

# Assessing Time Series Models in Predicting Stock Market Trends

Jae Yun

University of California, Santa Barbara  
jaewoong@ucsb.edu

## Abstract

This study investigated the efficacy of time series models in forecasting stock prices of four major technology companies: Amazon, Microsoft, Nvidia, and Google. The SARIMA (Seasonal ARIMA) models were used over ten years of weekly opening price data from January 2010 to January 2020 to predict the stock prices for the year 2019. The analysis was concentrated on evaluating the reliability and accuracy of these models by comparing predicted values with actual stock prices, and excluding the influence of external events that are very significant such as the COVID-19 pandemic and global financial crisis. The findings showed that the four organizations had different levels of accuracy. Amazon stock prediction was the most accurate, it followed the real price movement closely during the year of 2019. Google and Nvidia's forecasts followed the actual trends until the third quarter of 2019, but from then on, the models missed the market dynamics. The largest deviation was Microsoft's stock, as the model did not manage to capture the price surge of 2019. The results indicate that ARIMA models are capable of extracting valuable information despite their forecasting accuracy. The research underlines the inherent shortcomings of purely technical forecasting in the rapidly changing sector of technology, stressing the need for a broader spectrum of factors, including economic, political, and pandemic conditions, to be taken into account in future forecasting exercises. Despite these restrictions, the analysis visualizes that statistical models are an essential part of a multi-faceted investment strategy, which highlights the need for advanced analysis in the stock market to make sound investment decisions.

## I. Introduction

The stock market has a long history of analysis and predictive modeling, which makes it difficult to penetrate, especially for the technology sector where players like Amazon, Microsoft, Nvidia, and Google have positioned themselves. These organizations have huge valuations and they are growing at a rapid pace over the last decade compared with others in different industries such as manufacturing which has led them to claim their position in the economic front line. However, producing an accurate prediction model of future prices is not straightforward. This means that any person who wants to accurately predict the future direction of the stock market must have ideas about how it works plus several other uncertainties including market sentimentality, finance background knowledge, and external factors that influence its behavior.

To evaluate the performance of four big companies from homogeneous industries this study focuses on these four technology giants. The analysis aims to determine whether the model effectiveness differs substantially by comparing stocks within the same industry irrespective of their sectoral-based variance in individual stock prices. By doing this, we will come up with comprehensive strategies regarding capturing time series dynamics in our model as well as take into consideration reasons for accuracy discrepancy across different but closely related sectors during review sessions.

The analysis aims to forecast stock prices of Amazon, Microsoft, Nvidia, and Google using time series models over ten years containing 552 weekly opening prices from January 1st, 2010 to January 1st, 2020. The point is to assess the reliability of our models by analyzing market trends and seasonal patterns; this means that we won't consider outside events when looking at things like the COVID-19 pandemic, Russian-Ukrainian conflict, global financial crisis, and other major economic shocks that may skew the data and reduce the accuracy for a purely technical (as opposed to fundamental) forecasting exercise.

For instance, it would be important in this case not to include any significant sporadic happenings as the study will have aimed at getting rid of some unpredictable factors that usually occur due to such factors as technological advancements or government interventions. This way, evaluating only time series models would enable researchers to identify underlying patterns since these crises are known for causing long-term impacts on leading technology firms' stocks.

Given this, the research will convert our data into a stationary process to ascertain the parameters for utilizing a time series forecasting model known as the SARIMA (Seasonal ARIMA) model to predict future stock prices of Amazon, Microsoft, Nvidia, and Google. The accuracy of these models will be determined by using 2010 to 2018 data to forecast stock prices for 2019. Predictive validity will be assessed by comparing forecasted values with actual stock prices observed in 2019, thus allowing AIC (Akaike Information Criterion) and visual representation of confidence intervals as measures of how well the model performed.

The point of this analysis is to investigate not only the appropriateness of time series forecasting within a stock market context but also explore whether these models could be one significant part of an investment strategic roadmap. This study also aims to evaluate how well time series models can do in a general volatile stock market environment.

## II. Key Terms

### Augmented Dickey-Fuller Test (ADF Test)

The ADF Test checks if a time series is stationary or not. This assumption is crucial because, for SARIMA models, the time series must be stationary - only then can the model parameters be estimated accurately and the forecasts be made dependable. A time series is said to be stationary if its statistical properties do not change with time. In other words, the mean, variance, and autocorrelation should remain constant over time. The ADF test formula can be expressed as:

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

where  $\Delta y_t$  is the first difference of the time series at time  $t$ ,  $\alpha$  is a constant term,  $\beta$  is the coefficient of a linear time trend, and  $\gamma$  is the coefficient of the lagged value  $y_{t-1}$ . The terms  $\delta_i$  are the coefficients of the lagged first differences, up to a maximum lag of  $p - 1$ . The error term  $\varepsilon_t$  is assumed to be white noise. [1]

### Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

While the ADF test involves subtracting lagged values of the time series to detect non-stationarity and its evidence is presented by the lack of trend, the KPSS test evaluates the strength of the trend component by summing squared deviations, which in turn helps in detecting stationarity; low p-value from ADF suggests non-stationarity while low p-value from KPSS implies stationarity. In practice both tests are used together: this allows for more comprehensive and balanced judgment about time series stationarity properties. In practice, both tests are used in conjunction to provide a more robust assessment of the time series stationarity.

The KPSS test checks for stationarity by assessing the presence of a stochastic trend, and is formulated as follows:

$$\text{KPSS} = \frac{S_T^2}{T\hat{\sigma}^2}$$

where: -  $S_T^2$  is the sum of squared deviations of the cumulative sum of residuals (or deviations from the trend), -  $T$  is the total number of observations, -  $\hat{\sigma}^2$  is an estimate of the variance of the residuals.[1]

### Box-Cox Transformation

Box-Cox transformation is a statistical approach used to normalize the distribution of a time series. The primary purpose of normalizing the time series is to stabilize variance. This method is especially helpful when variability is not constant - the data demonstrates non-constant variance (heteroscedasticity). Additionally, Box-Cox transformation can assist in achieving homoscedasticity and normal distribution for residuals which are vital conditions leading towards stationarity prerequisites needed for precise time series forecasting: In particular, the transformation comes in handy where equal variance assumption does not hold. The mathematical definition of the Box-Cox transformation is as follows:

For a time series  $y_t$  and a transformation parameter  $\lambda$ , the Box-Cox transformed series  $y'_t$  is computed using:

$$y'_t = \begin{cases} \frac{y_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(y_t) & \text{if } \lambda = 0. \end{cases}$$

Here: -  $y_t$  is the observed value at time  $t$ . -  $\lambda$  is the Box-Cox transformation parameter, which is determined empirically to best stabilize the variance of the time series. -  $y'_t$  represents the transformed value at time  $t$ .

The parameter  $\lambda$  is typically estimated to maximize the likelihood function, and it can vary across a wide range. When  $\lambda = 1$ , the transformation is equivalent to no transformation (the identity function). When  $\lambda = 0$ , the Box-Cox transformation corresponds to taking the natural logarithm of  $y_t$ . The goal is to find the value of  $\lambda$  that results in a time series with constant variance and approximately normal residuals.

After transforming the stationary series, we evaluate further to ensure the data's homoscedasticity and normality. When these properties are established, the subsequent analyses on time series, which include ARIMA modeling, are carried out using the transformed data. Any predictions or forecasts obtained from this model need to be back-transformed to the original scale. This can be achieved through applying an inverse Box-Cox transformation.

### Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test is used for checking stationarity in a time series without having to identify the presence of a unit root. It tests whether the series is stationary around a deterministic trend or a stochastic trend. It helps distinguish between level stationarity and trend stationarity— important information especially when dealing with fluctuating data points due to different reasons like drifting mean or evolving trends. KPSS test formula is as follows:

For a time series  $y_t$ , the KPSS statistic is computed as:

$$\text{KPSS} = \frac{\sum_{t=1}^T \hat{S}_t^2}{T^2 \hat{\sigma}^2}$$

where: -  $T$  is the number of observations in the time series. -  $\hat{S}_t$  is the partial sum of the residuals (or deviations from the mean for level stationarity, or from the trend for trend stationarity), up to time  $t$ . -  $\hat{\sigma}^2$  is an estimate of the long-run variance of the time series.

The null hypothesis ( $H_0$ ) of the KPSS test indicates that the time series is stationary. A small p-value (typically less than 0.05) indicates that the null hypothesis is rejected, suggesting the presence of a unit root, suggesting a non-stationary time series. A large p-value suggests a stationary time series.

### Time Series Decomposition

Decomposition for a time series ( $Y(t)$ ) is used to break down a time series into several components, such as trend, seasonal, and random/residual components. The random/residual component following the decomposition may render the time series stationary. The purpose is to provide a better understanding, analysis, and forecast of the data. The formula is as follows

$$Y(t) = T(t) + S(t) + R(t)$$

- $Y(t)$  is the observed value of the time series at time  $t$ , given by  $T(t) + S(t) + R(t)$ .
- $T(t)$  is the trend component of the time series at time  $t$ , represented as  $T_t = f(t)$ .
- $S(t)$  is the seasonal component of the time series at time  $t$ , calculated as  $S_t = \sum_{i=1}^n s_i$ .
- $R(t)$  is the residual or random component of the time series at time  $t$ , defined as  $R_t = Y_t - T_t - S_t$ . [1]

## Autocorrelation function (ACF)

The Autocorrelation Function (ACF) measures the linear relationship between an observation in a time series and past observations (at previous times/lags). The ACF plot tells you how many of the previous points/lags in the time series are having a statistically meaningful impact on the currently active point. This is important in determining the ‘Moving Average’ (MA) component in ARIMA models.

The ACF is given by the expression

$$\text{ACF}(k) = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

where  $k$  is the lag,  $n$  is the number of observations in the time series,  $Y_t$  is the observed value at time  $t$ , and  $\bar{Y}$  is the sample mean of the time series.[1]

## Partial Autocorrelation Function (PACF)

The PACF provides a measure of the correlation between observations in a series that are a certain time lag apart while removing the correlations that might exist at shorter lags. This makes PACF particularly useful in identifying the ‘Autoregressive’ (AR) part of the ARIMA model, which points to the degree of direct relationship between an observation and a lag.

