

Statistical Analysis of Share Market Prices

by

Aneek Sarkar

*Project submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Statistics

Under the supervision of

Debashis Chatterjee



DEPARTMENT OF STATISTICS
VISVA-BHARATI UNIVERSITY

© *Aneek Sarkar*
All rights reserved

DECLARATION

Project Title Statistical Analysis of Share Market Prices
Authors *Aneek Sarkar*
Student IDs
Supervisor Debashis Chatterjee

We declare that this project entitled *Statistical Analysis of Share Market Prices* is the result of our own work except as cited in the references. The project has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Aneek Sarkar

Department of Statistics
Visva-Bharati University

Date: 24/05/2023

ACKNOWLEDGEMENTS

I am greatly indebted to so many people for helping me in the preparation of this project. I owe a deep debt of gratitude to my supervisor Dr. Debashis Chatterjee for necessary guidance, for this presentation of this dissertation, valuable comments and suggestions. I am extremely grateful to him for the necessary stimulus, support and valuable time.

Special thanks to Dr. Sudhansu Sekhar Maiti, Head of the Department of Statistics, Visva Bharati University. I am greatly indebted to Dr. Arindam Chakraborty, Dr. Tirthankar Ghosh, Dr. Saran Ishika Maiti, Dr. Soumalaya Mukhopadhyay, Dr. Sourav Rana; Faculty members often took pain and stood by me in adverse circumstances. Without their encouragement and inspiration, it was not possible for me to complete this project. Finally, my earnest thanks go to my friends who were always beside me when I needed them without any excuses and made these three years worthwhile. This project is not only a mere project. It is the memories spend with the whole department which has created a mutual understanding among us. There are many emotions related to this piece of work, especially respect and duty towards teachers and vice versa, educational attachment with my friends and social attachment with my college.

I am extremely thankful to my parents for their unconditional love, endless prayers, caring and immense sacrifices for educating and preparing me for my future. I would like to say thanks to my friends and relatives for their kind support and care.

Finally, I would like to thank all the people who have supported me to complete the project work directly or indirectly.

Aneek Sarkar

Visva-Bharati University

Date: 24/05/2023

Dedicated to
my Department

– *Aneek Sarkar*

ABSTRACT

I have conducted statistical forecasting on the share market prices of Microsoft (MSFT) and Starbucks (SBUX) using ARIMA, Holt-Winters, and ETS methods on real data from 2004-2014.

After analyzing the data, I fitted ARIMA models to both MSFT and SBUX share prices. The ARIMA models suggested that both MSFT and SBUX share prices exhibited seasonality and trend. The models forecasted that MSFT share prices would continue to rise, with occasional fluctuations, and that SBUX share prices would experience a decrease and then a gradual increase towards the end of the time period.

Next, I used Holt-Winters models to forecast MSFT and SBUX share prices. The models showed that MSFT share prices had an upward trend with an increase in volatility, while SBUX share prices exhibited a decreasing trend with a decrease in volatility. Overall, the models predicted that MSFT share prices would increase over time, while SBUX share prices would remain relatively stable.

Lastly, I used ETS models to forecast MSFT and SBUX share prices. The models revealed that MSFT share prices had a positive trend with a high level of volatility, while SBUX share prices had a negative trend with a low level of volatility. The models predicted that MSFT share prices would continue to increase, while SBUX share prices would experience a slight decline before stabilizing towards the end of the time period.

Overall, the statistical forecasting using ARIMA, Holt-Winters, and ETS methods on real data from 2004-2014 suggests that MSFT share prices will continue to rise, albeit with occasional fluctuations, while SBUX share prices will remain relatively stable with a slight decline towards the end of the time period. However, it's important to note that forecasting share prices is inherently uncertain, and other factors beyond the scope of statistical modeling can also influence share prices.

Keywords: ARIMA, HoltWinters,ETS,Forecasting

Contents

1	Introduction	1
1.1	About The Dataset	3
1.2	Aims, objectives and Motivation	3
1.2.1	Aims	3
1.2.2	Motivation	4
1.3	Project Specification	4
2	Methodology	6
2.1	Fitting of the data	7
2.2	checking stationarity	7
2.2.1	ADF test	8
2.3	Modify the data	9
2.4	ACF PACF	12
2.5	Delimit Train and Test	16
2.6	ARIMA	19
2.6.1	ARIMA Parameters	20
2.6.2	ARIMA and Stationary data	20
2.6.3	Usage	21
2.6.4	How Does ARIMA Forecasting Work?	21
2.6.5	Bottom Line	21
2.6.6	Order of ARIMA	21
2.6.7	ARIMA fit	24
2.7	Checking Residuals	27
2.8	Forecast Using Holt-Winters	31
2.9	Forecast Using ETS Method	36
2.9.1	Forecasting Using ETS	38

2.10	Checking Accuracy	40
2.11	Further Test	41
3	Interpretation	42
3.1	Stationarity	42
4	Conclusion	43

Chapter 1

Introduction

WHAT IS SHARE MARKET?

The stock market is a vital component of economy of a country. It is a place where publicly traded companies' stocks are bought and sold, allowing investors to buy shares of a company's stock and take ownership of a small piece of that company. The stock market is a key indicator of the overall health of the economy, as the performance of companies listed on the stock market reflects the current economic conditions. Here we are comparing world's two famous companies to compare their holdings, market value and growth to analyse which company is the highest earning or increasing company now a days. Those companies are Microsoft and Starbucks. Here we are analysing those companies for a fixed time-period and in same condition. We take the values from yahoo search engine.

figure

Primary Market:

This where a company gets registered to issue a certain amount of shares and raise money. This is also called getting listed in a stock exchange.

Secondary Market:

Once new securities have been sold in the primary market, these shares are traded in the secondary market. This is to offer a chance for investors to exit an investment and sell the shares. Secondary market transactions are referred to trades where one investor buys shares from another investor at the prevailing market price or at whatever price the two parties agree upon.

What Is Equity?

Equity is the money that would remain if all assets of a company were liquidated and all its debt was paid off. In short, equity is a company's net assets minus its

liabilities. Equity may also be referred to as shareholders' equity or stockholders' equity. It stands for the residual claim of a company owner after debts have been paid. Here's an example to explain what is equity: Take a look at the total assets and liabilities of ABC Corporation on 30 September 2019. Its total assets on that date stood at Rs 1 lakh. Meanwhile, its total liabilities—including loans and taxes—amounted to Rs 75,000. So, the equity of ABC Corporation on 30 September 2019 is Rs 25,000 (i.e. Rs 1 lakh – Rs 75,000).

What Is Equity?

Equity is the money that would remain if all assets of a company were liquidated and all its debt was paid off. In short, equity is a company's net assets minus its liabilities. Equity may also be referred to as shareholders' equity or stockholders' equity. It stands for the residual claim of a company owner after debts have been paid.

Here's an example to explain what is equity:

Take a look at the total assets and liabilities of ABC Corporation on 30 September 2019. Its total assets on that date stood at Rs 1 lakh. Meanwhile, its total liabilities—including loans and taxes—amounted to Rs 75,000. So, the equity of ABC Corporation on 30 September 2019 is Rs 25,000 (i.e. Rs 1 lakh – Rs 75,000).

what is return?

Return is the gain or loss that an investment generates over a period of time. A positive return indicates a profit while a negative return a loss.

What Is Return on Investment (ROI)?

Return on investment (ROI) is a performance measure used to evaluate the efficiency or profitability of an investment or compare the efficiency of a number of different investments. ROI tries to directly measure the amount of return on a particular investment, relative to the investment's cost.

To calculate ROI, the benefit (or return) of an investment is divided by the cost of the investment. The result is expressed as a percentage or a ratio.

KEY TAKEAWAYS

Return on Investment (ROI) is a popular profitability metric used to evaluate how well an investment has performed. ROI is expressed as a percentage and is calculated by dividing an investment's net profit (or loss) by its initial cost or outlay. ROI can be used to make apples-to-apples comparisons and rank investments in different projects or assets. ROI does not take into account the holding period

or passage of time, and so it can miss opportunity costs of investing elsewhere. Whether or not something delivers a good ROI should be compared relative to other available opportunities.

How to Calculate Return on Investment (ROI)

The return on investment (ROI) formula is as follows:

$$ROI = (\text{Current Value of Investment} - \text{Cost of Investment}) / \text{cost of investment}$$

Reason behind choosing two different sectors to compare

- First of all, if I select two companies from the same sector then the market of those companies will overlap with each other. I will have to face an interaction between two markets, as the inner products will be same and the companies will not be independent in market as well as capitalization and selling of products.
- If we choose both same sectors, the selling market (e.g. a continent or a country). It have a big impact on the share price. Even the currency of the main market have a big impact (e.g. the difference between dollar and rupee is very significant in share market).

Hence I use two different sectors to compare in stock market.

1.1 About The Dataset

Basically ,this data has been taken from R-Directory (code is given in appendix). R collected this data as an 'xts' object from "yahoo" search engine. For conformation I have rechecked the printed data from the online information available for public query. Hence I can say that this data , though it is taken from R directory, is 100% accurate to the actual public data.

1.2 Aims, objectives and Motivation

1.2.1 Aims

Aim is to identify the growth of that selected companies to study the market and forecast the stock price of those companies. Objective To find the plot of comparison between the Prices enlisted in share market and to analyse data by

statistical method. By this, we can see either MSFT (Microsoft) or SBUX (Starbucks) is profitable. secondly, the aim is also to check whether the forecast is precise for this kind of real life data.

1.2.2 Motivation

Basically stock market (or share price) shows the growth of the company and indicate the gross increase or decrease of price index for that companies. So by analysing them we can easily conclude that whether a company is in profit or loss to make a clear idea for investment to the investor. Here the main motivation to take two companies that are fully independent of each others cause if those companies, chosen for analysis, are of same field or same type (e.g. both electronics making company) then the analysis won't be unbiased as somehow they are connected to each other.

1.3 Project Specification

- This project is based on Share market price of MSFT aka Microsoft and SBUX aka Starbucks. The data of these companies means the share price and details I needed ,has been collected from R directory using some specific code that has been declared in the CODE details.
- From this project we can specified that which of these company is profitable and by forecasting those old data, we can see in which company we should invest. Hence in simple word, this code can be used for part time or full time trading to analyse which company is best (at that time period) to invest and when to put out the money from the stock.
- This project needs all the previous stock prices as data and also need some specific code that can complete the analysis and give result. Hence the specific required items are stock price, code and a minimum statistical knowledge to understand how the project works on.
- This project is basically based on two different sectors of companies, one is Microsoft and another one is Starbucks, first one based on electronics and later one is simple marketing (a multi-million coffee shop spread all over

the world), just to avoid the problem of overlapping market and interaction between those two similar companies.

Chapter 2

Methodology

Here we are doing these following steps:

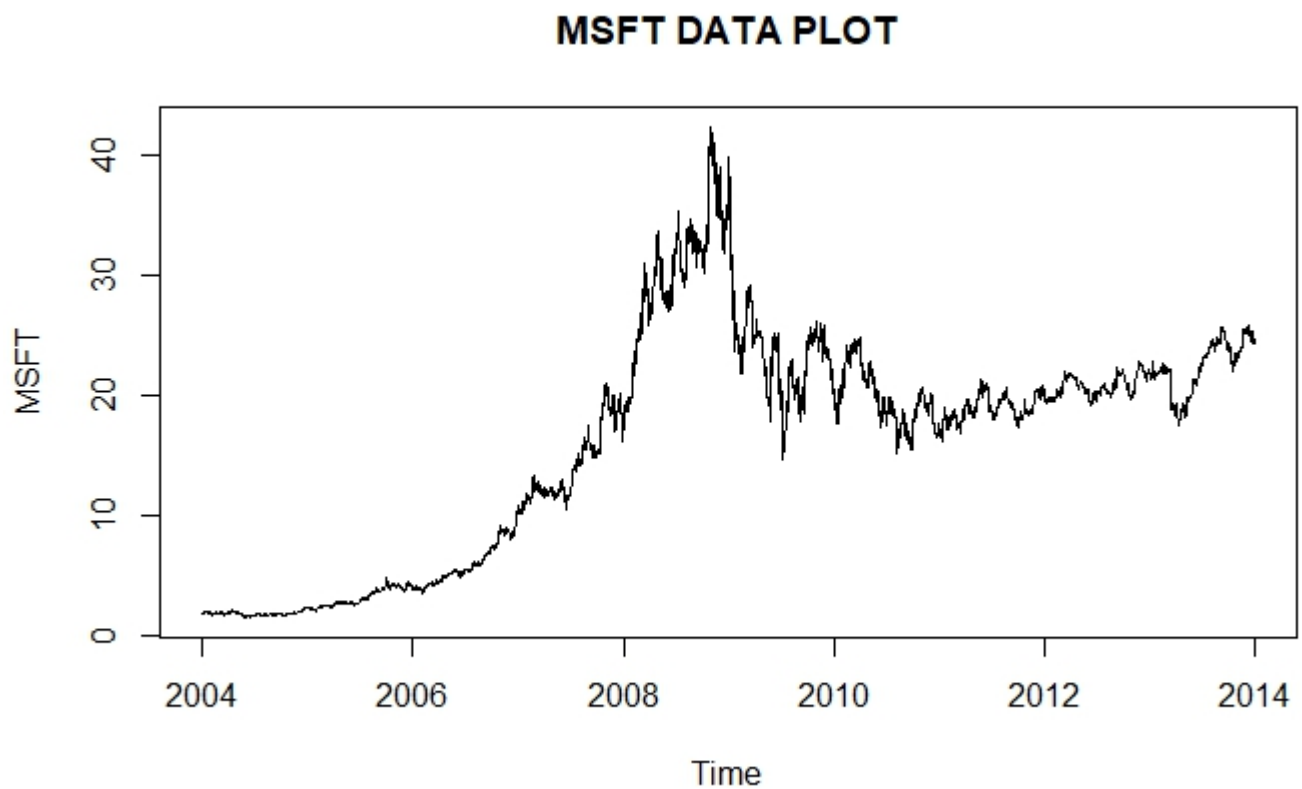
- Collecting the data from r directory,
- omitting those empty or faulty values,
- changing the data into time series model to fit them properly,
- Fitting Data,
- checking their stationarity,
- Modify the data,
- ACF and PACF,
- Delimit Training and testing range
- fitting the ARIMA model,
- checking their residual,
- Forecast using HoltWinters model,
- Forecast using ETS Model,
- checking the accuracy.

