TIME SERIES ANALYSIS AND FORECASTING OF GOOGLE AND MICROSOFT STOCKS

Dhaval Thakur, Rushi Bhuva, Tejas Pandit

Introduction

Despite its prevalence, Stock Market prediction remains a secretive and empirical art. Few people, if any, are willing to share what successful strategies they have. A major goal of this project is to add to the academic understanding of stock market prediction. In this project we have tried to implement statistical technique namely Time Series Analysis (TSA) in order to understand stock trend and furthermore to forecast the stock prices for next 24 months of two firms. In this project we would consider monthly stocks records for the companies, Google, and Microsoft (MSFT). We would be trying to build a statistical Time Series (TS) model that can forecast estimated stock prices in the future so that the stakeholders such as stock traders, company officials can have better insight on the stock prices of these two company using appropriate statistical methods and furthermore we build a comprehensive dashboard which gives quick insights on important aspects of our data analysis using Tableau.

Data Summary

In this project we would be using datasets of stock prices of Microsoft and Google from S&P 500 stock dataset provided by Kaggle. We have data of stock price from year 1986 to 2017(till November) for Microsoft and from year 2004 to 2017 for Google. Google and Microsoft both have **3239 number of observations**. Moreover, upon further observation, both the datasets have no NULL or missing values. Both the datasets have same 6 variables (with types) namely - Open, Close, High, low, Volume (**each continuous datatype**) & date (**interval**).
Now in our analysis, we would be taking the "closing" stock price and would be removing other variables from both the dataset (i.e. Open, High, low & volume). Logic behind the decision is that investors, traders, financial institutions, regulators and other stakeholders use it as a reference point for determining performance over a specific time such as one year, a week and over a shorter time frame such as one minute or less. In fact, investors and other stakeholders base their decisions on closing stock prices.

Transformation of Data

Now in order to simplify our analysis, we averaged daily close stock price and make our dataset corresponding to monthly stock averaged price. Thus, now our new datasets for Google and Microsoft has three variables – Year (interval type), month (interval type) & Average Closed (continuous) Values, with **155 number of observations each**. Furthermore, we have filtered our dataset from 2005 to 2017(till November), due to the fact that upon EDA of our datasets we found that the trend is distinguishable and comparable from year 2005.
Now to start the analysis, we would have to do Time series Analysis as we would be doing our analysis on an interval data in order to come up with some statistical results.

About Time Series

Time Series is a sequence of the observations recorded at regular time interval. According to frequency of the observations taken it can be divided in various types such as hourly, daily, weekly, monthly, quarterly, and annual. Note that we already have converted our dataset which has observations segregated on monthly basis.

Time Series Data Analysis

Primarily time series analysis is used for Descriptive, Forecasting, Intervention analysis & Quality control.

1. Descriptive: Identify patterns in correlated data—trends and seasonal variation.
2. Explanation: understanding and modelling the data.
3. Forecasting: prediction of short-term trends from previous patterns.

Plotting time series data is an important first step in analysing their various components, thus we converted the data-frame to a time series object using ***ts()*** found in R using ***tseries library***. Now we used ***ts_plot()*** of ***TSStudio library*** to plot time plot to visualize the data in time series format. The reason behind using TSstudio is that it provides a set of tools for time series analysis and forecasting applications, using (mainly) models from the forecast package and visualization tools based on the ***plotly package***.