The PACF is expressed as

$$\rho_k = \frac{\text{cov}(Y_t, Y_{t-k})}{\text{var}(Y_t)} - \sum_{j=1}^{k-1} \phi_j \rho_{k-j}$$

where  $\rho_k$  is the partial autocorrelation at lag  $k$ ,  $\text{cov}(Y_t, Y_{t-k})$  is the covariance between  $Y_t$  and  $Y_{t-k}$ ,  $\text{var}(Y_t)$  is the variance of  $Y_t$ ,  $\phi_j$  is the estimated coefficient of the autoregressive (AR) model at lag  $j$ , and  $k$  is the maximum lag for which the PACF is computed.[1]

## SARIMA Model

The SARIMA (Seasonal Autoregressive Integrated Moving Average) model is an enhanced version of the ARIMA model. It is more convenient for analyzing and forecasting time series data with seasonal features. Such data could be any physical quantity that has regular oscillations or seasonal features where the cycle repeats itself year after year.

The SARIMA Model is formulated as follows

$$\begin{aligned} & (1 - \phi_1 B - \dots - \phi_p B^p)(1 - \Phi_1 B^s - \dots - \Phi_P B^{sP})^d (y_t - \mu) \\ & = (1 + \theta_1 B + \dots + \theta_q B^q)(1 + \Theta_1 B^s + \dots + \Theta_Q B^{sQ}) \varepsilon_t \end{aligned}$$

where  $y_t$  is the observed value of the time series at time  $t$ ,  $\mu$  is the mean of the time series,  $\varepsilon_t$  is the error term at time  $t$ , and  $B$  is the backshift operator. The parameters  $p$ ,  $d$ , and  $q$  correspond to the order of the autoregressive (AR), integrated (I), and moving average (MA) components, respectively. The parameters  $P$ ,  $D$ , and  $Q$  correspond to the order of the seasonal AR, seasonal I, and seasonal MA components, respectively. The parameter  $s$  represents the length of the seasonal cycle.[1]

The  $\phi$  and  $\theta$  parameters are the coefficients of the AR and MA terms, respectively, while the  $\Phi$  and  $\Theta$  parameters are the coefficients of the seasonal AR and seasonal MA terms

## Ljung-Box Test

The Ljung-Box Test is one of the most commonly used types of statistical tests to check for the absence of autocorrelation at different lag lengths in residuals from a time series analysis. The idea is to ensure that the time series is random since this is a fundamental assumption in the formalism of time series modeling and also to make sure that the SARIMA model is adequate.

The Ljung-Box Test is expressed as

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k}$$

where  $Q$  is the test statistic,  $n$  is the sample size,  $h$  is the number of lags being tested, and  $\hat{\rho}_k$  is the sample autocorrelation at lag  $k$ . The null hypothesis of the test is that the autocorrelations up to lag  $h$  are equal to zero, indicating that the time series is a white noise process. The alternative hypothesis states that the autocorrelations are not equal to zero, indicating the presence of serial correlation in the data. The test statistic  $Q$  follows a chi-squared distribution with  $h-p$  degrees of freedom, where  $p$  is the number of parameters estimated in the time series model.[1]

### Box-Pierce Test

The Box-Pierce Test is also another well-known statistical test similar to the Ljung-Box Test, used to check the independence of residuals in a time series model. The test predates the Ljung-Box Test and is based on a simpler formulation. This purpose of the test is to detect any autocorrelation in the residuals of a fitted time series model, including the SARIMA model.

The Box-Pierce is formulated as

$$Q^* = n \sum_{k=1}^h \hat{\rho}_k^2$$

where  $Q^*$  is the test statistic,  $n$  is the sample size,  $h$  is the number of lags being tested, and  $\hat{\rho}_k$  is the sample autocorrelation at lag  $k$ . The null hypothesis of the test is that the autocorrelations up to lag  $h$  are equal to zero, indicating that the time series is a white noise process. The alternative hypothesis states that the autocorrelations are not equal to zero, indicating the presence of serial correlation in the data.

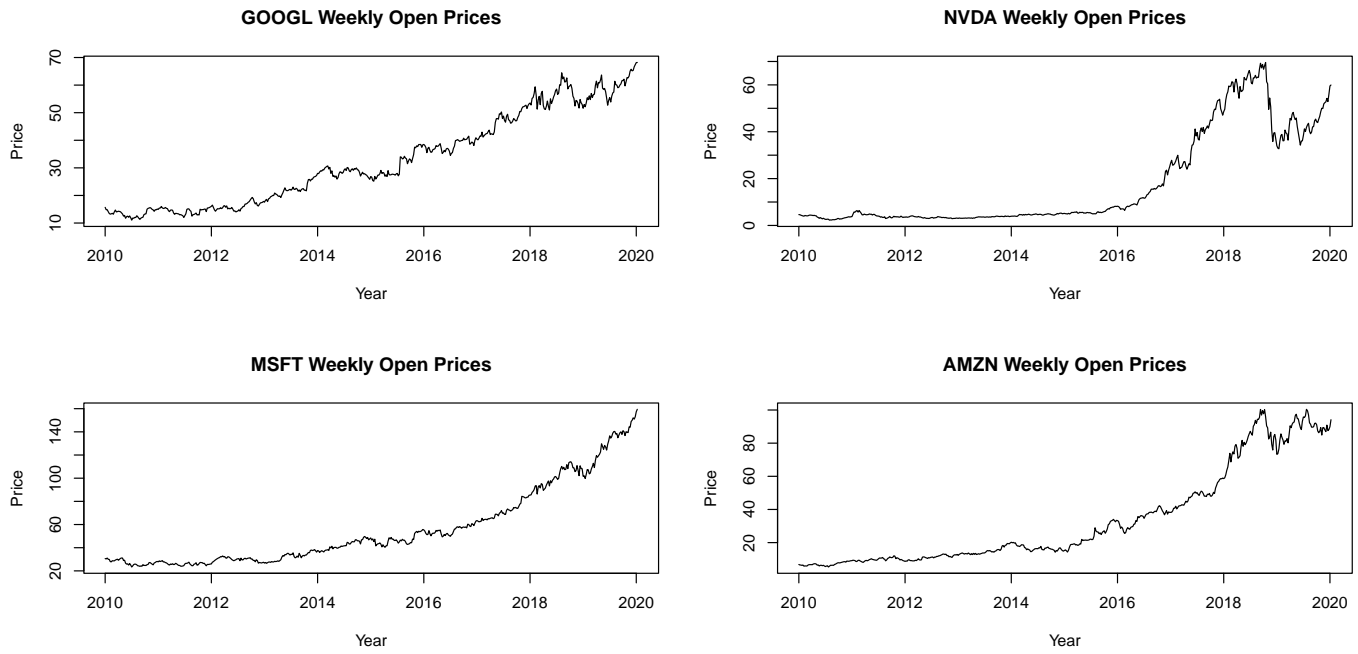
The test statistic  $Q^*$  follows an approximately chi-squared distribution with  $h-p$  degrees of freedom, where  $p$  is the number of parameters estimated in the time series model.[1]

## III. Data

In our time series dataset, we observe five key dimensions, commonly referred to as the 5V's, that significantly enhance the depth and utility of our analysis:

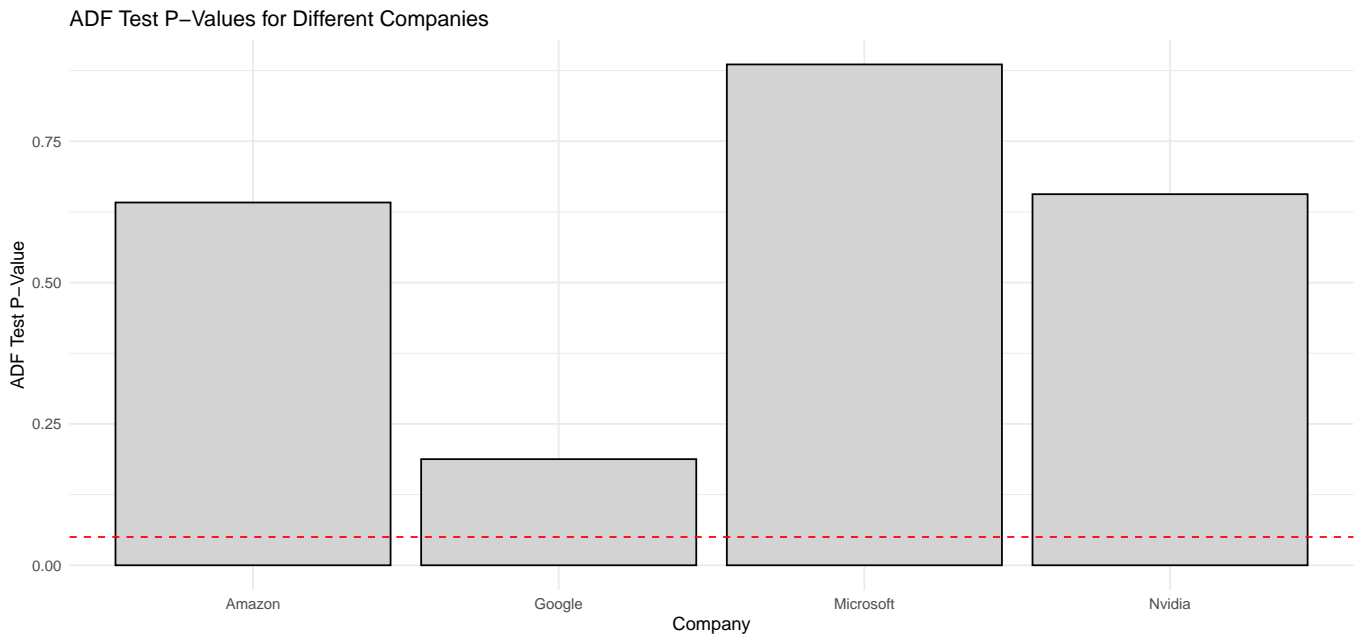
- **Volume:** Our dataset encompasses a substantial volume of data, spanning ten years of weekly stock market information. This extensive collection of data lays a solid foundation for making precise and effective predictions.
- **Velocity:** The dataset's velocity is characterized by its weekly frequency, providing a steady and consistent flow of information for timely analysis.
- **Variety:** Despite focusing on just four major stocks within the technology sector, our dataset is rich in variety. Each stock represents distinct market behaviors and price patterns, unique to each company, offering a diverse perspective.
- **Veracity:** The data, sourced from Yahoo Finance, is marked by its high level of veracity. Reflecting actual stock market prices, this dataset offers exceptional accuracy, crucial for reliable analysis.
- **Value:** Analyzing this data allows us to identify market trends and perform in-depth analysis. Such insights are invaluable not only for understanding past market behaviors but also for forecasting future trends. This is particularly significant given the influential nature of these technology stocks in the broader economic context.

Now, we can begin using R to import our data from yahoo, read csv, and plot the time series data:



Looking at the trends of each weekly opening price for our 4 technology stocks, we see a steady increase from 2010 to 2016, and a sharp, yet fluctuating increase from 2016 to 2020.