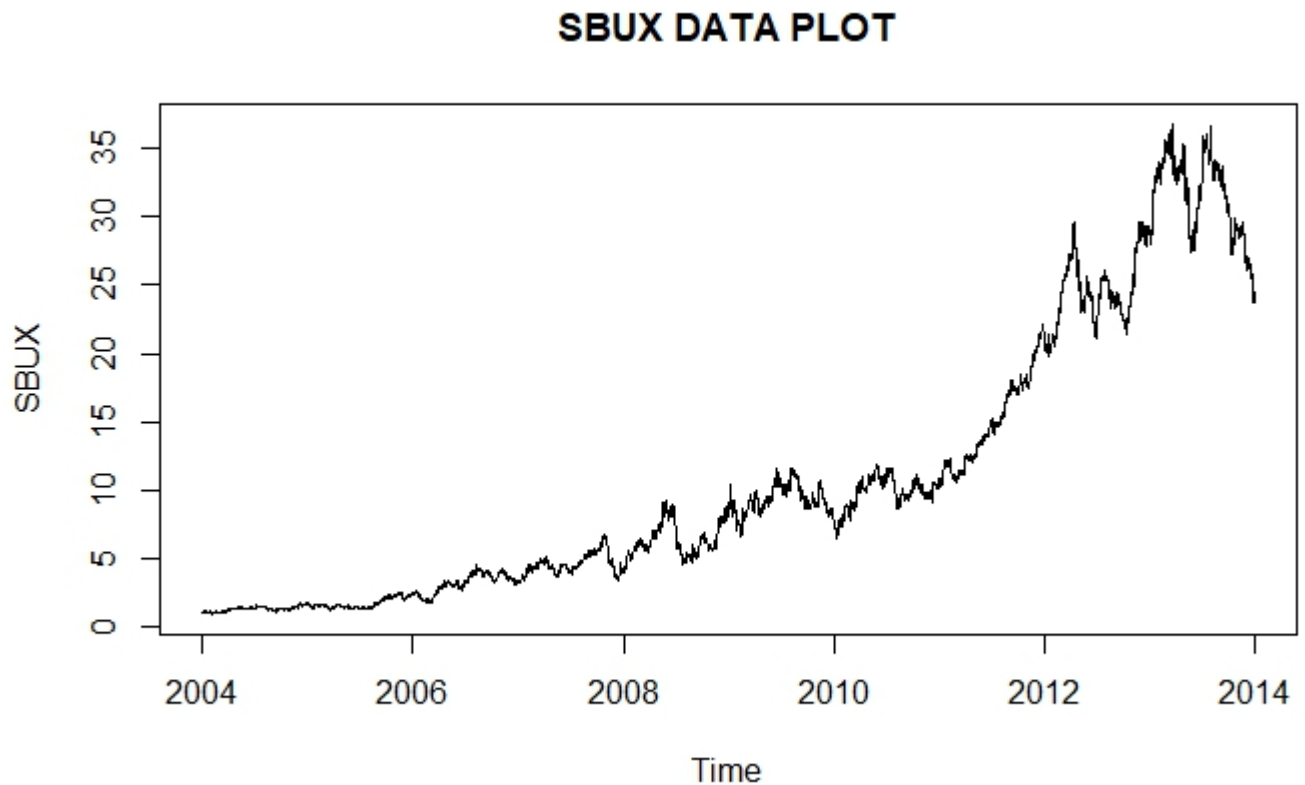
2.1 Fitting of the data

I got the raw data from r directory using xrs library

2.2 checking stationarity

After plotting the data, I have got this following graphs,





by this graphs/plots , I can see that there is a drop in both microsoft and starbucks market price. So I can say this are non stationary. I also have Augmented Dickey–Fuller test or adf test to check it out.

2.2.1 ADF test

In statistics, an augmented Dickey–Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity. It is an augmented version of the Dickey–Fuller test for a larger and more complicated set of time series models.

The augmented Dickey–Fuller (ADF) statistic, used in the test, is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence.

The testing procedure for the ADF test is the same as for the Dickey–Fuller test but it is applied to the model

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \dots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t, \quad \text{where } \alpha \text{ is}$$

a constant, β the coefficient on a time trend and p the lag order of the autoregressive process. Imposing the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk and using the constraint $\beta = 0$ corresponds to modeling a random walk with a drift. Consequently, there are three main versions of the test, analogous to the ones discussed on Dickey–Fuller test (see that page for a discussion on dealing with uncertainty about including the intercept and deterministic time trend terms in the test equation.)

Now from the adf test, I got this output

```
> adf.test(msft_data)
```

Augmented Dickey-Fuller Test

```
data: msft_data
Dickey-Fuller = -2.1821, Lag order = 15, p-value = 0.5012
alternative hypothesis: stationary
```

```
> adf.test(sbox_data)
```

Augmented Dickey-Fuller Test

```
data: sbox_data
Dickey-Fuller = -2.3015, Lag order = 15, p-value = 0.4507
alternative hypothesis: stationary
```

Here the alternative hypothesis is being rejected. Hence the data is non-stationary.

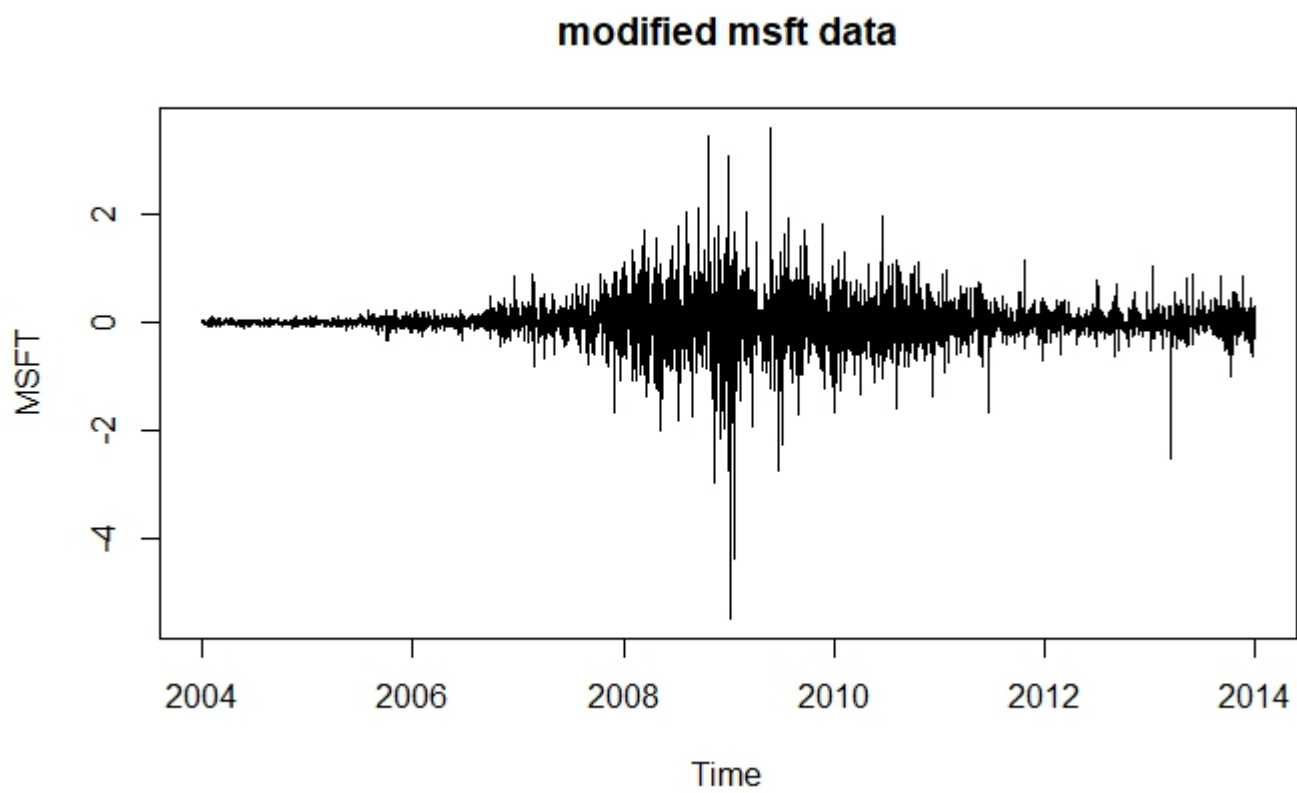
Though Working with non-stationary model is not in my syllabus , I have studied a little about it and proceed to finish my project work.

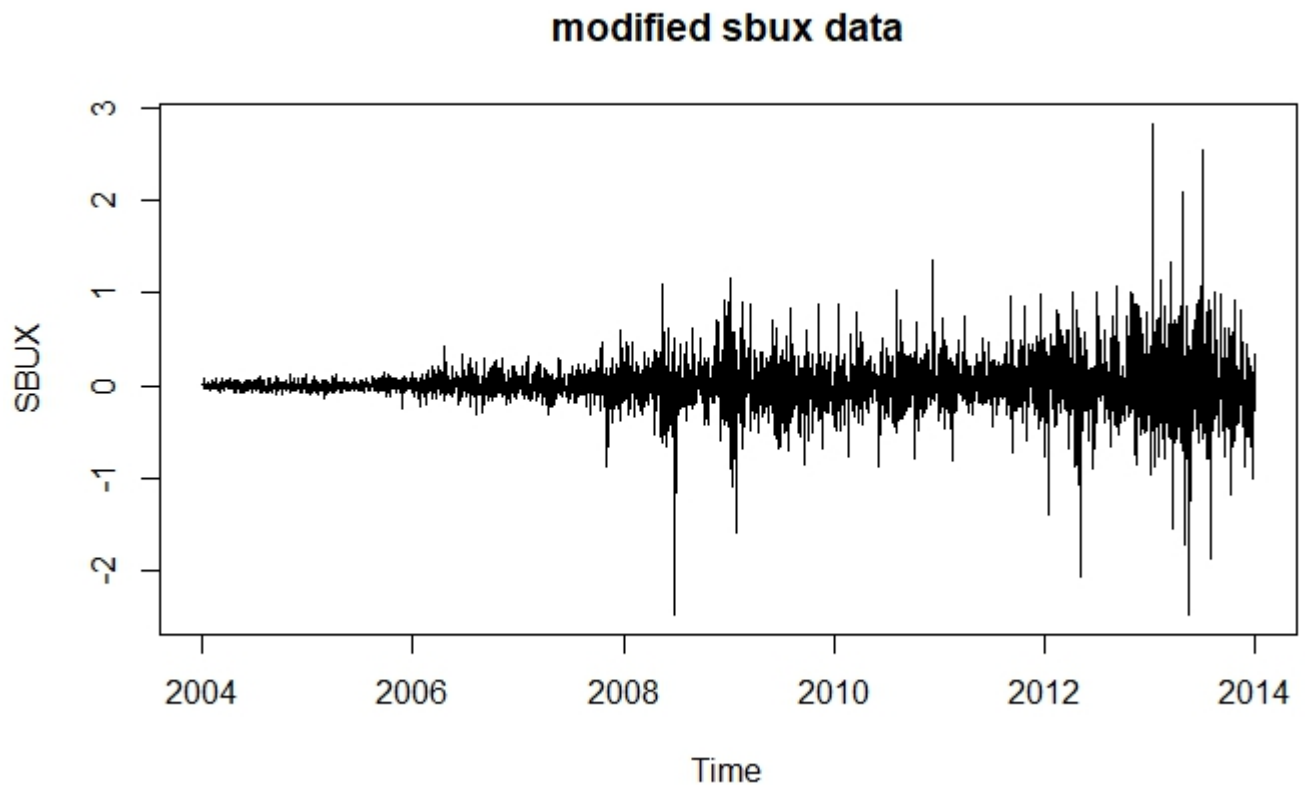
2.3 Modify the data

Now to modify the data (making it usable for testing and fitting) I use difference method as follows,

```
#difference
modified_msft_data<- diff(msft_data)
modified_sbox_data<- diff(sbox_data)
```

By this, I got the following plots,





Now I did the **ADF** test for those modified data and get the following results,
`> adf.test(modified_msft_data)# p-value <0.05, series is stationary.`

Augmented Dickey-Fuller Test

data: modified_msft_data
 Dickey-Fuller = -14.104, Lag order = 15, p-value = 0.01
 alternative hypothesis: stationary

Warning message:
 In `adf.test(modified_msft_data)` : p-value smaller than printed p-value
`> adf.test(modified_sbux_data)#p value is < 0.05 , series is stationary.`

Augmented Dickey-Fuller Test

data: modified_sbux_data
 Dickey-Fuller = -12.956, Lag order = 15, p-value = 0.01
 alternative hypothesis: stationary

Warning message:
 In `adf.test(modified_sbux_data)` : p-value smaller than printed p-value

Now this P values ,got from the above adf test says, the alternative hypothesis is accepted here. Hence the data is stationary.