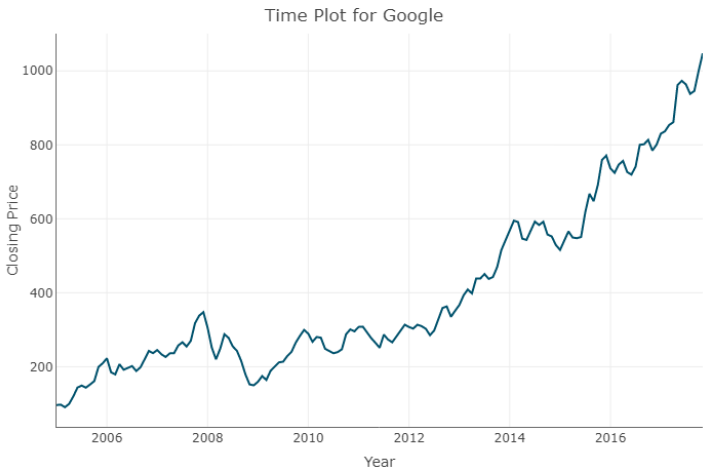

Fig.1

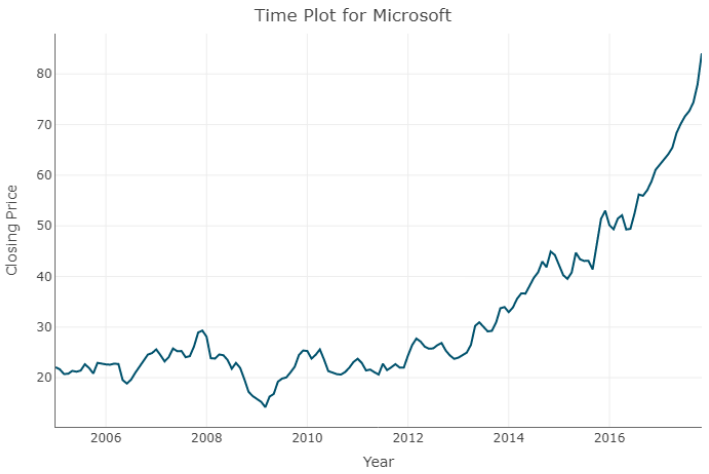

Fig.2

Observation: It can be observed from the plot that the price of closing stock has increased from the year 2005-2017 for both datasets in consideration which suggests that that the average price of the stock has increased in this period for both companies.

Decomposition of the Time Series

One of the main objectives for a decomposition is to estimate seasonal effects that can be used to create and present seasonally adjusted values. The decomposition reduces a time series into 3 components: trend, seasonal patterns, and random (residuals after seasonal and trend series are removed). In turn, we aim to model the random errors as some form of stationary process.

A common approach to modelling time-series data (Y) in which it is assumed that the four components of a time series; trend component (T), seasonal component (S), random component (R), are added to form the values of the time series at each time period.
In an additive model the time series is expressed as:

Time Series = Trend + Seasonal + Random.

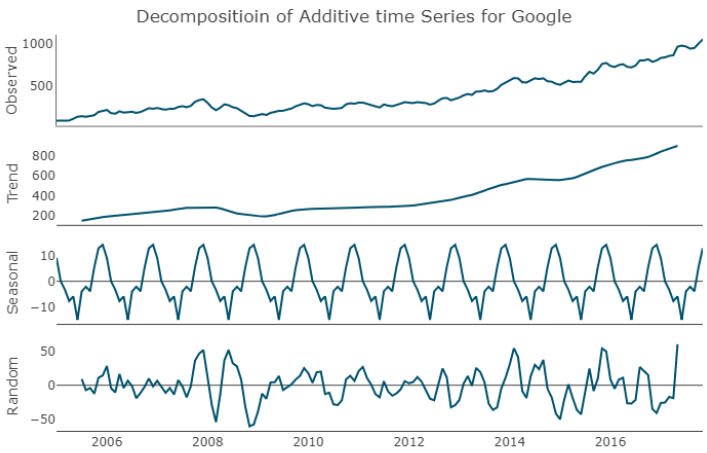

Fig.3

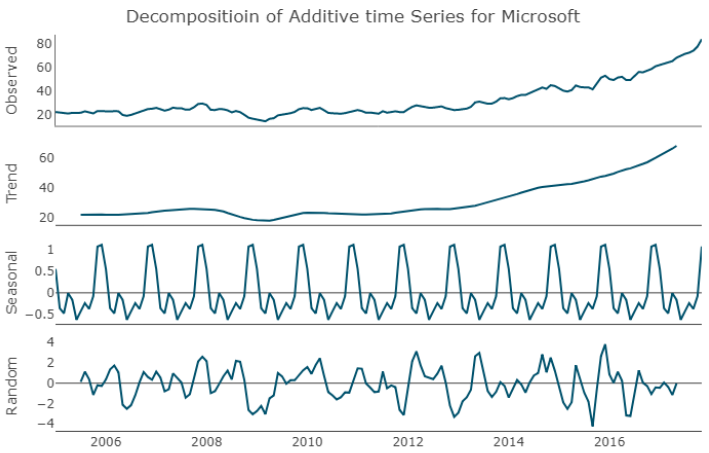

Fig.4

Observation from the decomposition plots: From the decomposed plot we observe that the trend is increasing throughout our period of consideration, also a seasonality (Yearly) can be observed from the Seasonal plot, and lastly there is some randomness present also in our time series plot for both of the companies' time plots.