These plots show the presence of non-stationarity and non-randomness within our datasets. Subsequently, we must perform differencing or decomposition to accurately utilize our forecasting model. We can also construct histograms to better visualize the distribution of our current non-stationary dataset:



Using the ADF test, we extract the p-value for each dataset. Each p-value exceeds the significance value represented as the red dotted line where  $p = 0.05$ . Consequently, we fail to reject the null hypothesis and conclude that we do not have sufficient evidence to determine that each data is non-stationary.

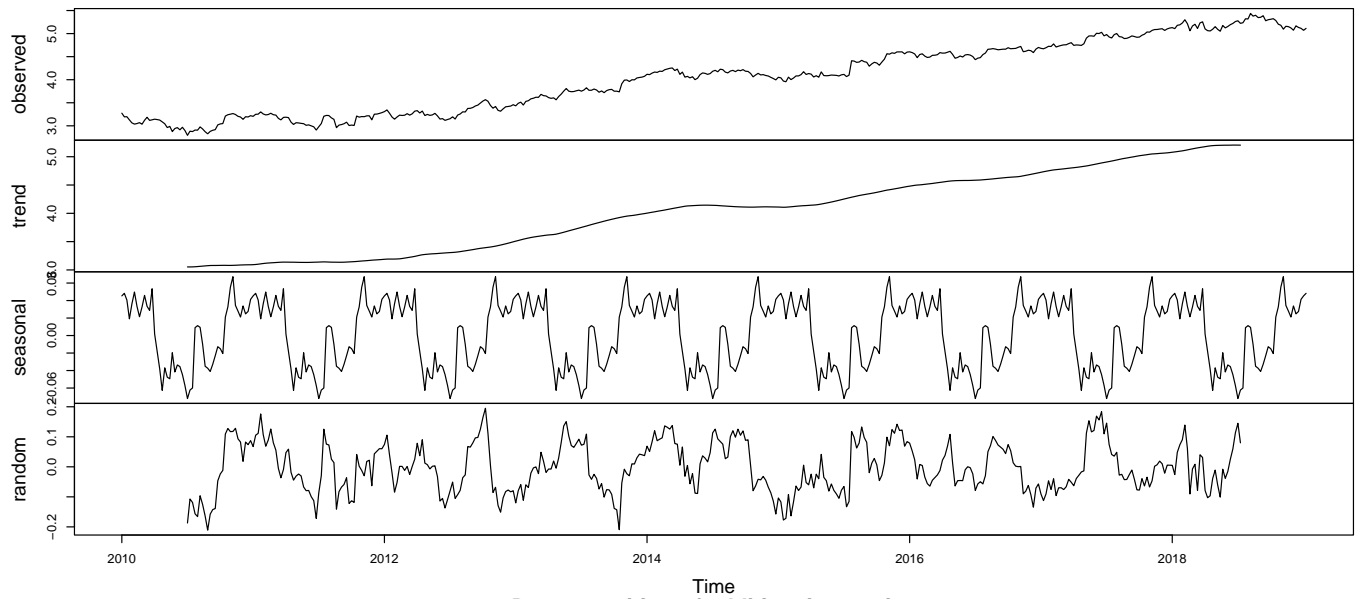
## IV. Data Augmentation

While conducting our study, we realized that decomposition by itself cannot render the random nature of the dataset, and therefore we used the Box-Cox transformation as a preliminary step before decomposition. This conversion is the method of variance non-stationarity mitigation, which is the typical characteristic of financial time series data. We intend to achieve this by converting the series into one that has a consistent variance and a more symmetric shape

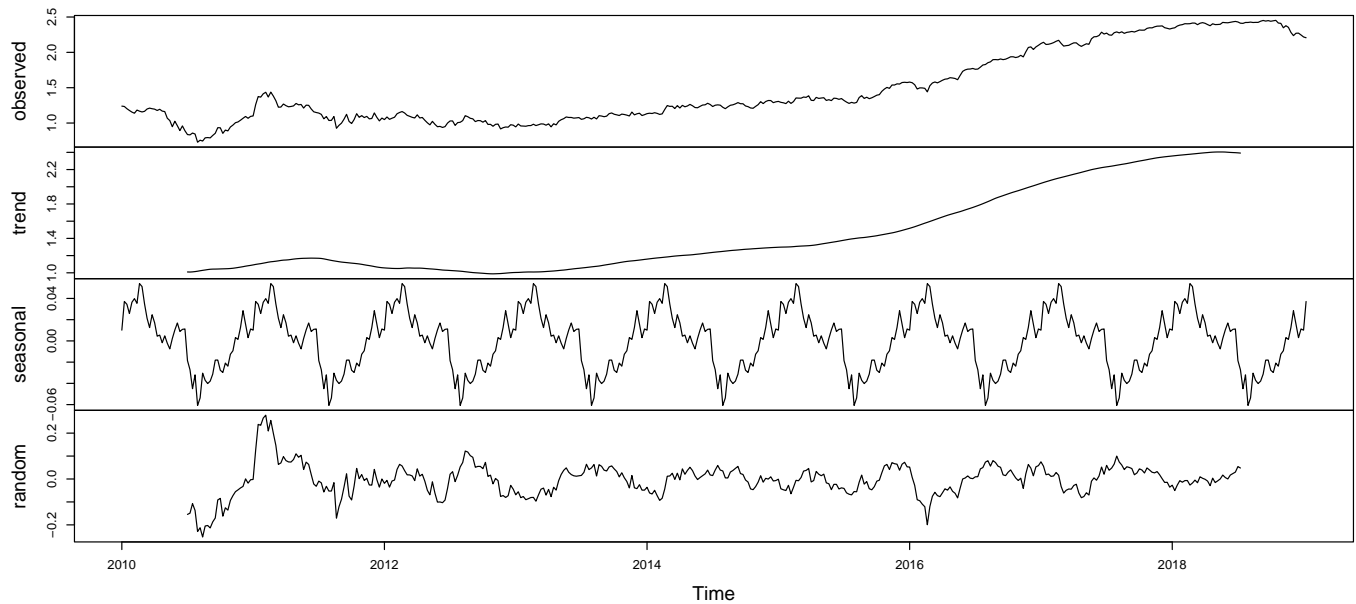
which will make the decomposition process easier since we will be able to isolate and remove trend and seasonal components effectively.

We can then plot a decomposition of the Box-Cox to identify any trends or seasonality patterns.

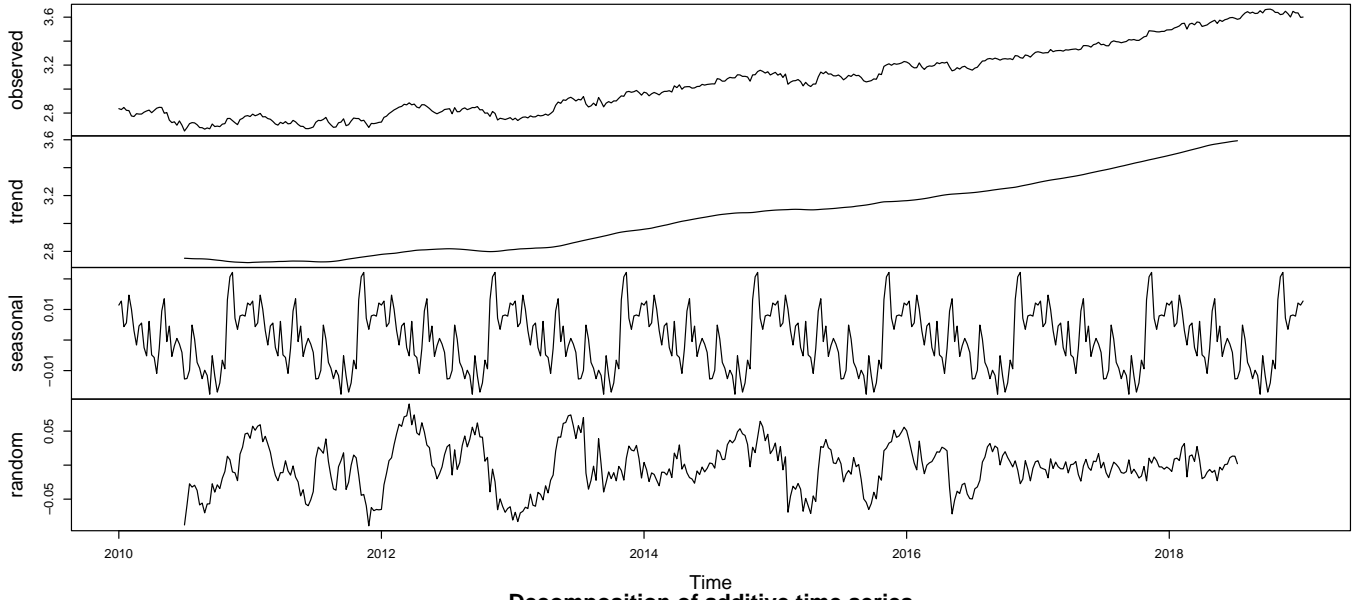
**Decomposition of additive time series**



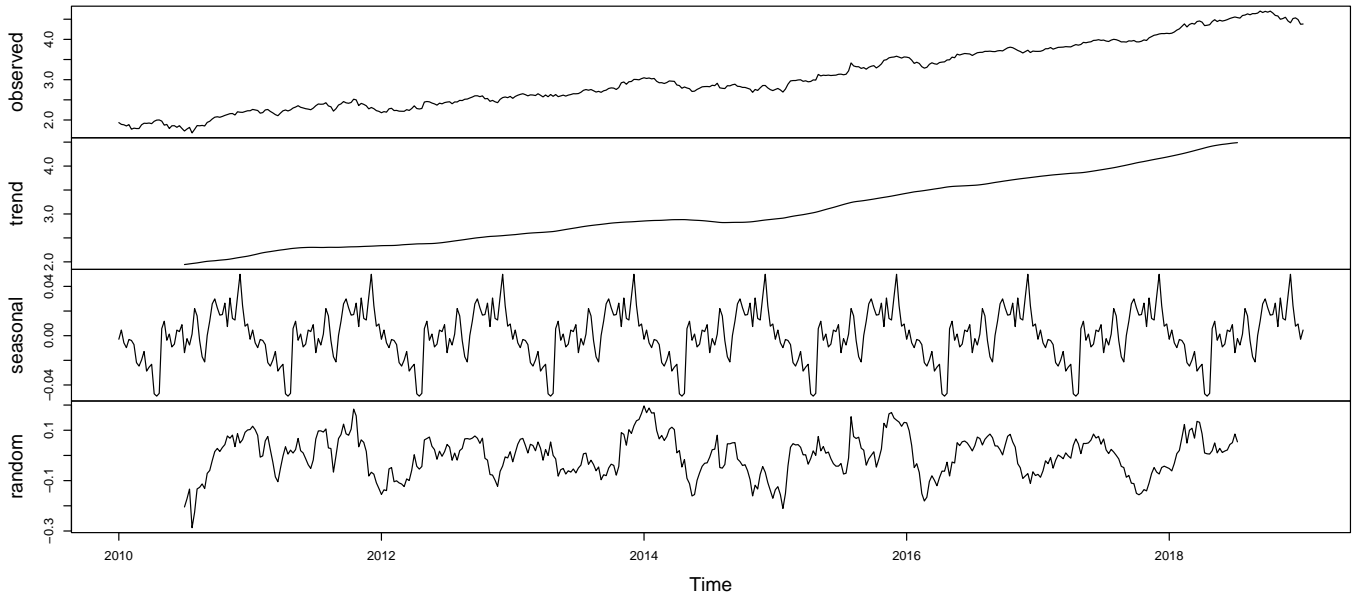
**Decomposition of additive time series**