Now I can do the other tests and fits for further details.

2.4 ACF PACF

Autocorrelation function (ACF) and Partial Autocorrelation Function (PACF, also called Partial ACF) are important functions in analyzing a time series. They generally produce plots that are very important in finding the values p, q and r for Autoregressive (AR) and Moving Average (MA) models.

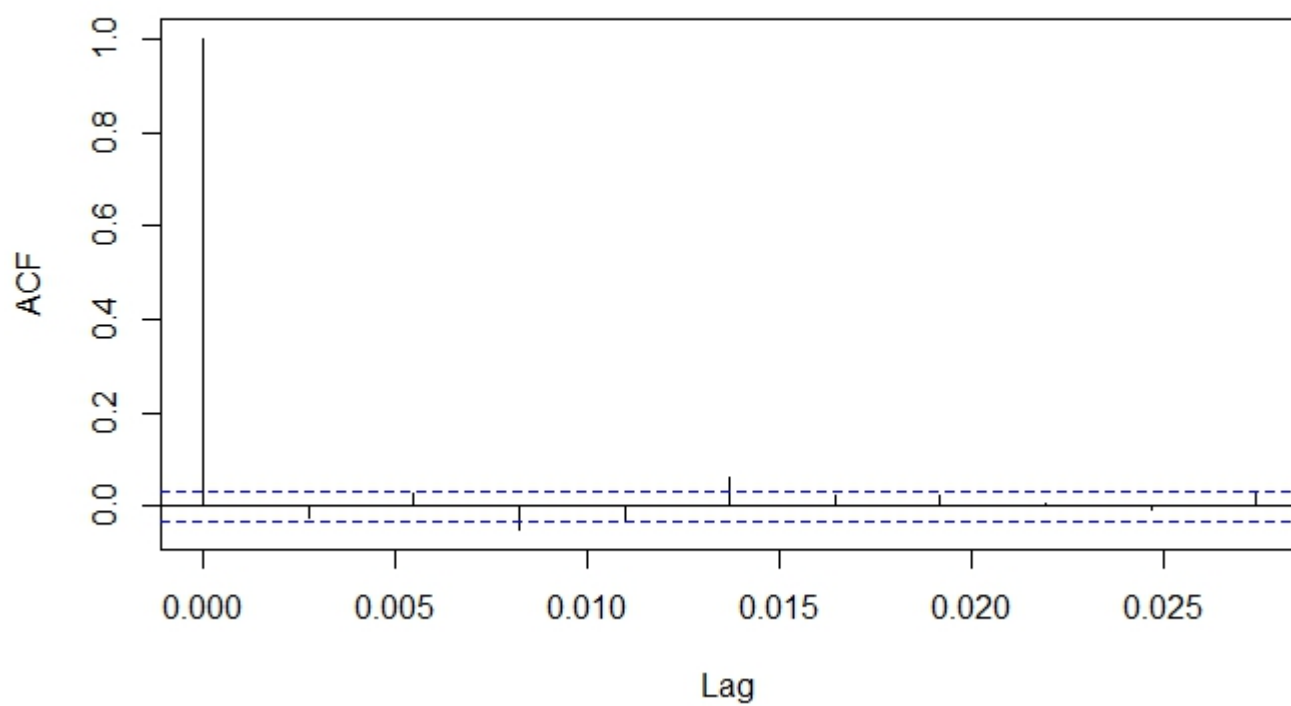
An ACF measures and plots the average correlation between data points in time series and previous values of the series measured for different lag lengths.

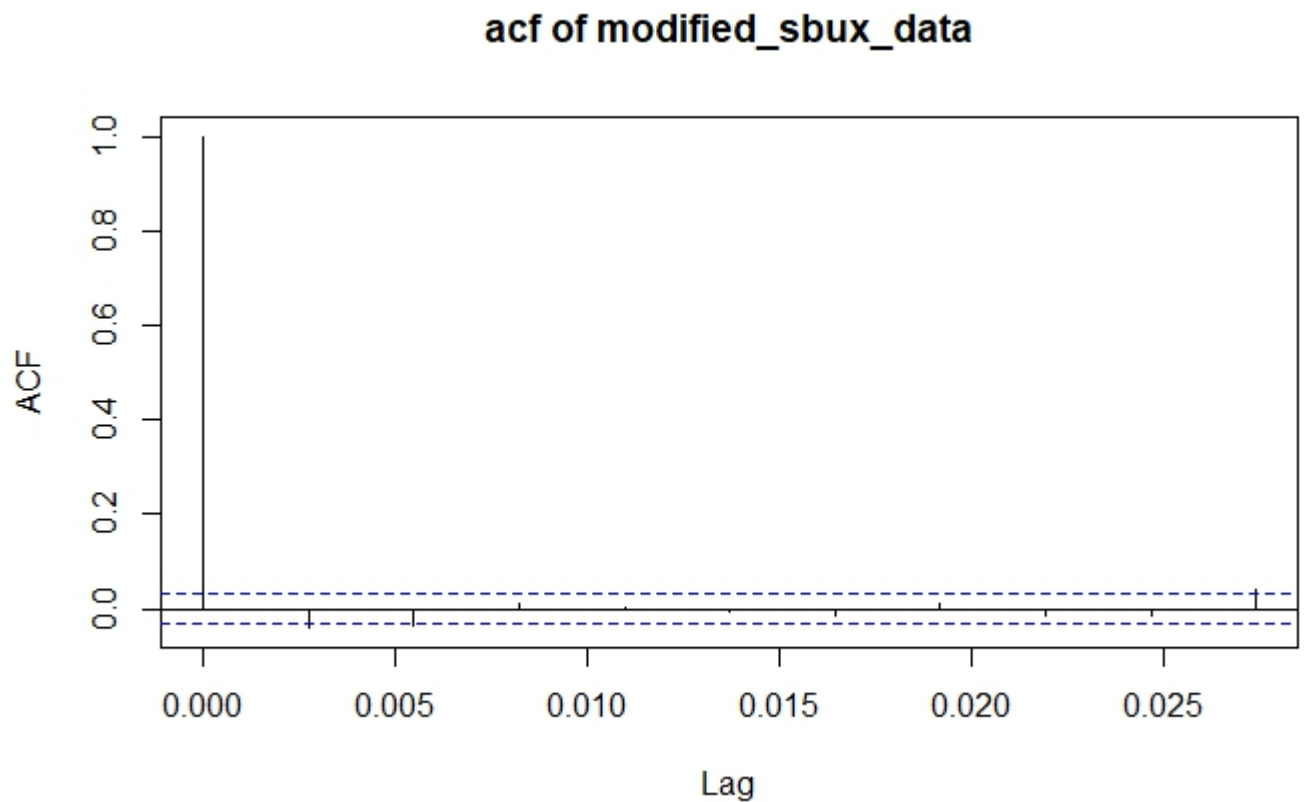
A PACF is similar to an ACF except that each partial correlation controls for any correlation between observations of a shorter lag length.

The value for an ACF and a PACF at the first lag are the same because both measure the correlation between data points at time t with data points at time $t-1$. However, at the second lag, the ACF measures the correlation between data points at time t with data points at time $t-2$, while the PACF measures the same correlation but after controlling for the correlation between data points at time t with those at time $t-1$.