Check for Stationarity

Setting up Hypothesis

H_0: A unit root is present in a time series sample
H_1: The time series is stationary (or trend-stationary)

Stationarity means that the statistical properties of a time series (mean and variance) do not change over time. Stationarity is important because many useful analytical tools and statistical tests and models such as ARIMA model rely on it.
Now we used visualization technique and non-parametric test such as Dickey-Fuller Test to test for stationarity.

Visualization of Time Series
We plotted our time series object using ts_plot() in Fig.1 and observed that our dataset for both the company's stock are non-stationary, and the increase and means seems to be uneven.
Furthermore, to support this observation we used non-parametric test called Dickey-Fuller Test on our both datasets.

Dickey-Fuller Test
Dickey–Fuller test tests the null hypothesis that a unit root is present in an autoregressive model. The alternative hypothesis is different depending on which version of the test is used but is usually stationarity or trend-stationarity.
Test Results: We ran this test for both Google and Microsoft time series object, and got **Dickey-Fuller coefficient = -0.26166** and **p-value = 0.99** which is greater than 0 for Google's stock price dataset and we find similar results for the Microsoft's dataset with **Dickey-Fuller coefficient = 1.4961** and **p-value = 0.99** , thus we reject our Null Hypothesis and accept the Alternative Hypothesis that our both the **time series are non-stationary**.
We cannot build **ARIMA** model on a non-stationary thus we have to convert our both time series objects to stationary so that we can fit our model for further statistical results such as forecasting.

Transforming the data into Stationary State
In order to transform our time series objects, we must perform two tasks. To maintain the constant variance, we need to do the log transformation in each time series and to maintain constant mean we need to do differencing (This is the commonly used technique to remove non-stationarity). Here, we get stationary ts for **difference = 1**. Thus, after above transformations, we visualized the transformed data as well as ran Dickey-Fuller and got **p-value = 0.01** for both time series objects and thus giving us confirmation that our transformed ts objects are now stationary. (Refer Figure [3] and [4] in Appendix)

Finding Optimal Parameters for ARIMA Model
In order to build a model like ARIMA model, we require hyper parameters p, q, and d. These are the hyper parameters which are required in order to build and fit an ARIMA Model.
p — the number of autoregressive
d — degree of differencing
q — the number of moving average terms
m — refers to the number of periods in each season
(P, D, Q)— represents the (p, d, q) for the seasonal part of the time series

ACF Plot
ACF is the plot used to see the correlation between the points, up to and including the lag unit. In ACF, the correlation coefficient is in the y-axis whereas the number of lags is shown in the x-axis.
Observation: If there is a Positive autocorrelation at lag 1 then we use the AR model, if there is a Negative autocorrelation at lag 1 then we use the MA.
As can be observed from Fig.5 and Fig.6, the decay of ACF chart is very slow, which means that the population is not stationary.

PACF Plot
A partial autocorrelation is the amount of correlation between a variable and a lag of itself that is not explained by correlations at all lower-order-lags.

Observation: If the PACF plot drops off at lag n, then use an AR(n) model and if the drop in PACF is more gradual then we use the MA term. Here as observed from Fig.5 and Fig.6, there is correlation of 1(significant spike) for lag 1.
If the data are from an ARIMA(p,d,0) or ARIMA(0,d,q) model, then the ACF and PACF plots can be helpful in determining the value of p or q 16 If p and q are both positive, then the plots do not help in finding suitable values of p and q.[7]