**Decomposition of additive time series**



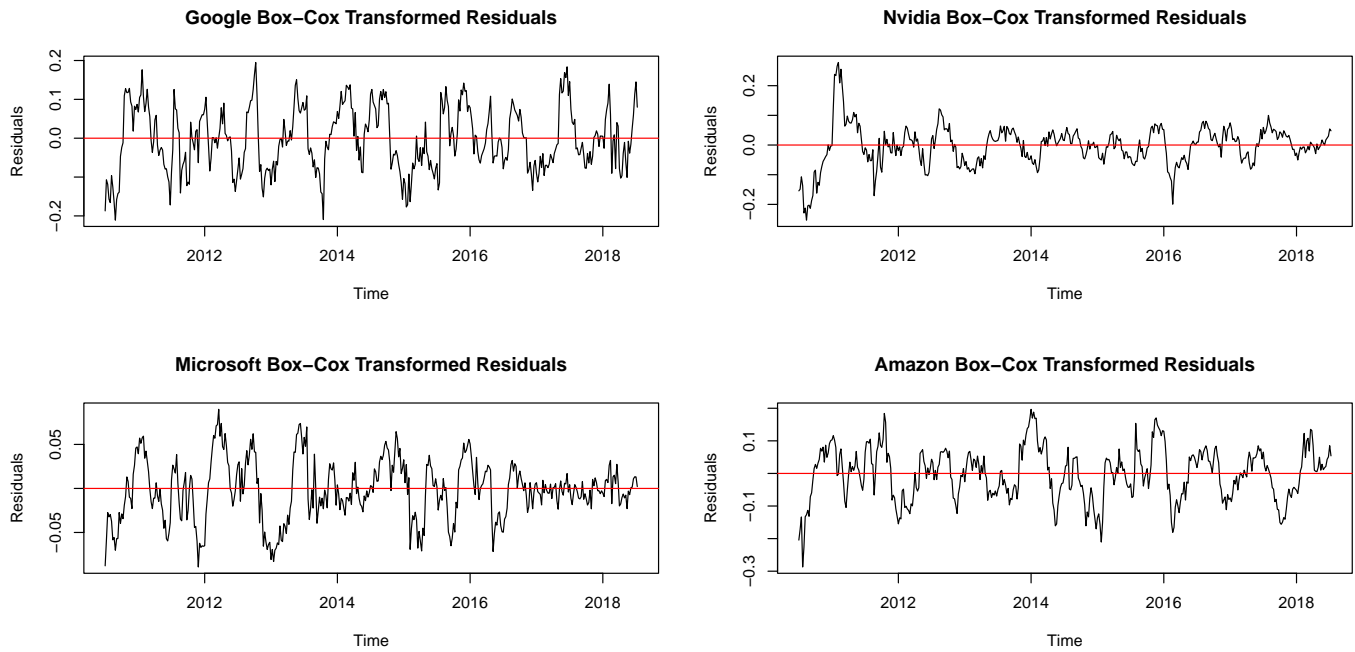
**Decomposition of additive time series**



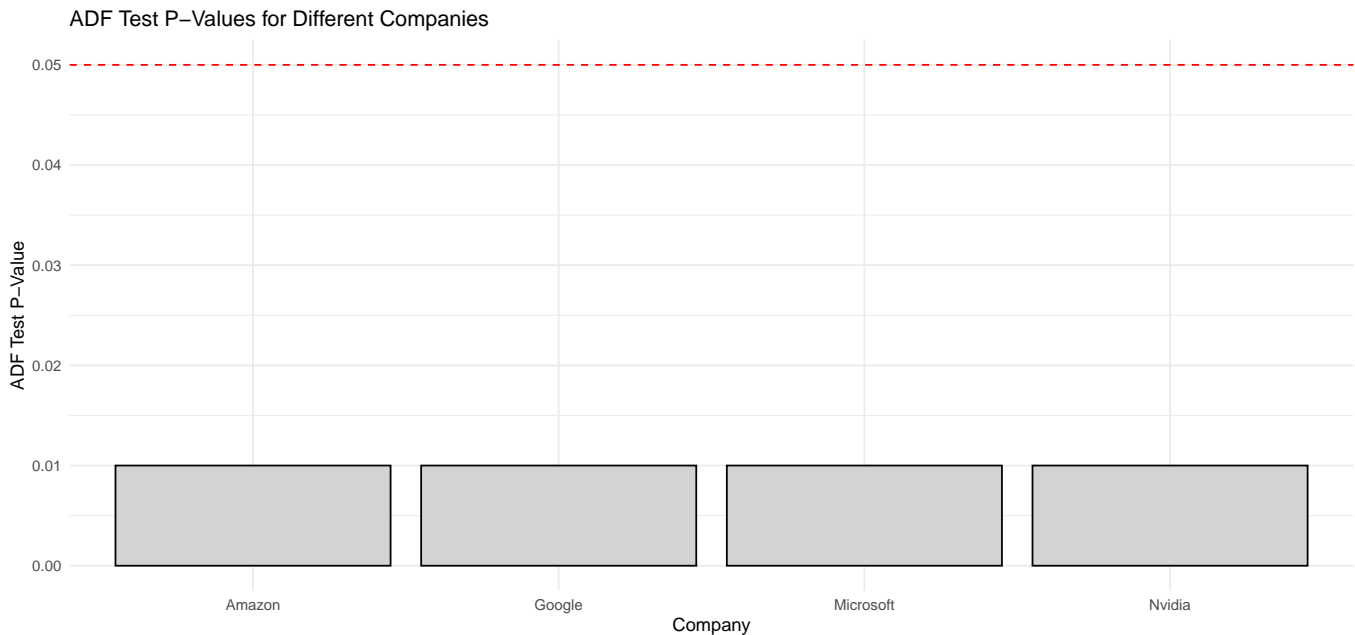
A white noise distribution is found to be the closest approximation to the random components of the Box-Cox transformed time series for all four of the time series. This similarity means that the random components are stationary, giving a basis for the identification of parameters of our SARIMA models with stronger confidence. To ensure, we can apply the ADF test on every decomposed box-cox transformed random component.

Here are our plots for our Box-Cox transformed residual-extracted components to check for stationarity.





It seems we have achieved stationarity within our twice-augmented dataset. We can also test the statistical hypothesis for stationarity using the ADF test.

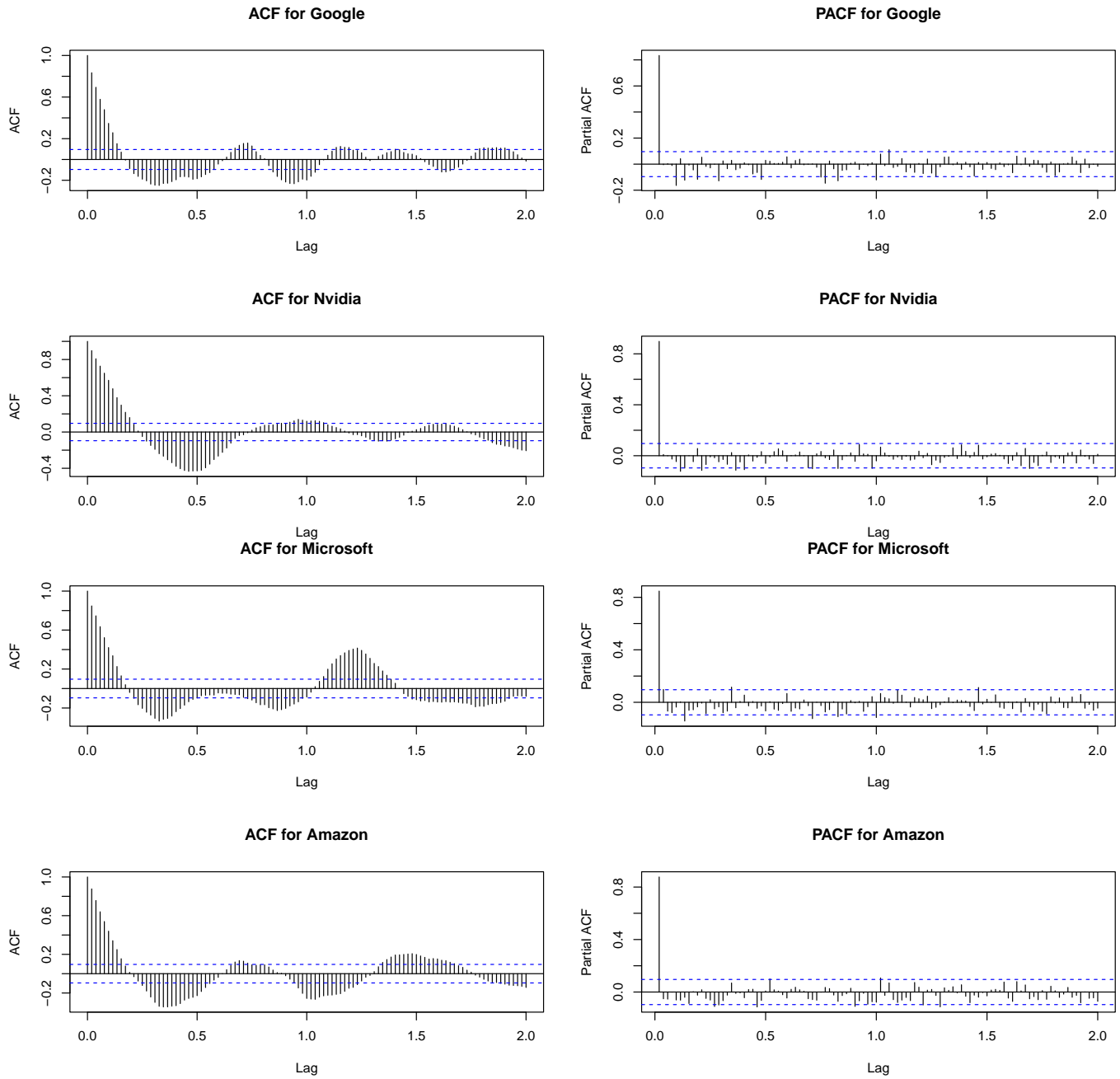


Our ADF Test P-values for each twice-augmented datasets are now below the significance level of 0.05, and thus we reject the null hypothesis for non-stationarity. Therefore, we conclude with significant evidence that each of our time series data are stationary. Consequently, we can move on with the next steps in selecting parameters in our SARIMA model using PACF and ACF plots.