At first I used acf and pacf for the raw data and get this values,

acf of modified_msft_data

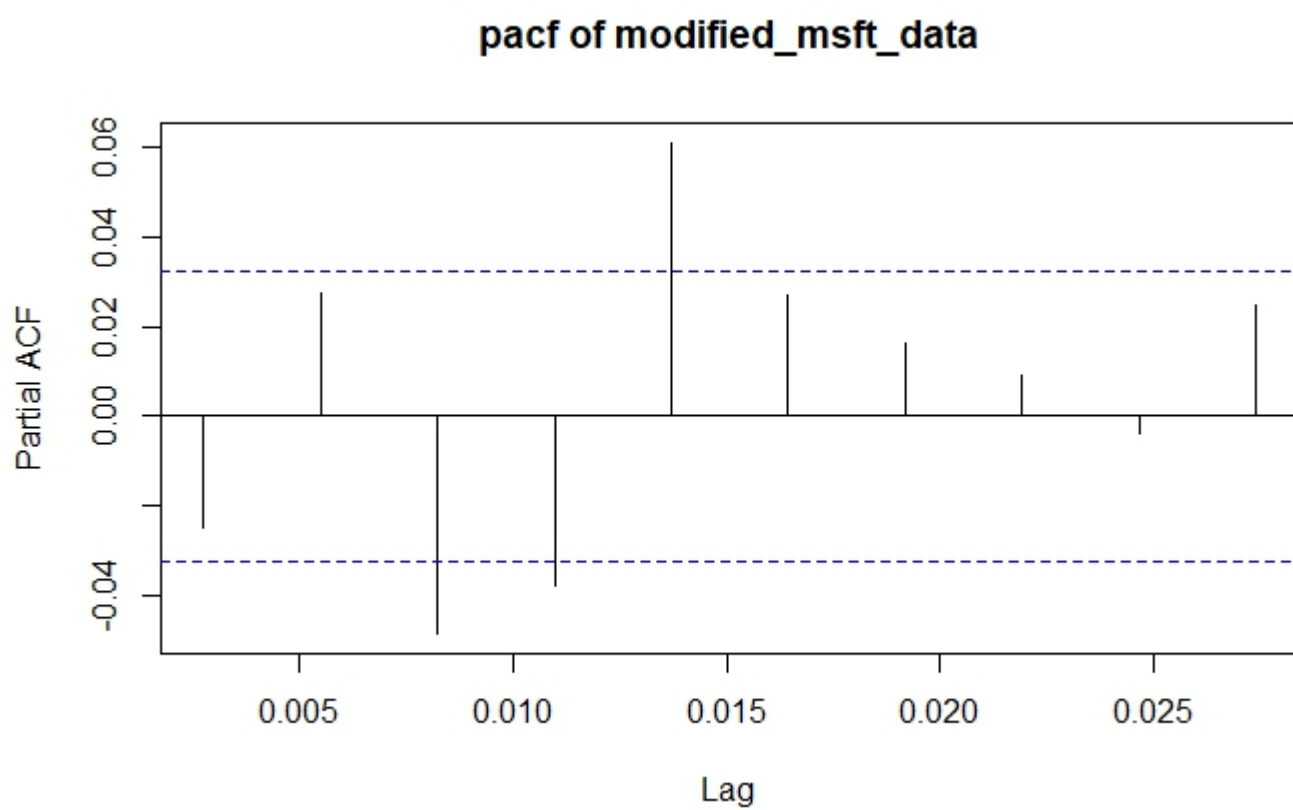


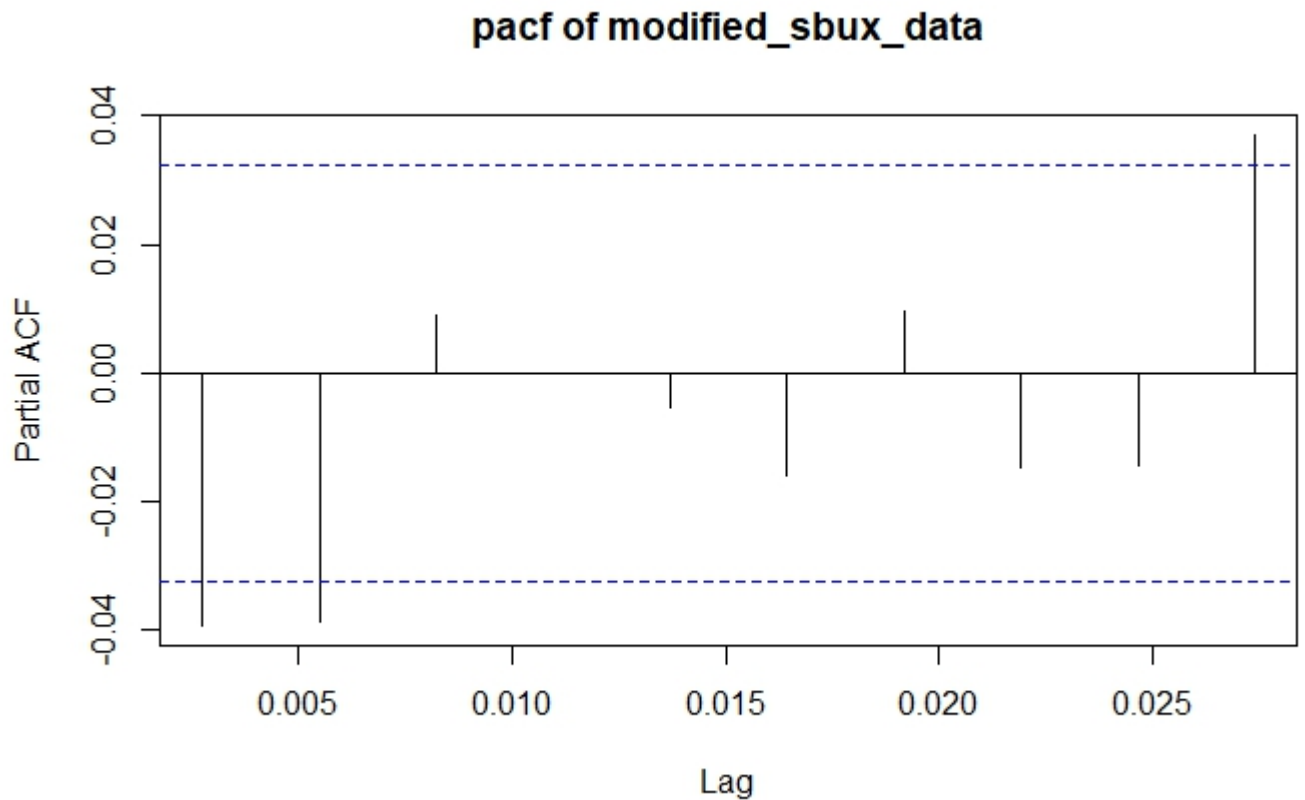


As we can see, the estimated correlations in the ACF do decay to zero. This suggests that the time series is stationary, in both case.

Hence the difference is perfect and the non-stationary data is workable for further tests.

By using `pacf`, we just check the order to fit the data in **ARIMA** model.





2.5 Delimit Train and Test

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

It can be used for classification or regression problems and can be used for any supervised learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

Train Dataset: Used to fit the machine learning model. Test Dataset: Used to evaluate the fit machine learning model. The objective is to estimate the performance of

the machine learning model on new data: data not used to train the model.

This is how we expect to use the model in practice. Namely, to fit it on available data with known inputs and outputs, then make predictions on new examples in the future where we do not have the expected output or target values.

The train-test procedure is appropriate when there is a sufficiently large dataset available.

When to Use the Train-Test Split The idea of “sufficiently large” is specific to each predictive modeling problem. It means that there is enough data to split the dataset into train and test datasets and each of the train and test datasets are suitable representations of the problem domain. This requires that the original dataset is also a suitable representation of the problem domain.

A suitable representation of the problem domain means that there are enough records to cover all common cases and most uncommon cases in the domain. This might mean combinations of input variables observed in practice. It might require thousands, hundreds of thousands, or millions of examples.

Conversely, the train-test procedure is not appropriate when the dataset available is small. The reason is that when the dataset is split into train and test sets, there will not be enough data in the training dataset for the model to learn an effective mapping of inputs to outputs. There will also not be enough data in the test set to effectively evaluate the model performance. The estimated performance could be overly optimistic (good) or overly pessimistic (bad).

If you have insufficient data, then a suitable alternate model evaluation procedure would be the k-fold cross-validation procedure.

In addition to dataset size, another reason to use the train-test split evaluation procedure is computational efficiency.

Some models are very costly to train, and in that case, repeated evaluation used in other procedures is intractable. An example might be deep neural network models. In this case, the train-test procedure is commonly used.

Alternately, a project may have an efficient model and a vast dataset, although may require an estimate of model performance quickly. Again, the train-test split procedure is approached in this situation.

Samples from the original training dataset are split into the two subsets using random selection. This is to ensure that the train and test datasets are representative

of the original dataset.

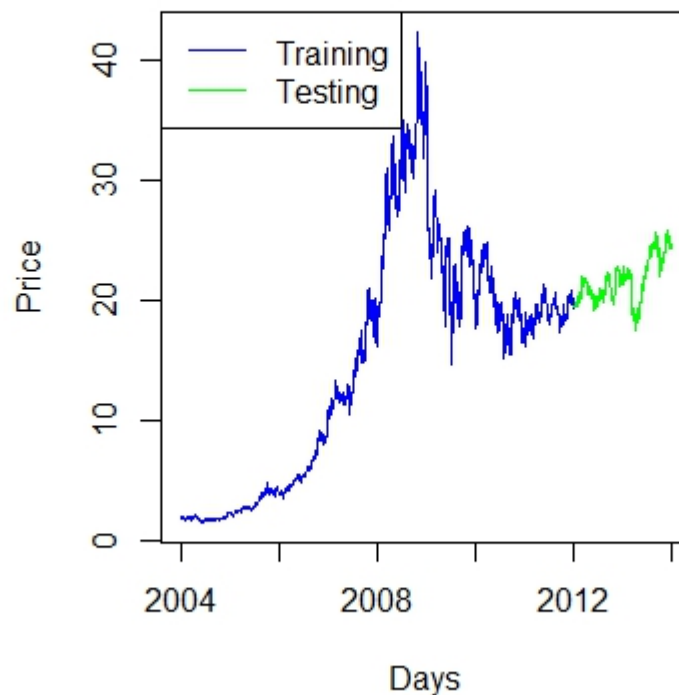
In this project, I delimited the train range as well as for test range as followed

```
# Delimit training range
msft_data.train <- window(msft_data, end = c(2012, 12))
sbux_data.train <- window(sbux_data, end = c(2012, 12))
# Delimit testing range
msft_data.test <- window(msft_data, start = c(2012, 12), end = c(2014, 12))
sbux_data.test <- window(sbux_data, start = c(2012, 12), end = c(2014, 12))
```

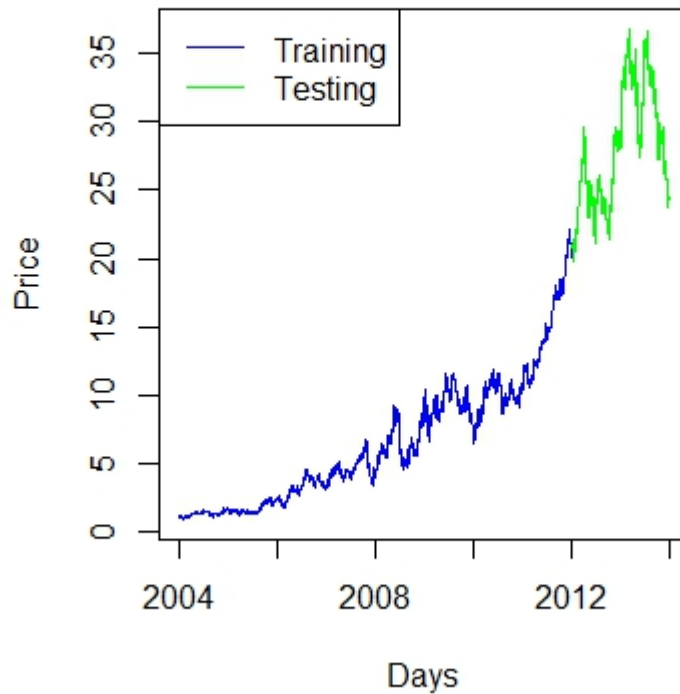
As we can see, I delimited the data in train ,from start to 2012 and test from 2013 till end for more specification and less complication.

To clarify it more, I have plotted them using different colours,

Train and test range of msft Daily Prices 2004-2014



Train and test range of sbux Daily Prices 2004-2014



2.6 ARIMA

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends.

A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods.

An autoregressive integrated moving average model is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The model's goal is to predict future securities or financial market moves by examining the differences between values in the series instead of through actual

values.

An ARIMA model can be understood by outlining each of its components as follows:

Autoregression (AR): refers to a model that shows a changing variable that regresses on its own lagged, or prior, values. Integrated (I): represents the differencing of raw observations to allow the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values). Moving average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

2.6.1 ARIMA Parameters

Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p , d , and q , where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

p : the number of lag observations in the model, also known as the lag order. d : the number of times the raw observations are differenced; also known as the degree of differencing. q : the size of the moving average window, also known as the order of the moving average. For example, a linear regression model includes the number and type of terms. A value of zero (0), which can be used as a parameter, would mean that particular component should not be used in the model. This way, the ARIMA model can be constructed to perform the function of an ARMA model, or even simple AR, I, or MA models.

2.6.2 ARIMA and Stationary data

In an autoregressive integrated moving average model, the data are differenced in order to make it stationary. A model that shows stationarity is one that shows there is constancy to the data over time. Most economic and market data show trends, so the purpose of differencing is to remove any trends or seasonal structures.

Seasonality, or when data show regular and predictable patterns that repeat over a calendar year, could negatively affect the regression model. If a trend appears

and stationarity is not evident, many of the computations throughout the process cannot be made and produce the intended results.

2.6.3 Usage

ARIMA is a method for forecasting or predicting future outcomes based on a historical time series. It is based on the statistical concept of serial correlation, where past data points influence future data points.

2.6.4 How Does ARIMA Forecasting Work?

ARIMA forecasting is achieved by plugging in time series data for the variable of interest. Statistical software will identify the appropriate number of lags or amount of differencing to be applied to the data and check for stationarity. It will then output the results, which are often interpreted similarly to that of a multiple linear regression model.

2.6.5 Bottom Line

The ARIMA model is used as a forecasting tool to predict how something will act in the future based on past performance. It is used in technical analysis to predict an asset's future performance.

ARIMA modeling is generally inadequate for long-term forecastings, such as more than six months ahead, because it uses past data and parameters that are influenced by human thinking. For this reason, it is best used with other technical analysis tools to get a clearer picture of an asset's performance.

2.6.6 Order of ARIMA

Now we understand how these processes work and what their order means, let's see how to find the parameters in practice. In order to find the parameters p , d , and q for respectively the AR, integrated, and MA parts of the model, we can use:

- ACF and PACF plots.

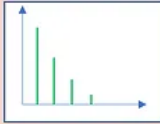

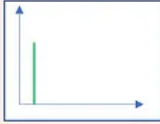
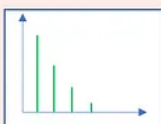
- The various fit of metrics (BIC, AIC) We remind that p is the number of lag observations in the model (also called the lag order), d is the number of times that the raw observations are differenced and q is the size of the moving average window.

Autocorrelation functions

A crucial aspect of a time series process is autocorrelation. Autocorrelation is a statistical property that occurs when a time series is linearly related to a previous or lagged version of itself. It is also used to detect possible seasonality in a time series.

We use the autocorrelation function to assess the degree of dependence in the time series and select an appropriate model (MA, AR, or ARIMA).

The Auto Correlation Function (ACF) and Partial AutoCorrelation Function (PACF) can be computed for any time series (not only stationary). In practice, we use the combination of both of these plots to determine the order of the ARMA process. ARIMA(0, 0, 0) is a white noise model. ($V_t = N_t$) ARIMA(0, 1, 0) is a random walk ($V_t - V_{t-1} = c + N_t$, c is mean) ARIMA(0,1,1) exponential smoothing ($V_t - V_{t-1} = E_t + a_1 * E_{t-1}$)

	ACF	PACF
AR($p=1$)	Exponential decrease 	Sudden fall after lag p 
MA($q=1$)	Sudden fall after lag q 	Exponential decrease 

AIC/BIC criteria

Plotting ACF/PACF is effective for identifying AR and MA processes. But for ARIMA processes, it is more common to use the auto arima functions. Auto arima is a brute-force method that tries different values of p and q while minimizing two criteria: AIC and BIC.

The most common metric to assess the regularized goodness-of-the-fit are:

Bayesian information criterion (BIC) Akaike information criterion (AIC). These metrics provide measures of model performance that account for model complexity. AIC and BIC combine a term reflecting how well the model fits the data with a term that penalizes the model in proportion to its number of parameters [3].

As a regularization technique, we want to penalize based on the number of parameters in the model. Indeed, the larger p and q, the more lags you use to predict the next value, and the more likely you are to overfit your data.