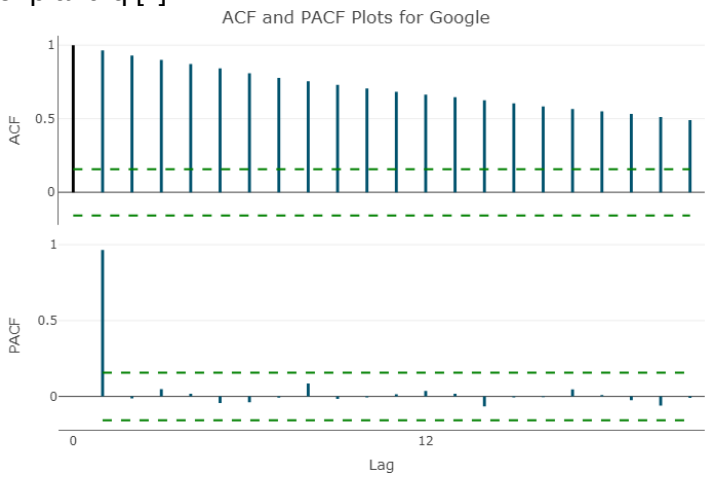

Fig.5

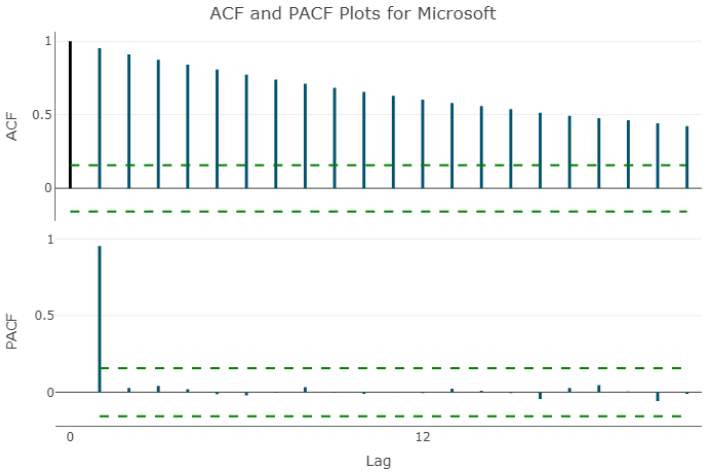

Fig.6

Build the ARIMA Model for Google and Microsoft

Now to obtain the optimal values for hyper parameters p, q and d we would be Using the auto.arima function from the forecast R package to fit the best model and coefficients, given the default parameters including seasonality as TRUE.
Observations: After running auto.arima, for Google the ARIMA (1,0,1) (0,1,1) [12] model parameters are lag 0 differencing (d), an autoregressive term of first lag (p) and a moving average model of order 1 (q). Then the seasonal model has an autoregressive term of first lag (D) and a moving average model of order 1 (q) at model period 12 units, in this case months.
On other hand, for Microsoft, the ARIMA (0,1,1) (2,1,0) [12] model parameters p is 0, d =1 and q =1 and for seasonal model p is 2, d is 1 and q is 0.
The check_res() from TSstudio performs model diagnostics of the residuals and the ACF will include an autocovariance plot.