## V Models and Parameter Selection

We will employ the SARIMA model to evaluate performance, selecting the optimal model based on the lowest Akaike Information Criterion (AIC) values. Model parameters will be determined through analysis of the ACF and PACF plots.

Considering SARIMA model follows parameters  $Sarima(p, d, q, P, D, Q)$ , we will use this in our evaluation.



- A significant spike at lag 1 (outside the confidence interval) may suggest an AR(1) component ( $p = 1$ ) for the non-seasonal part of the model.
- The need for differencing ( $d$ ) may not be necessary since the tests indicate the series is already stationary ( $d=0$ ).
- The exact number ( $q$ ) can be challenging to determine solely from the ACF plot when there are many significant lags.
- The sine-wave like features represent our data still contains a seasonal pattern. Thus, we need to difference it further using seasonal differencing. Thus we will set  $D = 1$ , and set  $P = 0$  or  $1$ , and  $Q = 0$  or  $1$ .

Thus, we choose a SARIMA model with parameters SARIMA( $p = 1, d = 0, q = 1$  or  $2, P = 0$  or  $1, D = 1, Q = 0$  or  $1$ ).

- We also see very similar patterns for the other ACF and PACF plots

The chosen SARIMA model parameters for the NVIDIA, Microsoft, and Amazon residuals could be SARIMA( $p = 1, d = 0, q = 1$  or  $2, P = 0$  or  $1, D = 1, Q = 0$  or  $1$ ).

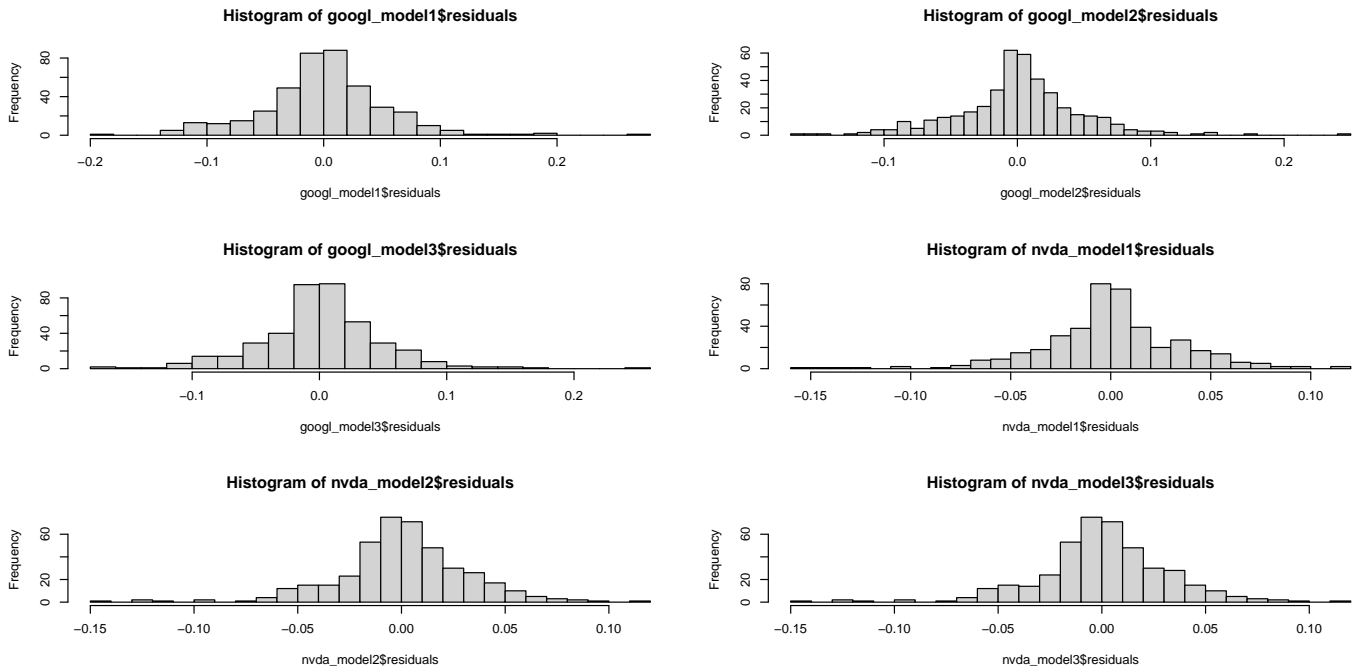
```
## Google Model 1 AIC Google Model 2 AIC Google Model 3 AIC Nvidia Model 1 AIC
## 1 -1041.696 -1080.486 -1080.745 -1345.771
## Nvidia Model 2 AIC Nvidia Model 3 AIC
## 1 -1389.588 -1387.605

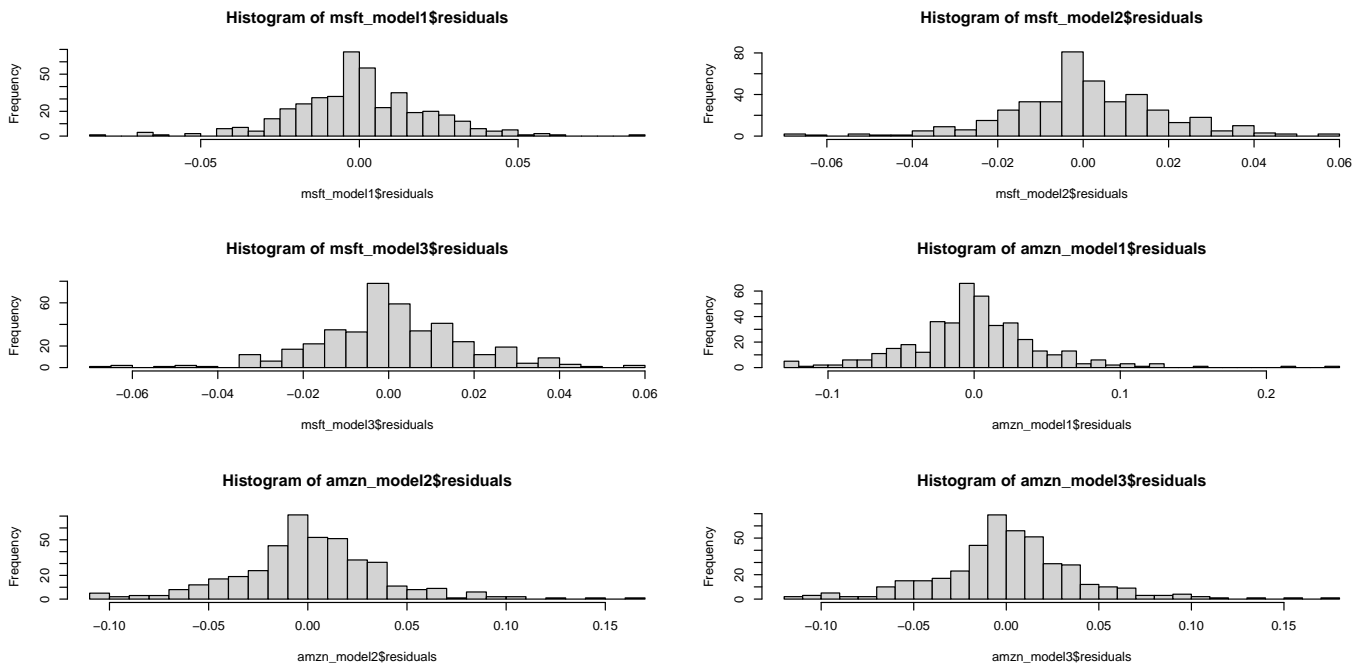
## Microsoft Model 1 AIC Microsoft Model 2 AIC Microsoft Model 3 AIC
## 1 -1710.289 -1759.648 -1758.092
## Amazon Model 1 AIC Amazon Model 2 AIC Amazon Model 3 AIC
## 1 -1150.074 -1205.664 -1205.35
```

The analysis of the AIC values for the three SARIMA models utilized for each of the four technology stocks (Google, Nvidia, Microsoft, and Amazon) shows that Model 1 is superior to the other two models in every case. Model 1's AIC values being the lowest for all four stocks suggest that it is the best fit among the three models in terms of the balance between the goodness of fit and model complexity. The AIC values for all three models are acceptable implying that the SARIMA approach is appropriate for modeling the stock price time series. Nevertheless, the fact that the AIC values are not the same for all the models indicates the importance of closely choosing the model parameters. It has been shown that Model 1 outperforms the other two models, and is thus the best one to use for forecasting the stock prices of these technology companies.

## VI. Model Evaluation on Train Data

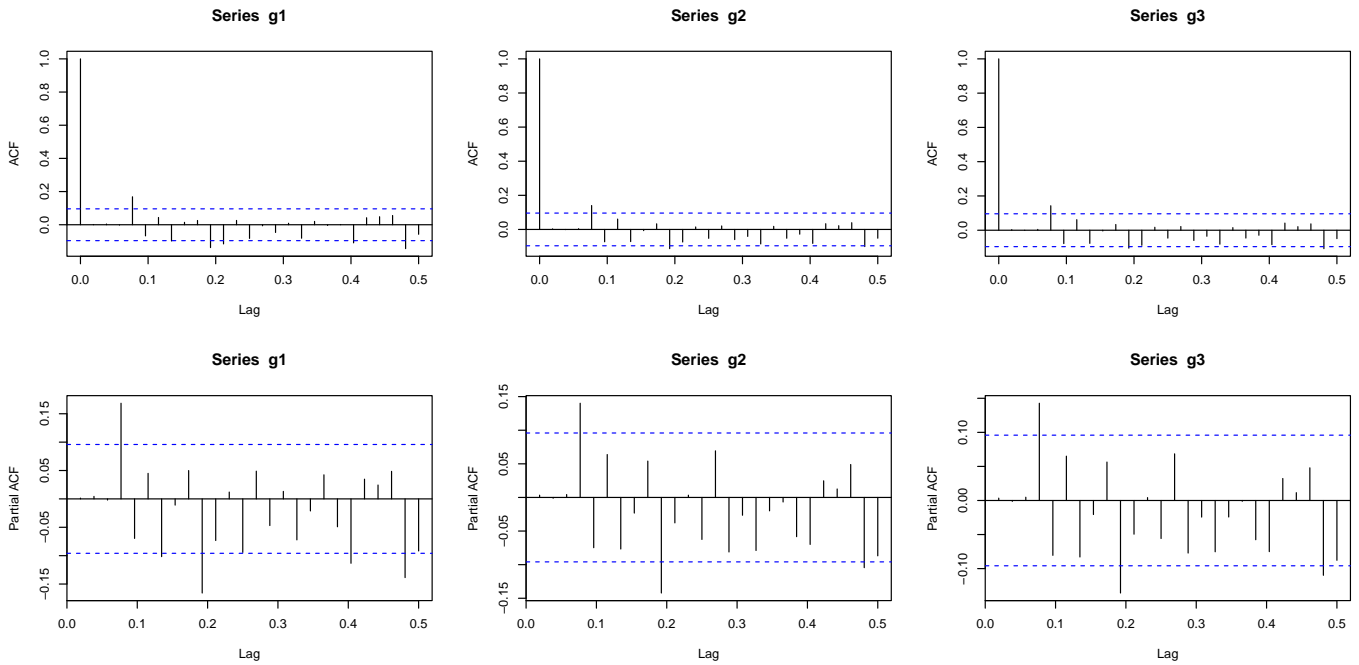
Our residuals should look like a normal white noise to represent an adequate model fit. We can test this using histograms.