ARIMA using R]

Simpliest way to deal with arima order is **AUTO.ARMA** code uses in R to find out the order.It's not perfect each and every time,but not totally wrong.I have used this in my project,

```
> #to check the order of MA and AR for ARMA
> auto.arima(modified_msft_data)
Series: modified_msft_data
ARIMA(0,0,0) with zero mean
```

```
sigma^2 = 0.1843: log likelihood = -2093.09
AIC=4188.19 AICc=4188.19 BIC=4194.39
> auto.arima(modified_sbux_data)
Series: modified_sbux_data
ARIMA(2,0,0) with non-zero mean
```

```
Coefficients:
      ar1      ar2      mean
    -0.0409 -0.0388  0.0064
s.e.    0.0165  0.0165  0.0042
```

```
sigma^2 = 0.07488: log likelihood = -447.5
AIC=903 AICc=903.01 BIC=927.81
```

The order is (0,0,0) and (2,0,0) here. But it will not give a perfect fit and precised forecast.

2.6.7 ARIMA fit

Though i can see this ARIMA orders aren't good for fitting but still I do fit and get the following result,

```
> # ARIMA Fitting
> model_msft<-Arima(modified_msft_data,order=c(0,0,0))
> print(summary(model_msft))
Series: modified_msft_data
ARIMA(0,0,0) with non-zero mean
```

Coefficients:

```
      mean
      0.0062
s.e.  0.0071
```

```
sigma^2 = 0.1844: log likelihood = -2092.71
AIC=4189.43  AICc=4189.43  BIC=4201.83
```

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	7.494468e-14	0.4293031	0.2506841	-Inf	Inf	0.6273213	-0.02478089

```
> model_sbux<-Arima(modified_sbux_data,order=c(2,0,0))
> print(summary(model_sbux))
Series: modified_sbux_data
ARIMA(2,0,0) with non-zero mean
```

Coefficients:

```
      ar1      ar2      mean
      -0.0409  -0.0388  0.0064
s.e.   0.0165   0.0165  0.0042
```

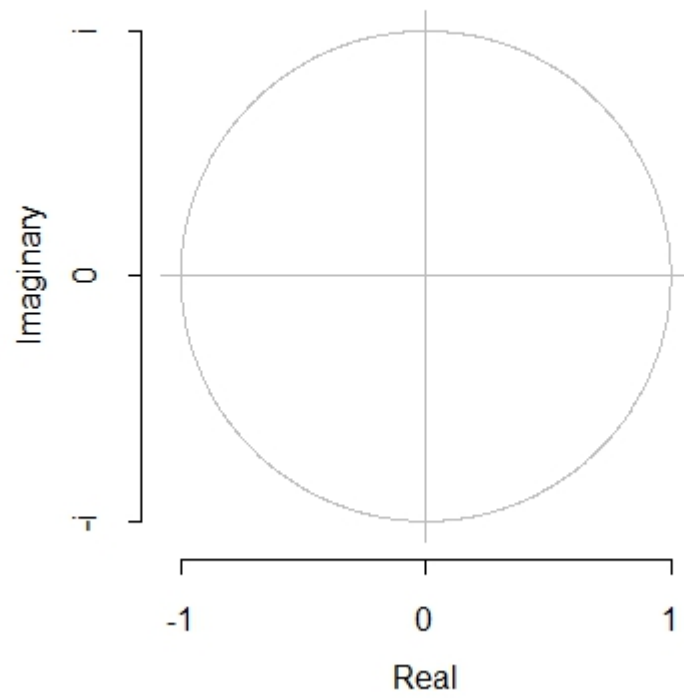
```
sigma^2 = 0.07488: log likelihood = -447.5
AIC=903  AICc=903.01  BIC=927.81
```

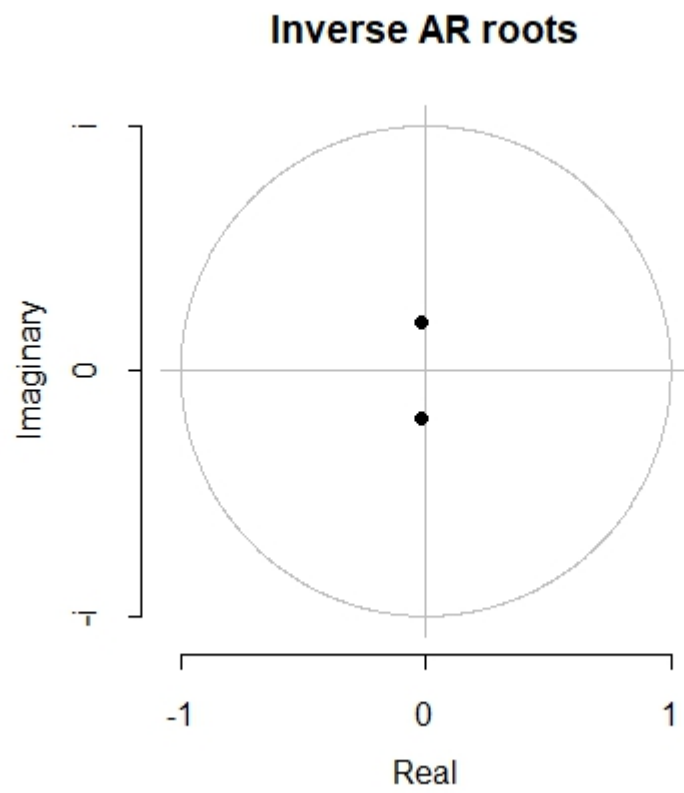
Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	1.963698e-06	0.273532	0.1668277	NaN	Inf	0.6824005	0.0003364087

After plotting the ARIMA model we get these following plot,

No AR or MA roots

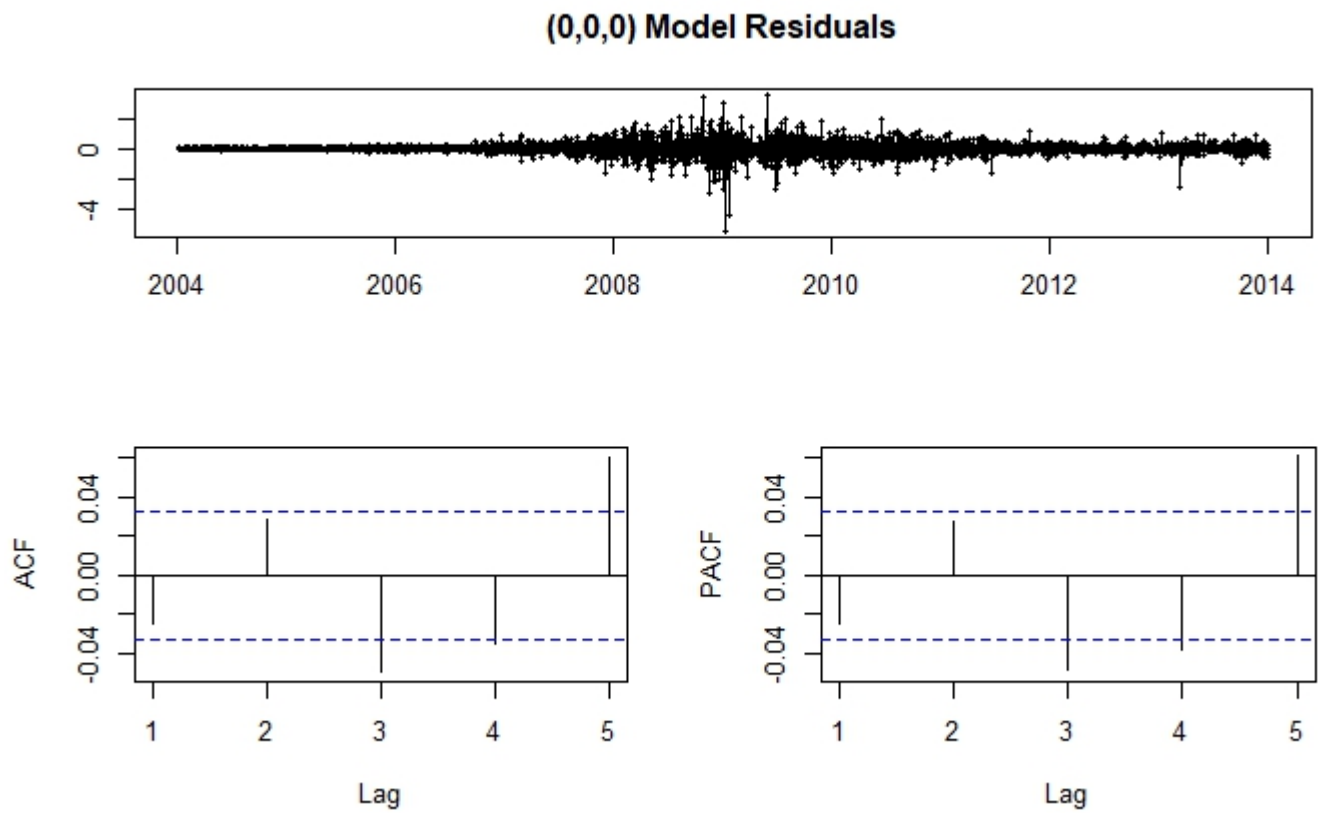


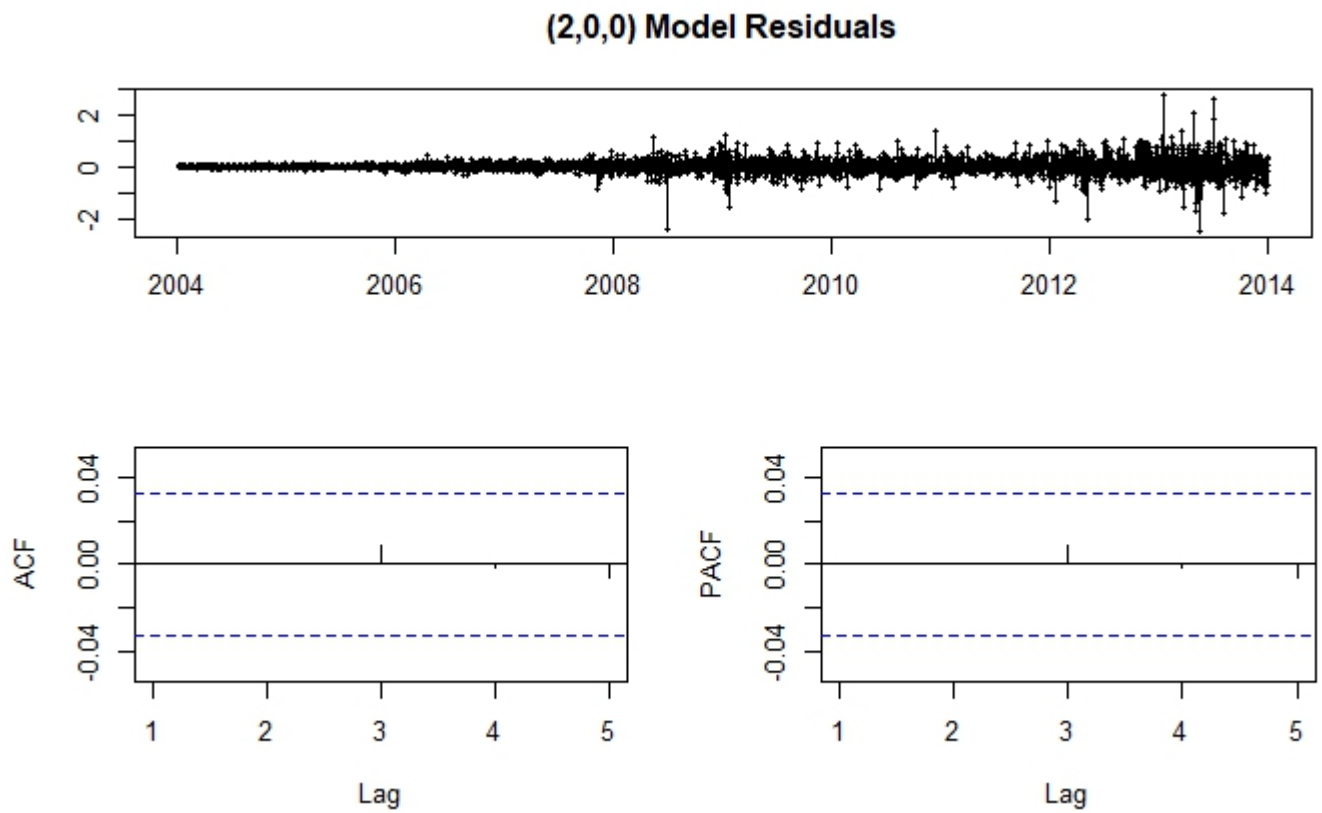


These plots clearly says, the arima model isn't perfect.

