on 1 and 1561 DF,  p-value: < 2.2e-16
```

Residual standard error is 480.5 lower than any model we built before. Plus the multiple R-squared is 0.6514 means all these factors are able to explain over 65.14% of the variance in price.
The standard error is also small enough to get the linear regression feature of the variables.

In a word, linear regression including simple linear regression, multivariate linear regression and log-linear regression cannot exactly explain the relationship between bitcoin price and bulk commodities.

We definitely find somehow linear relation between bitcoin currency volume and its price, but it also confused us why the relation is a positive sloop line.

After all, we need to explore more to find out a fitter relationship between bitcoin price with other bulk commodities.

❖ Part IV

Alignment with External Sources

We did a lot of background research about this question, and two of those studies coincide our pre-assumption that there is a log-linear regression between bitcoin price and time. According to analysis by Dennis Porto, a bitcoin investor and Harvard academic, bitcoin's price would double every eight months if it continues to follow one of tech's "golden rules" — Moore's law.

The first graph represents the log-linear regression about the number of days between the date of each piece of data from the start date and close price of bitcoins. The second graph shows that price of bitcoin over time on a logarithmic scale. These two graphs show the same trend over time which come to the relationship between time and price of bitcoin.

The links of these two studies are listed as below:

- http://www.businessinsider.com/bitcoin-price-and-moores-law-2017-8
- https://99bitcoins.com/price-chart-history/

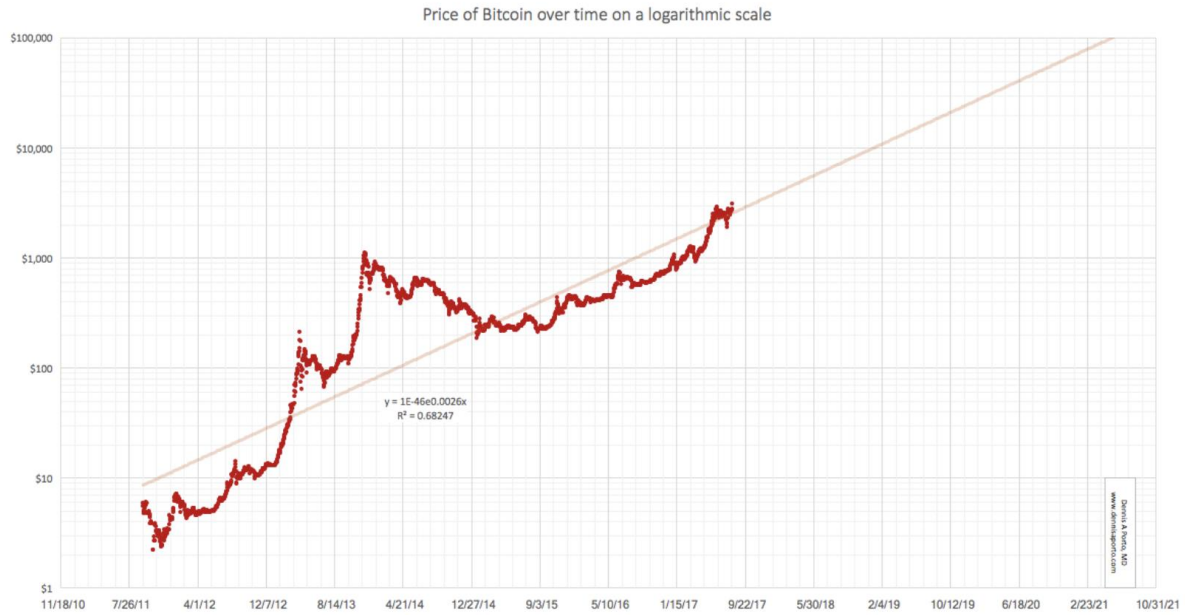


figure 12 Price of Bitcoin over time on a logarithmic scale (from Business Insider)

❖ Part V Reference

- Bitcoin can get to $100,000 if it keeps following one of tech's golden rules (http://www.businessinsider.com/bitcoin-price-and-moores-law-2017-8)
- Bitcoin price chart with historic events (https://99bitcoins.com/price-chart-history/)
- Pathak, M. Beginning Data Science with R (Springer, 2014)
- Wickham & Grolemund. R for Data Science (O'Reilly Media, Inc., 2016)