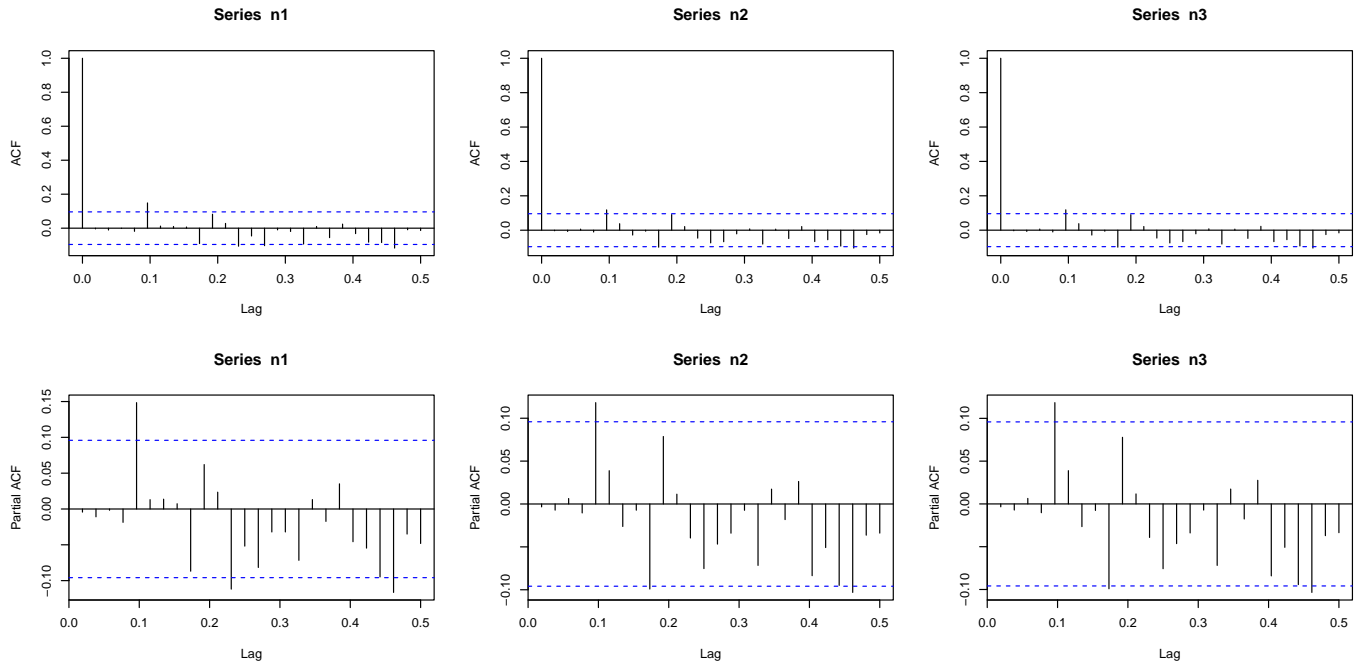


Our histograms look normal and resembles a gaussian white noise, which is a clear indicator that our models performed well.

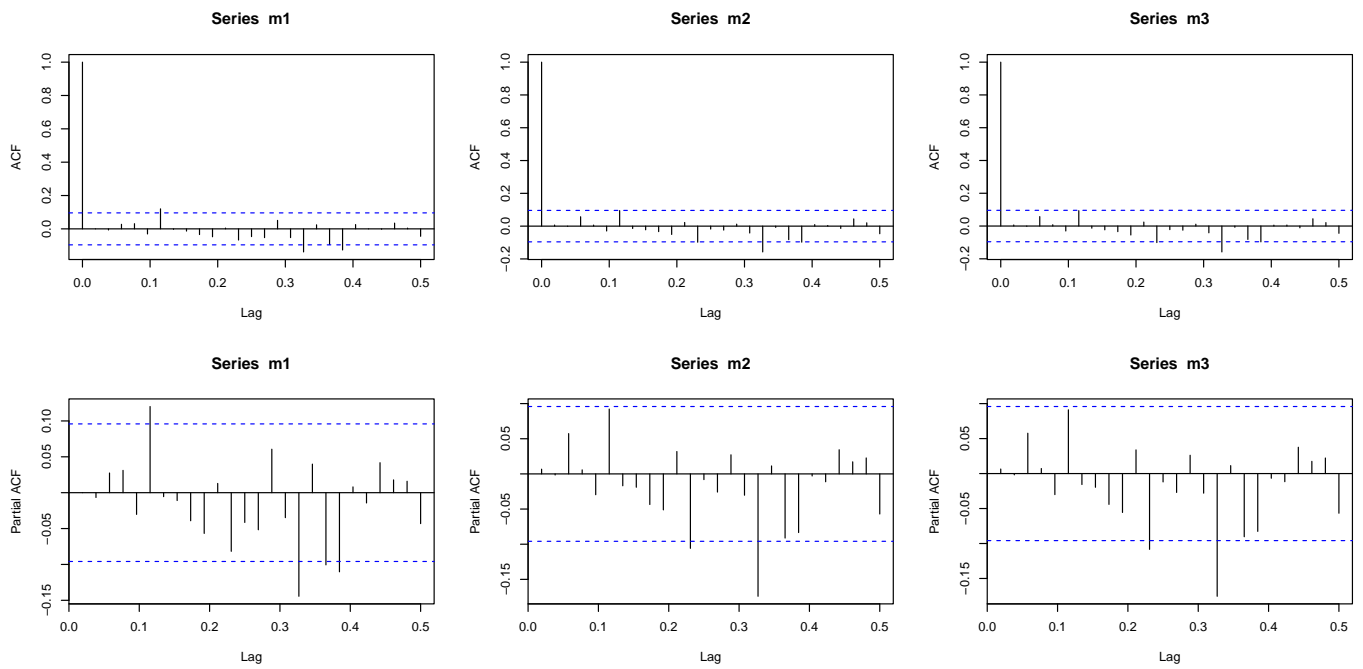
### PACFS and ACF of Google model residuals



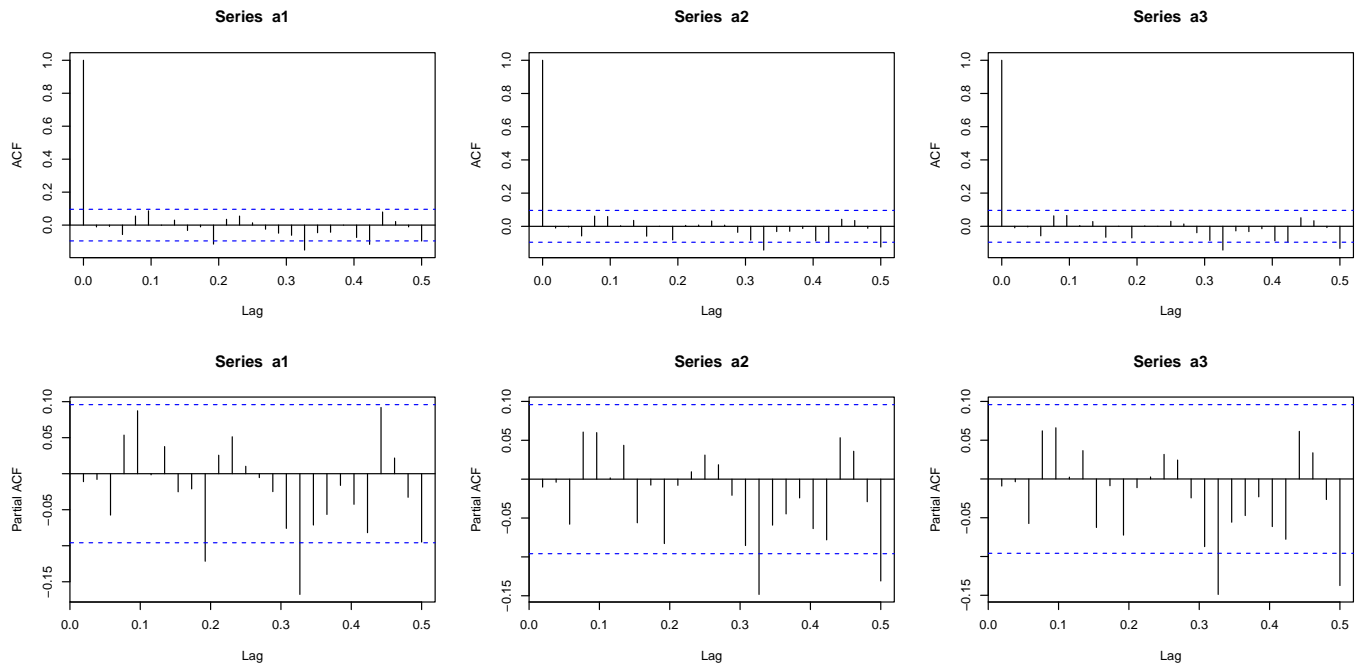
## PACFS and ACF of Nvidia model residuals