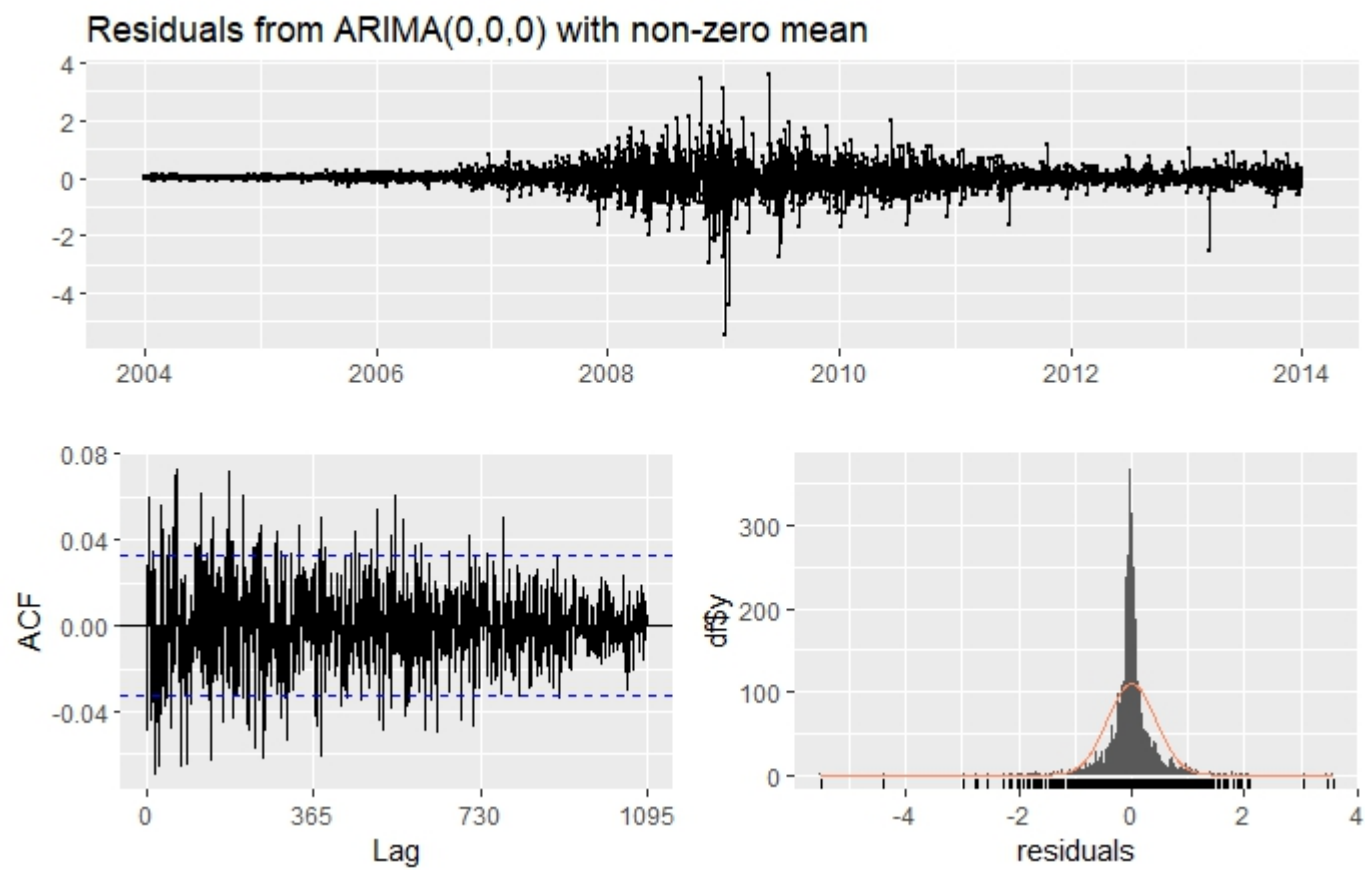
2.7 Checking Residuals

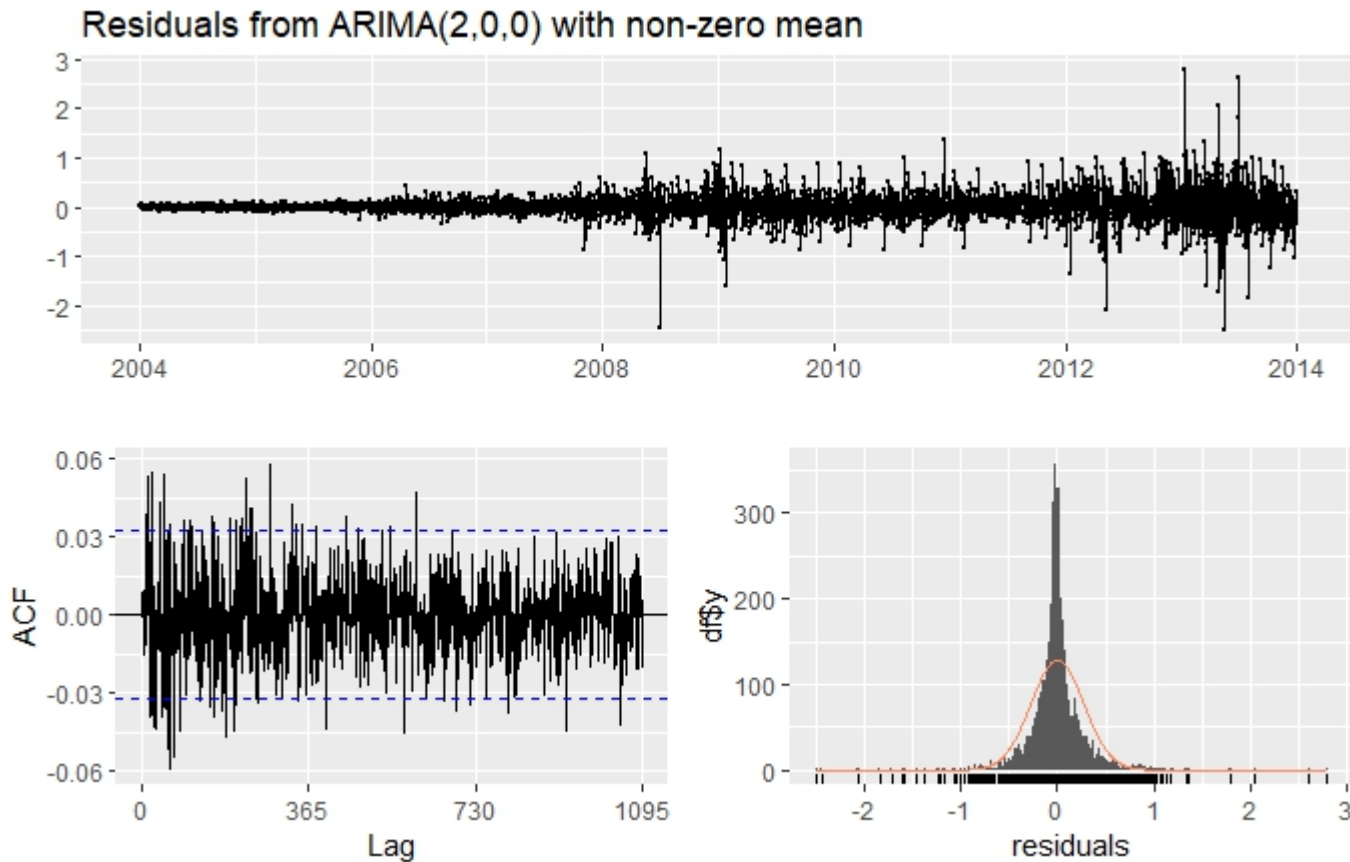
To check the residuals we use this plots,





This plot is given by using TSdisplay method to show the details. Now, the summary of this residual check is,





and

```
> #checking residual
> checkresiduals(model_msft)
```

Ljung-Box test

data: Residuals from ARIMA(0,0,0) with non-zero mean
 $Q^* = 1507.7$, $df = 730$, $p\text{-value} < 2.2e-16$

Model df: 0. Total lags used: 730

```
> checkresiduals(model_sbux)
```

Ljung-Box test

data: Residuals from ARIMA(2,0,0) with non-zero mean
 $Q^* = 938.25$, $df = 728$, $p\text{-value} = 1.976e-07$

Model df: 2. Total lags used: 730

2.8 Forecast Using Holt-Winters

Holt-Winters is a model of time series behavior. Forecasting always requires a model, and Holt-Winters is a way to model three aspects of the time series: a typical value (average), a slope (trend) over time, and a cyclical repeating pattern (seasonality).

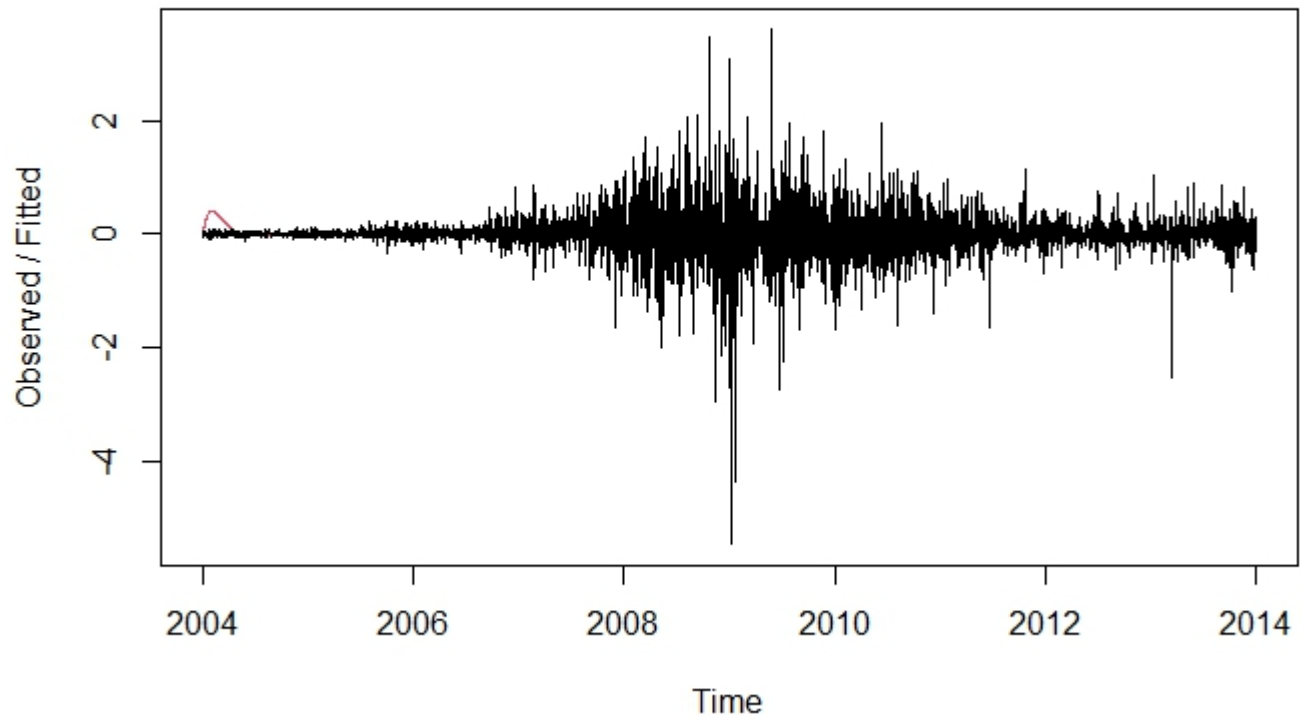
Time series anomaly detection is a complicated problem with plenty of practical methods. It's easy to get lost in all of the topics it encompasses. Learning them is certainly an issue, but implementing them is often more complicated. A key element of anomaly detection is forecasting—taking what you know about a time series, either based on a model or its history, and making decisions about values that arrive later.