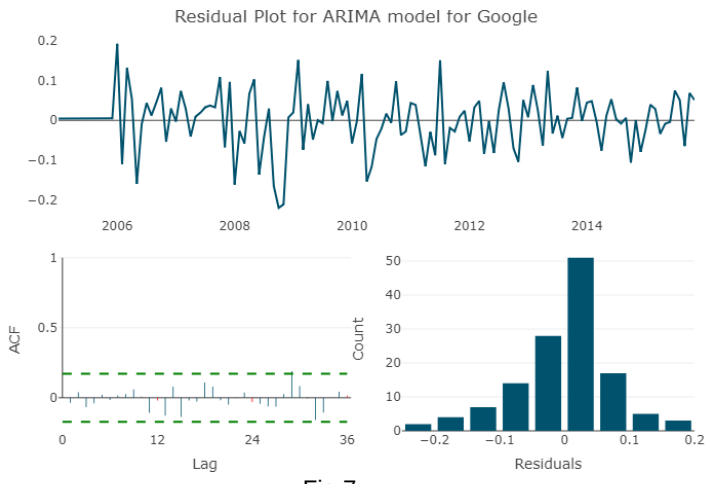

Fig.7

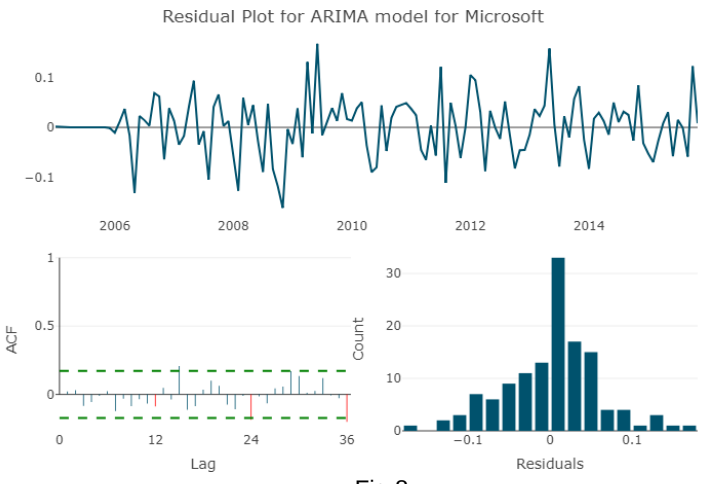

Fig.8

Observations: The residual plots appear to be centred around 0 as noise, with no pattern. Both the models have normally distributed residuals. The autocorrelation plot shows that for all sample autocorrelations except for that at lag 29 fall inside the 95 % confidence bounds indicating the residuals appear to be random. Thus, the ARIMA model is a good fit for the time series of Google. The same can be depicted from the residuals of Microsoft[Fig.8]. The autocorrelation plot shows that for all sample autocorrelations except for those at lag 15,

24 and 36 falls inside the 95% confidence interval indicating the residuals appear to be random. Thus, the ARIMA model is a good fit for the time series of Microsoft as well.
Now, before we start validity testing or forecasting, it is noted that we transformed our data (log transformation) to make it stationary. Forecasted value has a list of forecasted value and confidence interval. and thus, we would do the conversion by performing this step: $2.718^{forecasted\_value}$

Hypothesis Creation for Forecasting for both firms

Now let us setup hypothesis for our Time-series forecasting. [1]
H_0: There is no increase/decrease of stock prices over period of 24 months of time for the firm.
H_1: There is gradual increase/ decrease of stock prices over period of 24 months of time for the firm.

Validating the Model

Though we have built a good model, but it is still left to test the feasibility of our model. Thus, in order to test the accuracy of our model we would perform these steps:

- Split the original data sets of each company into 2(Train dataset, test dataset (24 months)).
- Fit the respective ARIMA models to their respective train datasets.
- Compare the predicted values and the original values for each company's stocks.

Observations: For both companies Google and Microsoft we observed that the predictions are very close to the original values. We would be using Mean Absolute Percentage Error (MAPE) and Mean Percentage Error (MPE), which is a measure of prediction accuracy of a forecasting method in statistics. We got **MAPE** value of **4.677%** and **4.85%** for Google and Microsoft ARIMA model on our test data. Furthermore, we got **MPE** value of **1.685%** and **2.488%** for Google and Microsoft ARIMA model on our test data. Thus, our both models are Justifiable.

Stock Forecasting for both companies

Forecasting is estimating future values by taking regard past values in the dataset. Now using our ARIMA models for respective companies we are going to forecast for next four years and see in which company it is safer to invest and for whom.
So, we convert the predicted values again to the power of 2.718 (as we did the transformation to make the data stationary) for both the datasets and use the forecast function of R library using their respective ARIMA models.