## PACFS and ACF of Microsoft model residuals



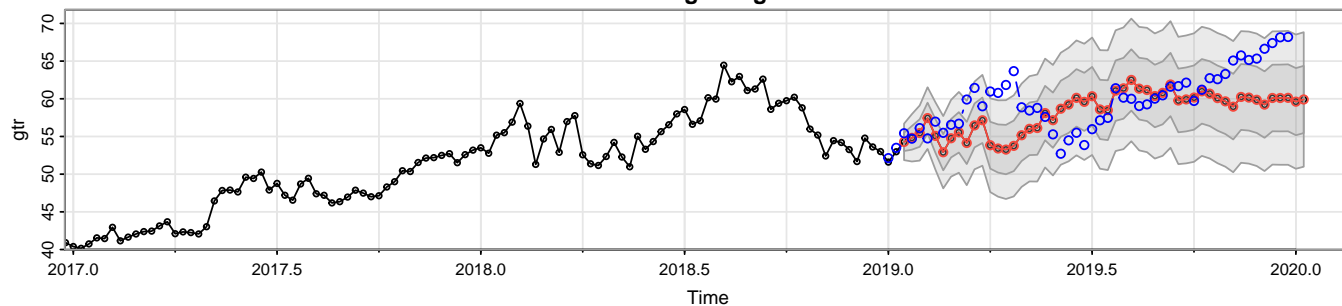
## PACFS and ACF of AMAZON model residuals



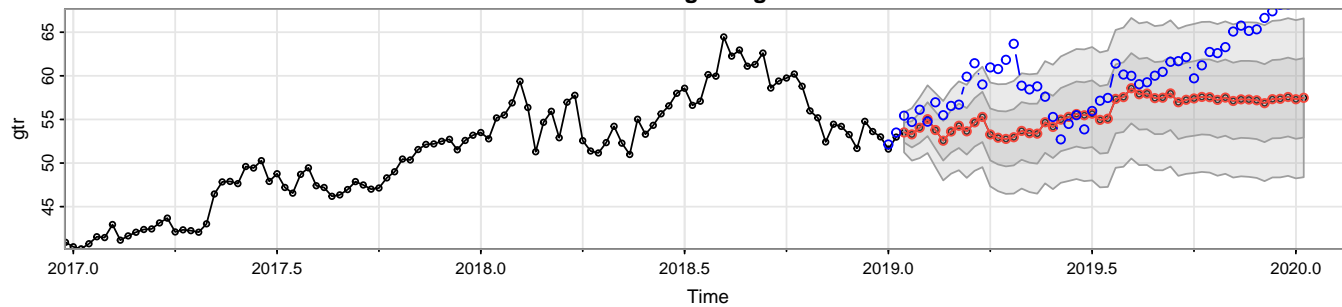
All in all, our pacf and acf charts for all of our stocks do resemble a white noise distribution, and we can continue forecasting our datasets.

## VII. Forecasting with Test Data

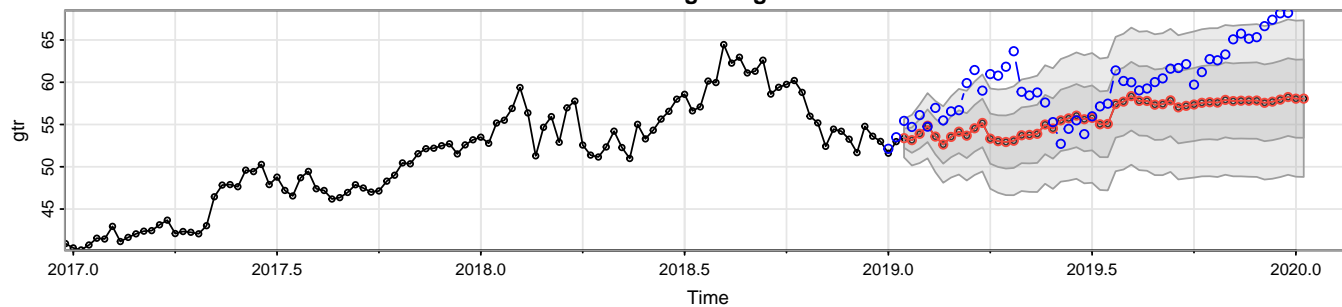
**Model 1: Forecasting Google Stock for 2019**



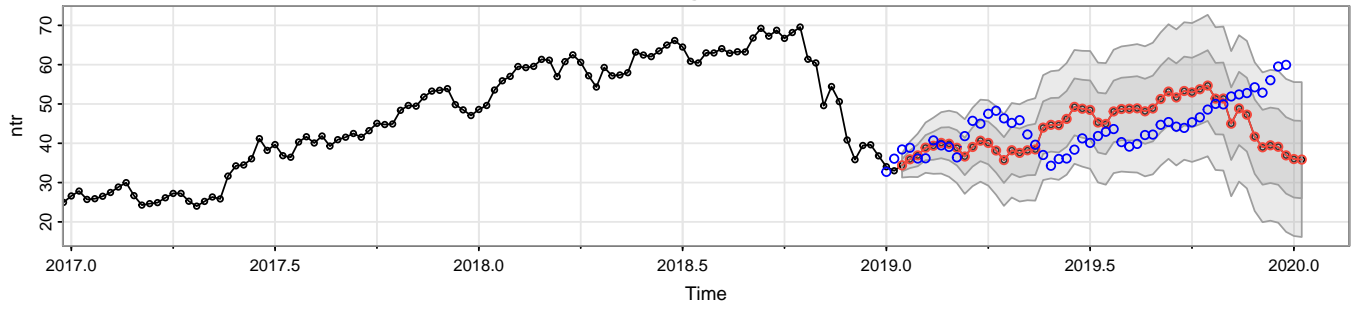
**Model 2: Forecasting Google Stock for 2019**



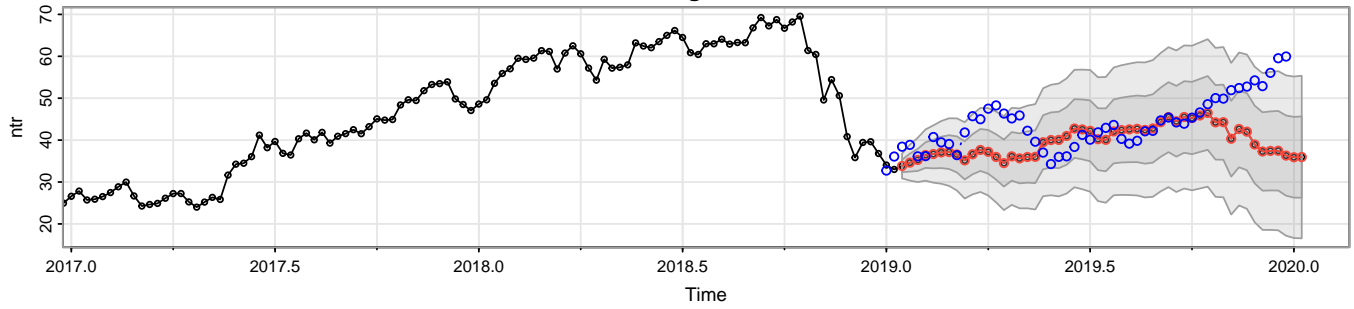
**Model 3: Forecasting Google Stock for 2019**



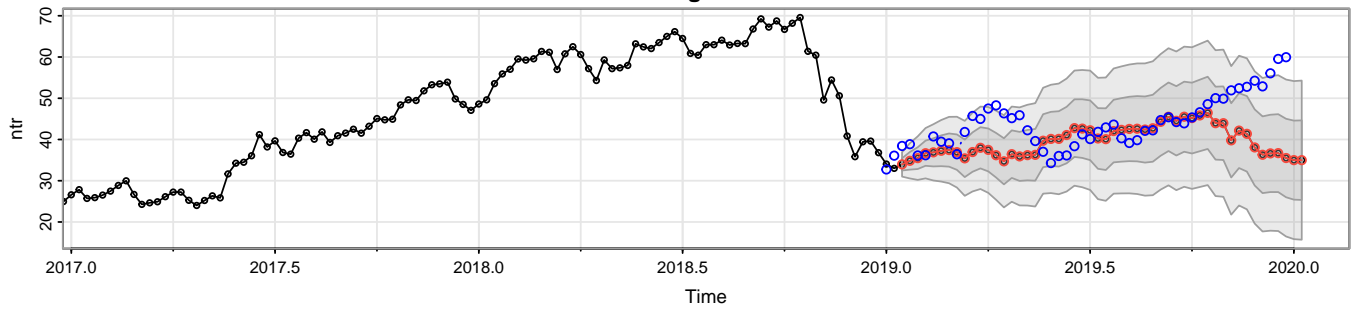
**Model 1: Forecasting Nvidia Stock for 2019**