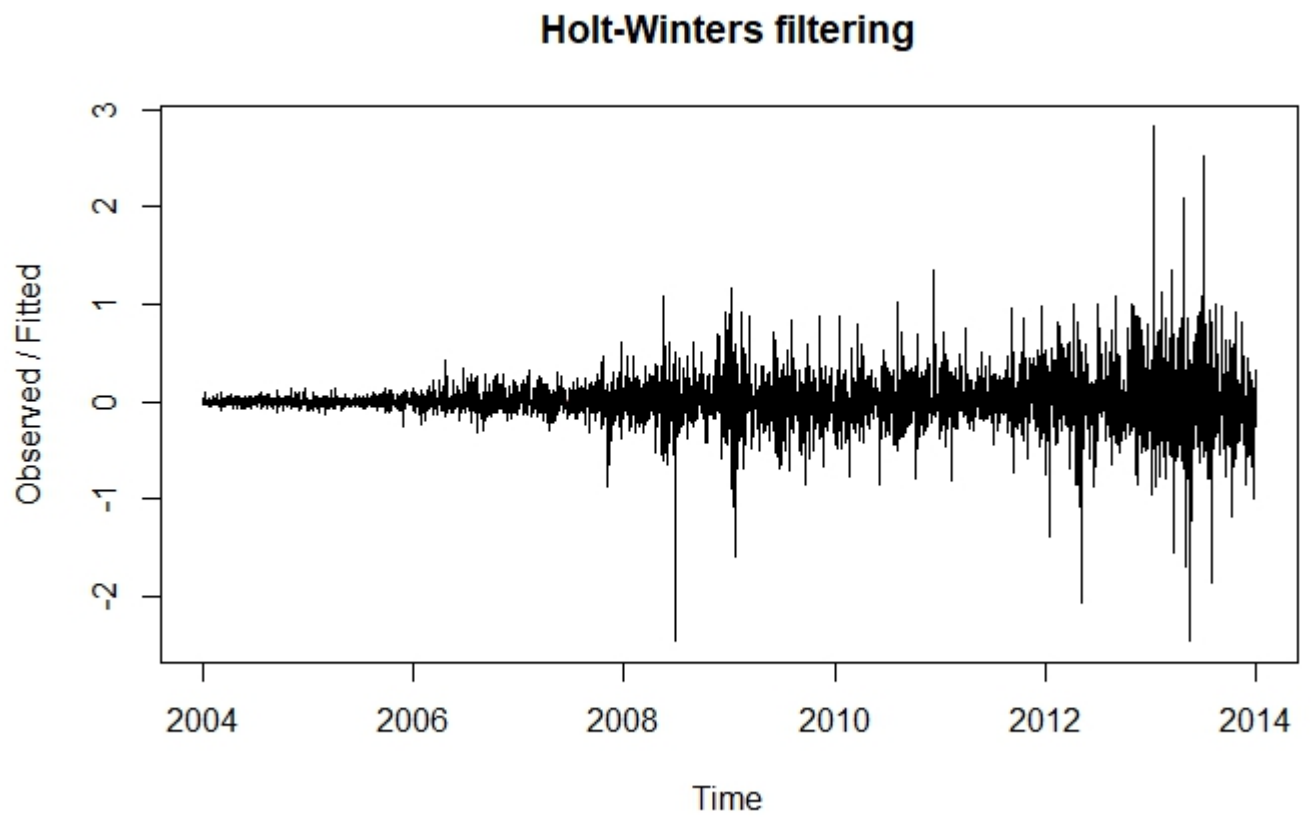
You know how to do this already. Imagine someone asked you to forecast the prices for a certain stock, or the local temperature over the next few days. You could draw out your prediction, and chances are it's a pretty good one. Your brain works amazingly well for problems like this, and our challenge is to try to get computers to do the same.

If you take an introductory course on time series, you'll learn how to forecast by fitting a model to some sample data, and then using the model to predict future values. In practice, especially when monitoring systems, this approach doesn't work well, if at all! Real systems rarely fit mathematical models. There's an alternative. You can do something a lot simpler with exponential smoothing.

As the ARIMA model is not perfect enough, I use Holt-Winters model. Now the filtering plot is,

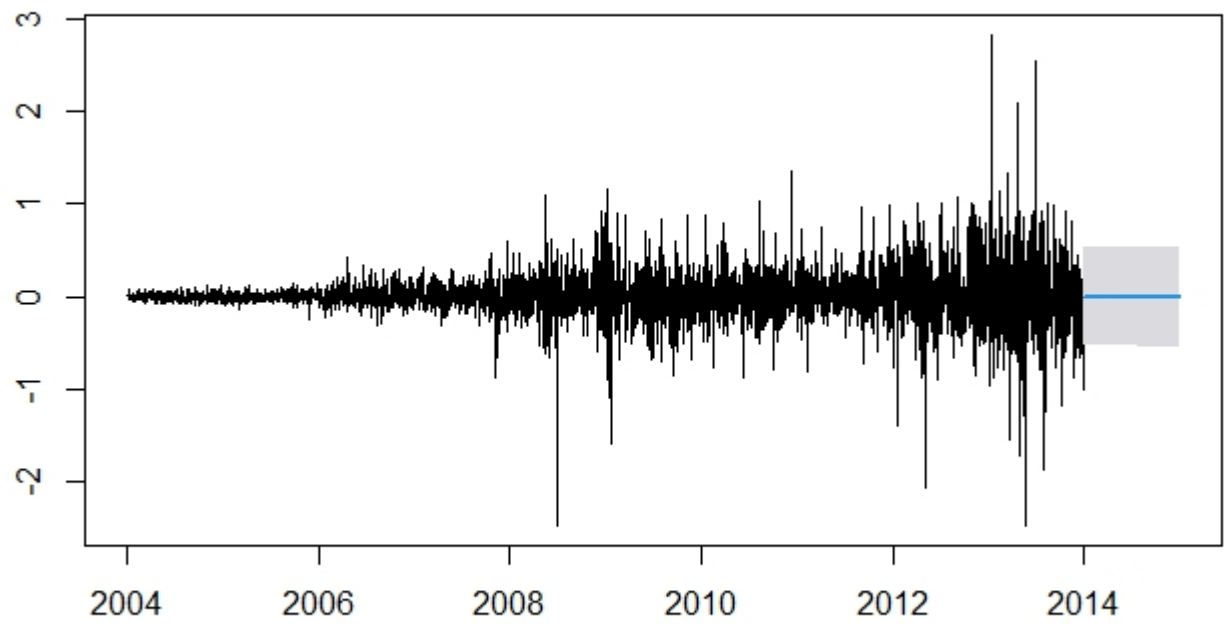
Holt-Winters filtering



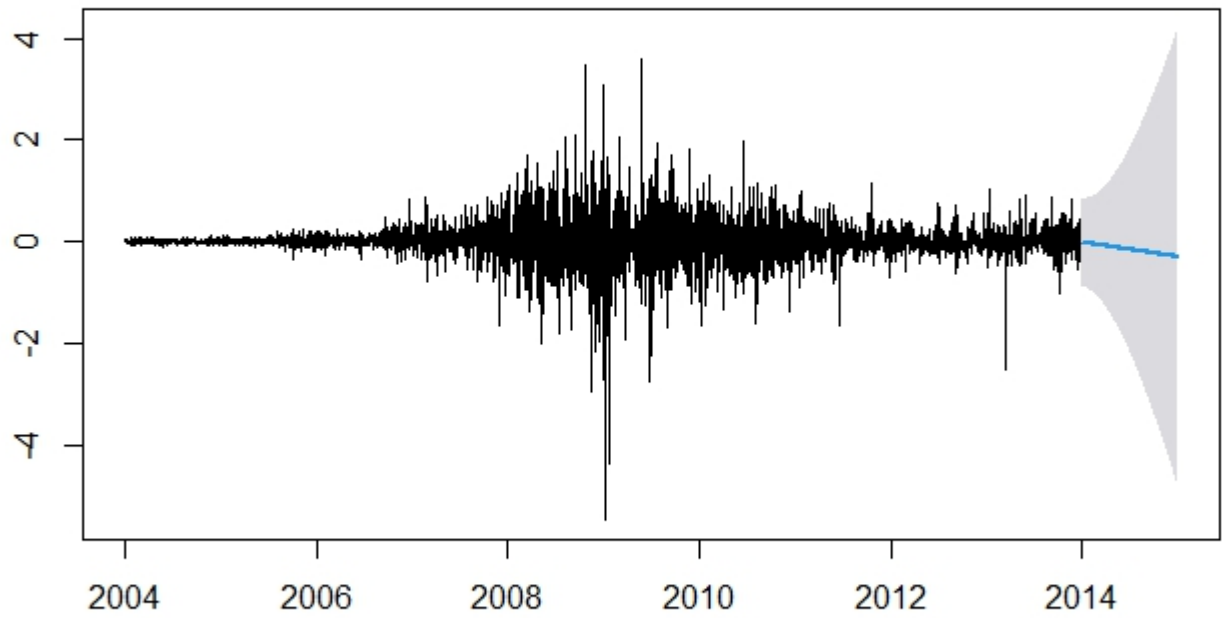


The forecasted figure using holt-winters are,

Forecasted Value of 365 steps ahead sbux price



Forecasted Value of 365 steps ahead msft price



2.9 Forecast Using ETS Method

Estimating ETS models

An alternative to estimating the parameters by minimising the sum of squared errors is to maximise the “likelihood”. The likelihood is the probability of the data arising from the specified model. Thus, a large likelihood is associated with a good model. For an additive error model, maximising the likelihood (assuming normally distributed errors) gives the same results as minimising the sum of squared errors. However, different results will be obtained for multiplicative error models. In this section, we will estimate the smoothing parameters α , β , γ and ϕ , and the initial states ℓ_0 , b_0 , s_0 , s_{-1}, \dots, s_{-m+1} , by maximising the likelihood.

The possible values that the smoothing parameters can take are restricted. Traditionally, the parameters have been constrained to lie between 0 and 1 so that the equations can be interpreted as weighted averages. That is, $0 < \alpha, \beta^*, \gamma^*, \phi < 1$. For the state space models, we have set $\beta = \alpha\beta^*$ and $\gamma = (1 - \alpha)\gamma^*$. Therefore, the traditional restrictions translate to $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$. In practice, the damping parameter ϕ is usually constrained further to prevent numerical difficulties in estimating the model. In R, it is restricted so that $0.8 < \phi < 0.98$.

Another way to view the parameters is through a consideration of the mathematical properties of the state space models. The parameters are constrained in order to prevent observations in the distant past having a continuing effect on current forecasts. This leads to some *admissibility* constraints on the parameters, which are usually (but not always) less restrictive than the traditional constraints region (Hyndman et al., 2008, p. Ch10). For example, for the ETS(A,N,N) model, the traditional parameter region is $0 < \alpha < 1$ but the admissible region is $0 < \alpha < 2$. For the ETS(A,A,N) model, the traditional parameter region is $0 < \alpha < 1$ and $0 < \beta < \alpha$ but the admissible region is $0 < \alpha < 2$ and $0 < \beta < 4 - 2\alpha$.

Model selection

A great advantage of the ETS statistical framework is that information criteria can be used for model selection. The AIC, AIC_c and BIC, introduced in Section 5.5, can be used here to determine which of the ETS models is most appropriate for a given time series.

For ETS models, Akaike's Information Criterion (AIC) is defined as

$$AIC = -2\log(L) + 2k,$$

where L is the likelihood of the model and k is the total number of parameters and initial states that have been estimated (including the residual variance).

The AIC corrected for small sample bias (AIC_c) is defined as

$$AIC_c = AIC + \frac{2k(k+1)}{T-k-1},$$

and the Bayesian Information Criterion (BIC) is

$$BIC = AIC + k[\log(T) - 2].$$

Three of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties. Specifically, the models that can cause such instabilities are $ETS(A,N,M)$, $ETS(A,A,M)$, and $ETS(A,A_d,M)$, due to division by values potentially close to zero in the state equations. We normally do not consider these particular combinations when selecting a model.

Models with multiplicative errors are useful when the data are strictly positive, but are not numerically stable when the data contain zeros or negative values. Therefore, multiplicative error models will not be considered if the time series is not strictly positive. In that case, only the six fully additive models will be applied.

2.9.1 Forecasting Using ETS

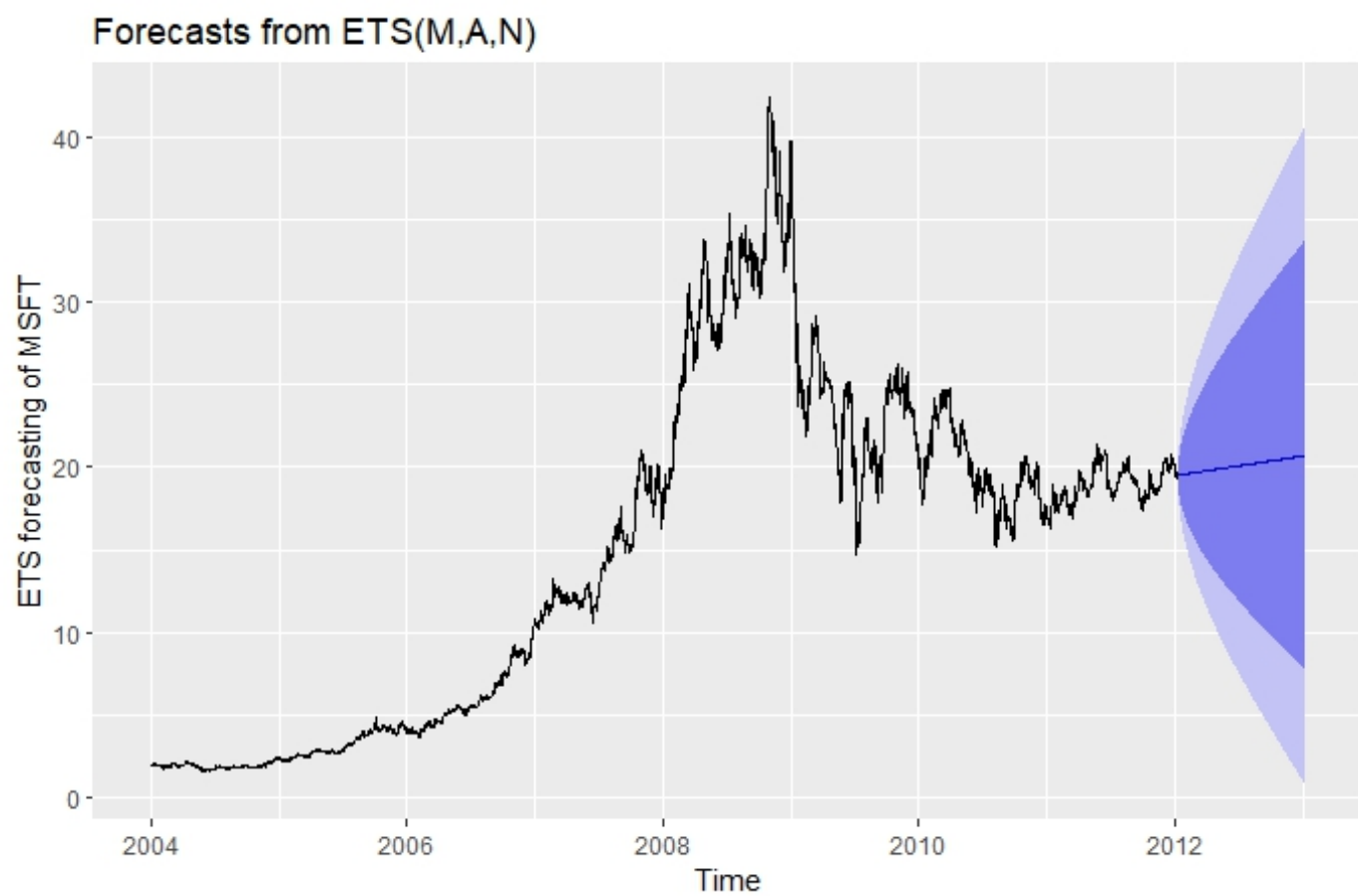
Point forecasts are obtained from the models by iterating the equations for $t = T + 1, \dots, T + h$ and setting all $\varepsilon_t = 0$ for $t > T$.