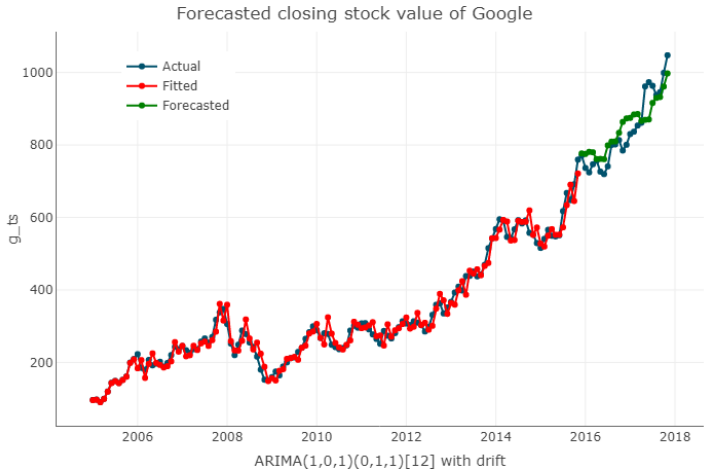

Fig.9

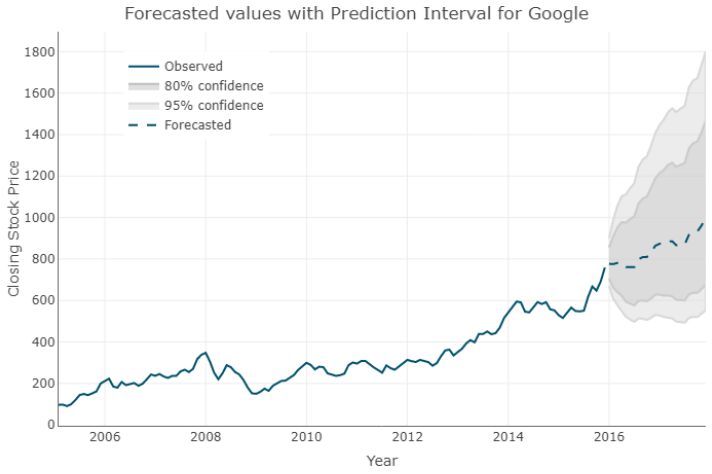

Fig.10

Observation: For both Google(Figure [9] and [10]) and Microsoft (Refer Figure [5] and [6] Appendix) we observe that, in both cases, the predicted stock price values are increasing with each consecutive year (in dotted lines). The forecast graph also depicts the 95% level of confidence level for both MSFT and Google's stock price values. We observe that for both Google and Microsoft the increase is not abrupt, but the estimated prices of both firms depict that their stock price would increase gradually and would not skyrocket in the upcoming 24 months.

We observed the prediction interval (at 80% and 95% confidence level) [8] and found that prediction interval of forecasted closing stock prices becomes more wider as we try to forecast consecutive closing stock prices of upcoming months.

Dashboard

Along with our time-series analysis in this project, we also built a simple navigable dashboard for our exploratory data analysis for normal users who can visualize some of our findings and many key findings for the targeted people. The dashboard is built on Tableau and can be accessed live from this link.

We built the dashboard to give a more clean and important insights from all the statistical methods we performed in R in layman language. The dashboard consists of many visualisations such as Past Trends, Average prices of stocks, including forecast. We made few of the dashboard pages interactive, so that the readers can filter variables and can compare different values on same screen. More information, working and motivation literature can be found in About page of the dashboard.

Problem that this dashboard solves: It uses both the datasets under consideration in this project, and for people who can't follow our EDA or want a quick insight of stocks of Google and Microsoft without getting their hands dirty on data analysis can use this dashboard.

Conclusion

1. Time-series Analysis ARIMA models are created with hyper-parameters p,q,d as (1,0,1) (0,1,1) [12] and (0,1,1) (2,1,0) [12] respectively for both Google and Microsoft from past stock price data.
2. The optimal ARIMA model has an **AIC** of **-259.58** and **-313.42** for Google and Microsoft respectively, which is the best value among all the models that we generated via **auto.arima** function.[9]
3. Moreover, according to AIC, all models are approximations to reality, and reality should never have a low dimensionality. Thus, we considered calculating **BIC** value also for the same model, it came to be **-245.68** and **-302.34** for Google and Microsoft respectively which is the best among the rest of the candidate models.
4. We built the models with **MAPE** values of **4.677%** and **4.85%** for Google and Microsoft ARIMA model on our test data, thus depicting the feasibility of the model.
5. Furthermore, we achieved **Mean Percentage Error (MPE)** value of **1.685%** and **2.488%** for Google and Microsoft ARIMA model on our test data respectively, which again supports our notion of feasibility of our built ARIMA model.
6. From the predicted time-plots of both companies we notice that both companies have good future and their stock prices are estimated to become higher in coming **24 months**. Adding to that, the value of stock price of Google is more than that of Microsoft and in coming 24 months it **will not cross** that of Google's.
7. Also this is just a statistical model, and it **DOES NOT** account for any irregular behaviour that a firm might have or any external changes in the environment in coming future, such as introduction of an innovative product by competing firm or evolution of any plague in the world effecting the economy.
8. Thus, we **reject H_0**, Null Hypothesis and can accept the Alternative Hypothesis (**H_1**) for both stocks price datasets (MSFT, Google), which states that there is gradual increase of stock prices over 24 months period.

Reference

[1] https://elvyna.github.io/2018/time-series-hypothesis-testing/
[2] https://stats.stackexchange.com/questions/114752/forecast-accuracy-calculation
[3] https://stackoverflow.com/questions/47119765/how-to-interpret-the-second-part-of-an-auto-arima-result-in-r
[4] https://www.itl.nist.gov/div898/handbook/pmc/section6/pmc624.htm
[5] https://otexts.com/fpp2/residuals.html
[6] https://cran.r-project.org/web/packages/TSstudio/index.html
[7] https://otexts.com/fpp2/non-seasonal-arima.html
[8] http://freerangestats.info/blog/2016/12/07/arima-prediction-intervals
[9] https://www.reddit.com/r/AskStatistics/comments/5ydt2c/if_my_aic_and_bic_are_negative_does_that_mean/