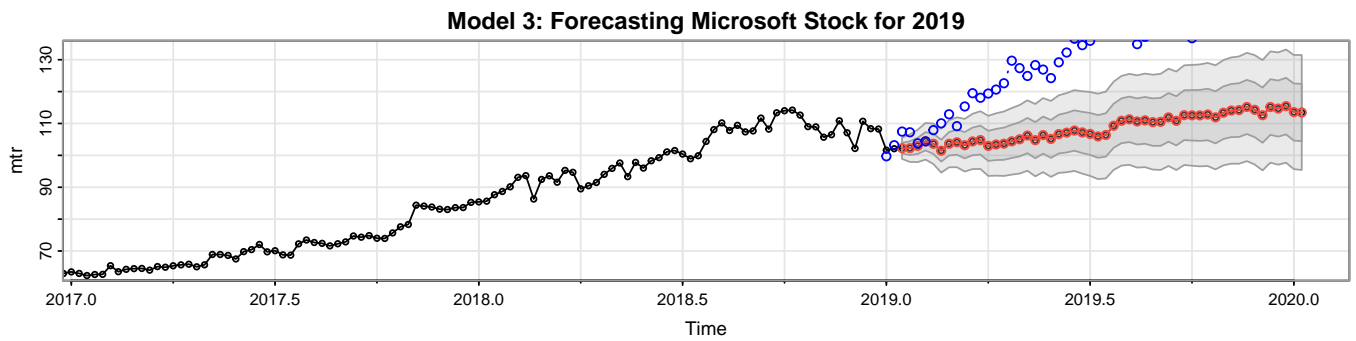
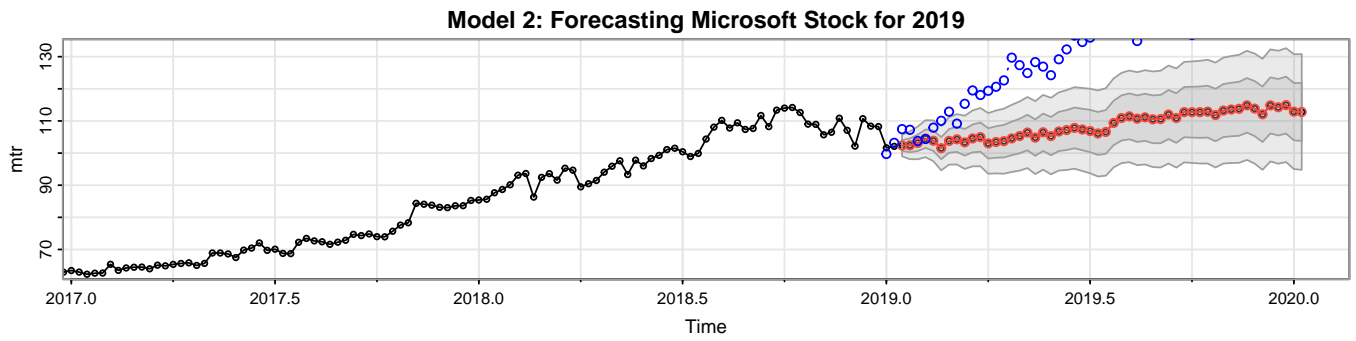
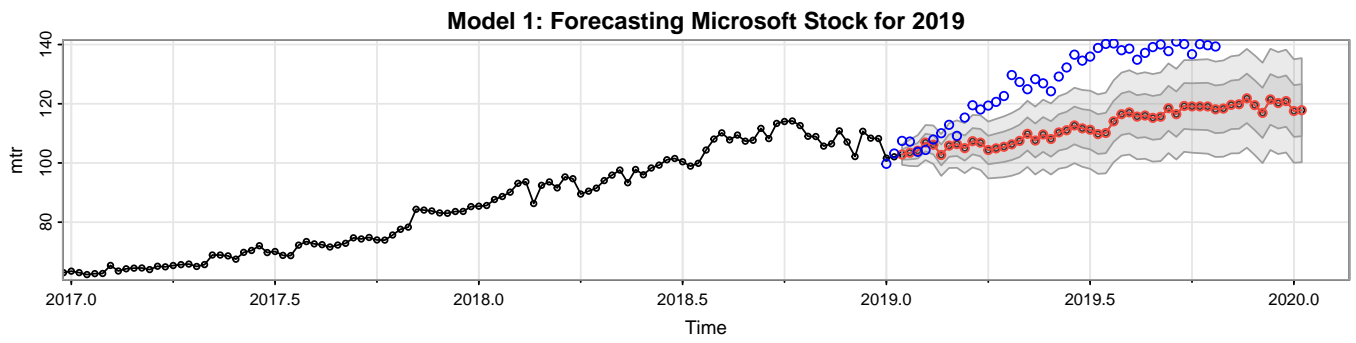
**Model 2: Forecasting Nvidia Stock for 2019**

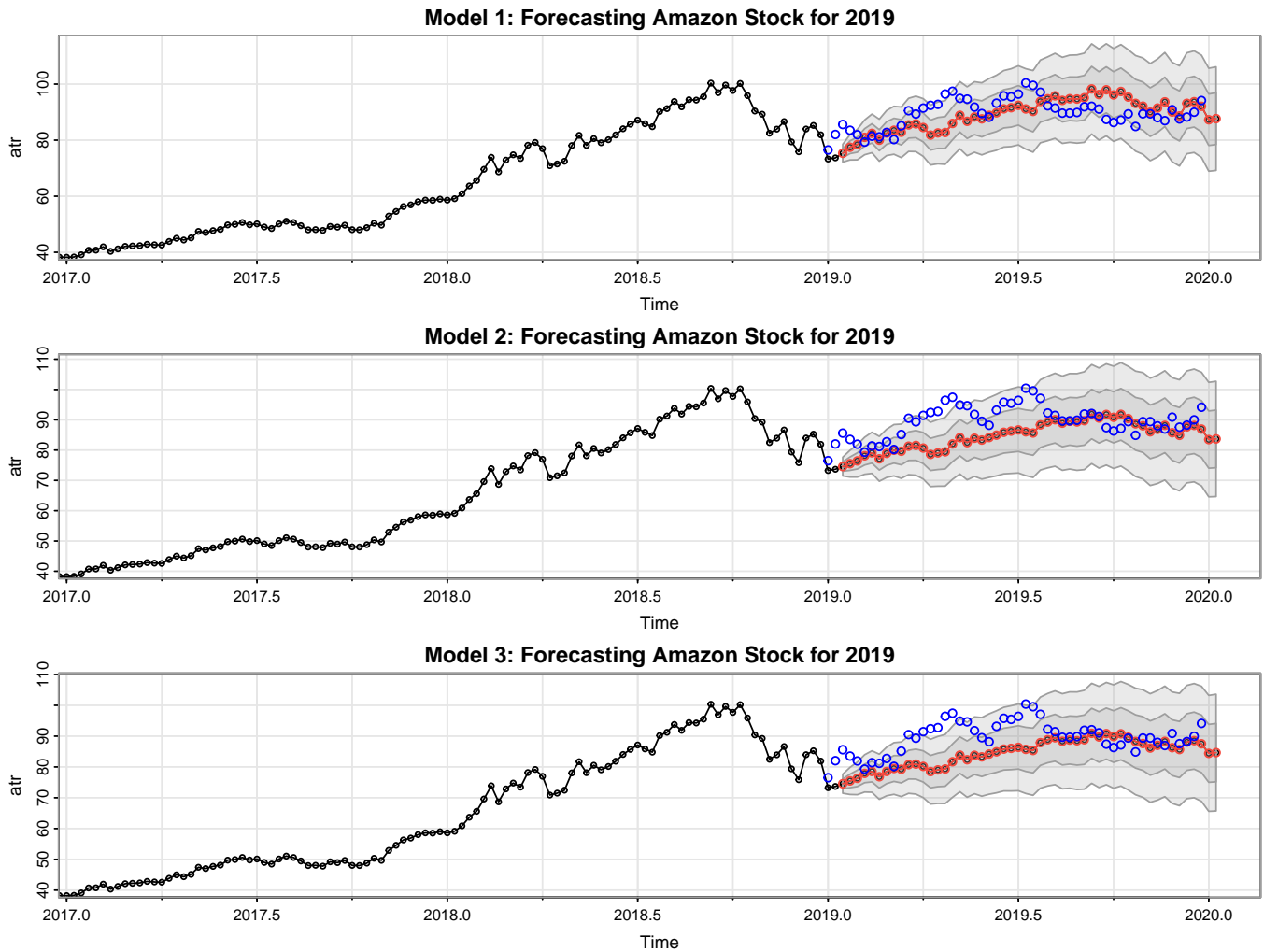


**Model 3: Forecasting Nvidia Stock for 2019**









## VIII. Results

**Google:** The SARIMA model applied to Google’s stock data indicated steady and gradual growth without the influence of any seasonal factors. Until the third quarter of 2019, the model closely followed the actual stock prices until the stock prices experienced a more pronounced increase than the model had anticipated. The deviation indicates that it is true that the model caught the trend but it was not quick enough to respond to the rapid change in the market.

**Nvidia:** Nvidia’s stock forecast in its initial years had an impressive track record, having closely followed the actual stock prices. Nevertheless, a deviation from the forecasted path was found in the 3rd quarter of 2019. Even though the model is divergent, its performance was commendable as it was the second best among the analyzed stocks.

**Microsoft:** The forecasts for Microsoft’s stock showed a significant discrepancy in their performance. The model seemed to use the past trend of steady growth as a basis, but this approach did not anticipate the significant price increase that started in January 2019. Such discordance, therefore, points to the model’s inability to promptly respond to the sudden shifts in the market.

**Amazon:** The forecasted stock price for Amazon was the most accurate as the model was able to capture its relatively consistent growth trend through most of 2019. The actual stock values largely resided within the predicted confidence intervals, indicating the viability of the model. This could be indeed the case because of the high stability of Amazon’s stock in comparison to other volatile market subjects.

It is worth mentioning that there are a lot of factors which might have had an impact on the forecasts. The key event was the US-China trade war that heated up in the middle of July 2019, the time when the dataset was ending. The geopolitical development has left the technology sector, which is very sensitive to manufacturing and labor resources from China, in a very difficult situation. These external factors, even though they are not taken into account in our

model, might have also been relevant in the deviations we have observed in our forecasts.

Overall, we can affirm that our models were able to track the overall trends of the market well, but their performance was variable across different companies and they had their limitations in responding to rapid market changes and external geopolitical factors.

## IX. Discussion

This research focused on an extensive analysis of the weekly stock prices of four tech companies that were listed on the stock market for ten years (2010-2020). The primary objective was to evaluate the forecasting power for the year 2019, applying ARIMA models as a main statistical tool.

Our comprehensive analysis model compares the forecasts to the actual stock performances, which vary on the accuracy of the model. Particularly, the stock predictions of Google were in line with the actual data until the third quarter of 2019, a time when the market dynamics were underestimated, as is evident by the significant rise observed. Nvidia's forecast and the actual trends were in the same direction until the approximate period when the prediction went off track. Nevertheless, Microsoft's stock faced the most difficult problem as the model did not replicate the rise during 2019. However, among all the forecasts, Amazon's stock forecast turned out to be the most precise, where the predictive model successfully followed the price pattern of the company throughout the year.

These findings suggest that although ARIMA models are useful in extracting such information, the precision of their forecasts is not consistent for all the stocks. Amazon is a probable choice for investors who are looking for a reliable technology sector-based investment. However, such disparity found in Google and Microsoft, to some extent, shows the difficulties that exist even in stock market forecasting, particularly in stocks that are more volatile and susceptible to external factors.

This analysis demonstrates the usefulness of statistical models in financial forecasting, and the fact that they have their limitations in highly dynamic industries such as technology. The divergence patterns are a reminder that a wider range of factors, such as economic, political, and global health events, ought to be considered in future forecasting activities.

Finally, although the precision of our ARIMA models didn't completely meet our expectations, they still gave us an idea about stock market behaviors, which is proof that sophisticated analysis is important in making the right investment decisions. More advanced research can add more parameters to the model to consider a wider range of the factors influencing the market, which were not included in this study. This may help improve the accuracy and reliability of stock market forecasting.

## References

1. Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4th ed.). Springer.
2. Yahoo Finance. (2023). Microsoft Stock History. Retrieved February 20, 2023, from <https://finance.yahoo.com/quote/MSFT/history?p=MSFT>
3. Yahoo Finance. (2023). Nvidia Stock History. Retrieved February 20, 2023, from <https://finance.yahoo.com/quote/NVDA/history?p=NVDA>
4. Yahoo Finance. (2023). Google Stock History. Retrieved February 20, 2023, from <https://finance.yahoo.com/quote/GOOGL/history?p=GOOGL>
5. Yahoo Finance. (2023). Amazon Stock History. Retrieved February 20, 2023, from <https://finance.yahoo.com/quote/AMZN/history?p=AMZN>