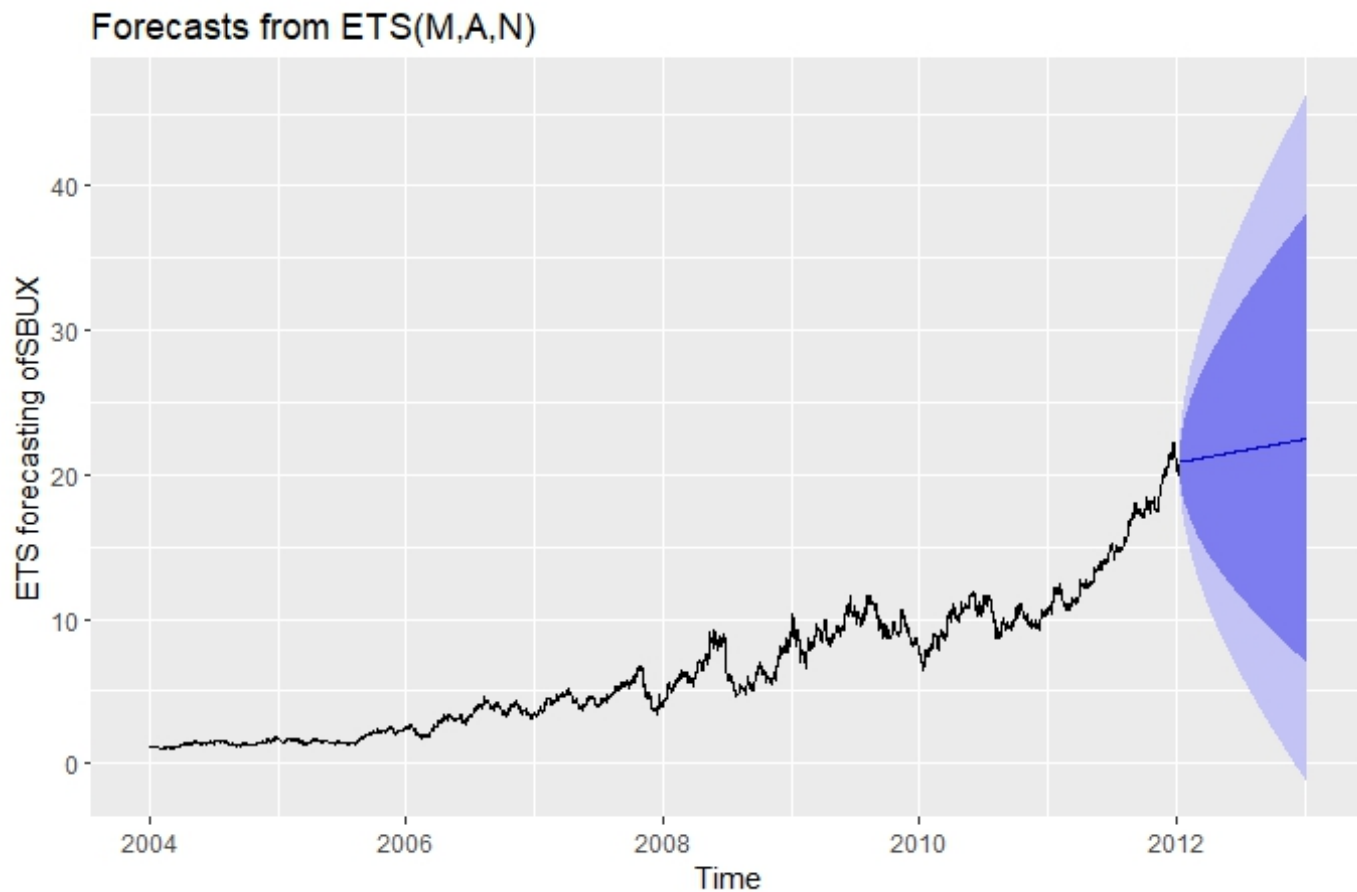
For example, for model ETS(M,A,N), $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$. Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$. Similarly,

$$\begin{aligned} y_{T+2} &= (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2}) \\ &= [(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}](1 + \varepsilon_{T+2}). \end{aligned}$$

Therefore, $\hat{y}_{T+2|T} = \ell_T + 2b_T$, and so on. These forecasts are identical to the forecasts from Holt's linear method, and also to those from model ETS(A,A,N). Thus, the point forecasts obtained from the method and from the two models that underlie the method are identical (assuming that the same parameter values are used).

ETS point forecasts are equal to the medians of the forecast distributions. For models with only additive components, the forecast distributions are normal, so the medians and means are equal. For ETS models with multiplicative errors, or with multiplicative seasonality, the point forecasts will not be equal to the means of the forecast distributions.





2.10 Checking Accuracy

```
> # Calculate the accuracy of the forecast
> accuracy(forecast1, msft_data.test)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	4.355796e-06	0.4643925	0.2720269	-0.009397272	1.715082	0.04364105	-0.003579048
Test set	9.878261e-01	1.8059695	1.4728540	4.028129735	6.591139	0.23628876	0.986668573

Theil's U

Training set	NA
Test set	7.220567

```
> accuracy(forecast2, sbux_data.test)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.003534099	0.1979754	0.1250133	-0.04485236	2.113309	0.05946217	-0.05732013
Test set	5.666377107	6.8332531	5.7009953	18.52776697	18.694318	2.71166102	0.98955880

Theil's U

Training set	NA
Test set	13.15614

2.11 Further Test

Here the process should be follows step by step to decode the final solution. without omitting the errors or checking the seasonality we cant fit the model. After fitting the values, for final assumption we need to check the curve pattern and for forecast we need to put it as time series model and for prediction we need to delimit the training and testing range. Also after checking for stationarity we have to change the data to stationary one (by taking difference of various order) and then forecast 'n' ahead forecasts.

Chapter 3

Interpretation

3.1 Stationarity

- From the **Stationarity Checking**, I can interpretate that all the data is not stationary for both **Microsoft Starbucks**.
- I get the same interpretation from **ADF test** but it is more effective cause the interpretation is from a test.
- Hence after taking difference format of this data, I got the final stationary data set and used ARIMA at first then HoltWinters model and lastly, ETS method to precise the forecasted value.
- The statistical forecasting using ARIMA, Holt-Winters, and ETS methods on real data from 2004-2014 suggests that MSFT share prices will continue to rise, albeit with occasional fluctuations, while SBUX share prices will remain relatively stable with a slight decline towards the end of the time period. However, it's important to note that forecasting share prices is inherently uncertain, and other factors beyond the scope of statistical modeling can also influence share prices.

Chapter 4

Conclusion

The result analysis is already done in the Interpretation part. The final conclusion is all about the following,

- Both MSFT and SBUX is increasing after the end point. But, Microsoft is not overtaking the loss. Hence, it is better to invest in SBUX than MSFT cause Starbucks is crossing the loss and make some profit.
- We can conclude that any company from any code can be analyzed by this code. Accuracy is not that precise but the value is workable.

Appendices

minted

Code For This Project

```
#begin
suppressPackageStartupMessages(library(IntroCompFinR))#The package is for the intro
suppressPackageStartupMessages(library(xts))
suppressPackageStartupMessages(library(methods))
suppressMessages(library(forecast))
suppressMessages(library(tseries))
suppressMessages(library(quantmod))
suppressMessages(library(caret))
suppressMessages(library(nnet))
library(PerformanceAnalytics)
library(tseries)
library(forecast)

data(msftDailyPrices, sbuxDailyPrices)

msft_data<-ts(msftDailyPrices,start="2004",end="2014",frequency = 365)
sbux_data<-ts(sbuxDailyPrices,start = "2004", end= "2014", frequency = 365)

# Stationarity Checking

#plot the non-modified data
plot(msft_data,main="MSFT DATA PLOT")
```

```

plot(sbx_data,main="SBUX DATA PLOT")

#adf test
adf.test(msft_data)
adf.test(sbx_data)

#difference
modified_msft_data<- diff(msft_data)
modified_sbx_data<- diff(sbx_data)

#plot the modified data
plot(modified_msft_data,main="modified msft data")
plot(modified_sbx_data,main="modified sbux data")

#adf test of modified data
adf.test(modified_msft_data)# p-value <0.05, series is stationary.
adf.test(modified_sbx_data)#p value is < 0.05 , series is stationary.

#acf and pacf of modified data
acf(modified_msft_data,lag.max = 10,main= "acf of modified_msft_data")
acf(modified_sbx_data,lag.max = 10,main="acf of modified_sbx_data")
pacf(modified_msft_data,lag.max=10,main = "pacf of modified_msft_data")
pacf(modified_sbx_data,lag.max = 10,main="pacf of modified_sbx_data")

# Delimit training range
msft_data.train <- window(msft_data, end = c(2012, 12))
sbx_data.train <- window(sbx_data, end = c(2012, 12))
# Delimit testing range
msft_data.test <- window(msft_data,start = c(2012, 12), end = c(2014, 12))
sbx_data.test <- window(sbx_data,start = c(2012, 12), end = c(2014, 12))
# Training and testing ranges chart
plot(msft_data,main="Train and test range of msft Daily Prices 2004-2014",ylab="Price")
lines(msft_data.train,col="blue")
lines(msft_data.test,col="green")

```

```

legend("topleft",col=c("blue","green"),lty=1,legend=c("Training","Testing"))

plot(sbox_data,main="Train and test range of sbux Daily Prices 2004-2014",ylab="Price")
lines(sbox_data.train,col="blue")
lines(sbox_data.test,col="green")
legend("topleft",col=c("blue","green"),lty=1,legend=c("Training","Testing"))

#to check the order of MA and AR for ARMA
auto.arima(modified_msft_data)
auto.arima(modified_sbux_data)

# ARIMA Fitting
model_msft<-Arima(modified_msft_data,order=c(0,0,0))
print(summary(model_msft))
plot(model_msft)
model_sbux<-Arima(modified_sbux_data,order=c(2,0,0))
print(summary(model_sbux))
plot(model_sbux)

#display residuals
tsdisplay(residuals(model_msft), lag.max=5, main='(0,0,0) Model Residuals')
tsdisplay(residuals(model_sbux), lag.max=5, main='(2,0,0) Model Residuals')

#checking residual
checkresiduals(model_msft)
checkresiduals(model_sbux)

#holtwinters
model1 = HoltWinters(modified_msft_data, gamma = FALSE)
pred1 = predict(model1, n.ahead = 365)
plot(model1)

```

```

f1<-forecast(model1, level=c(95), h=365)
plot(f1,main="Forecasted Value of 365 steps ahead msft price")

model2 = HoltWinters(modified_sbux_data, gamma = FALSE)
pred2 = predict(model2, n.ahead = 365)
plot(model2)
f2<-forecast(model2, level=c(95), h=365)
plot(f2,main="Forecasted Value of 365 steps ahead sbux price")

##Forecast using ETS and accuracy test
fit1 <- ets(msft_data.train)
fit2 <- ets(sbux_data.train)

forecast1 <- forecast(fit1, h = length(msft_data.test))
forecast2 <- forecast(fit2, h = length(sbux_data.test))
fit1 %>% forecast(h=365) %>%
  autoplot() +
  ylab("ETS forecasting of MSFT")
fit2 %>% forecast(h=365) %>%
  autoplot() +
  ylab("ETS forecasting of SBUX")

# Calculate the accuracy of the forecast
accuracy(forecast1, msft_data.test)
accuracy(forecast2, sbux_data.test)

```


Chapter 5

Real Life Data

As I took the data to december of 2014, I can check the forecasted data fro next 365 days or any date asked for. Hence, fore accuracy checking , I will present those datas.

