Forecasting Stock Prices with Seasonal ARIMA

Mykhailo Glushchenko

Abstract

This paper aims to address the feasibility of Seasonal ARIMA models being used to forecast daily stock prices. In the beginning, we split our data 4:1 for training and testing. We use differencing techniques to remove seasonality and trend from the chart, and autocorrelation and partial autocorrelation plots to estimate potential model parameters. The corrected Alkaline Information Criterion is used to decide the top three candidate models. The best model is decided by residual normality and independence, as well as how well the model forecast fits the test data. We use the residual histogram plot, residual qq-plot, as well as the Shapiro-Wilk test to check the normality of residuals the model produced. The Box-Pierce and Box-Ljung tests are used to check independence of residuals. If a good model is found, it is then re-trained on the full dataset to make an out-of-sample price forecast.

The paper concludes that Seaonal ARIMA is successful at making short-term forecasts for a select few stock charts, but greatly deviates from truth when trying to make long-term predictions for all charts. We conclude that the time series approach only works for stock charts that do not exhibit sharp changes in behavior throughout their price history ($GOOG, and $SPY being examples of successful model fitting, $AAPL being an example of a decent model, and $TSLA being an unsuccessful example of Seasonal ARIMA use).

1. Introduction

This paper attempts to model several large-cap stock charts with a Seasonal ARIMA model, explores how well or poorly prices can be predicted with said model; the paper then tries to forecast the tickers' movements for the next few months. Financial markets are a major field in our lives due to the amount of individuals greatly involved in participating within the financial system. It is thus imperative to be able to understand, both in the short and long term, the potential movements stocks and indices may experience. We attack this question by exploring the daily open prices of $AAPL, $TSLA, $GOOG, as well as the ETF $SPY for the last 18-20 years. To acquire data, we use Python to scrap the TD Ameritrade API, storing the data in structured JSON files. We then use R to graphically represent the data we are working with, perform time-series model-fitting (using ACF and Partial ACF plots for guidance) on each chart, run diagnostic tests on top-contending models (including checking normality and independence of model residuals), and make predictions. Finally, we summarize all findings using LaTeX. The paper concludes that Seaonal ARIMA successfully makes short-term predictions for a number of charts, but fails to deliver good results on long-term forecasts. Furthermore, we conclude that Seaonal ARIMA is not a great model to use to forecast stock prices, but that it can be useful for short-term estimations. Finally, we note that there exist charts where a Seaonal ARIMA model is not a good predictive model for any time horizon.

2. Initial Time Series Analysis

First, for comparison, let's examine all the time series we will be working with side-by-side:

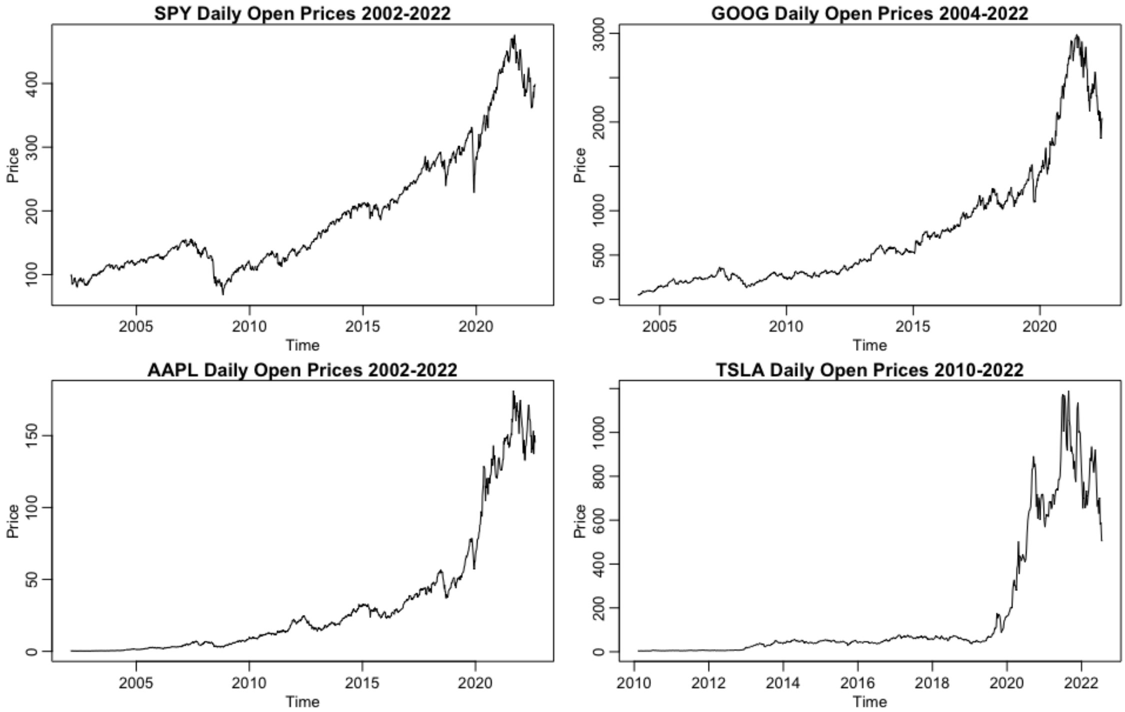


Figure 1. Initial Time Series Charts

We can see a trend is present in all 4 charts, and 3 of the 4 charts exhibit seasonality. Tesla seems to not have a seasonality parameter to it and behaves differently from the other charts. We will analyze each chart one-by-one, and put everything together when making forecasts.

2.1. SPY

2.1.1. Train-Test Split

We begin by splitting the SPY data into a train and a test set, with a 4:1 ratio:

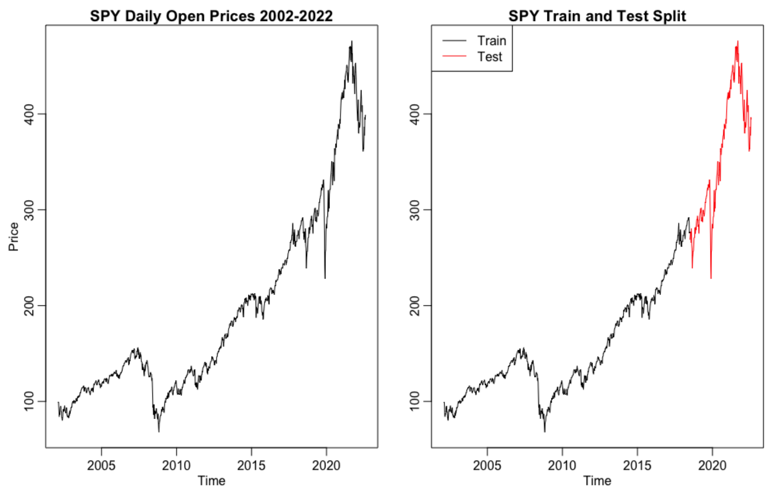


Figure 2. SPY Train Test Split

2.1.2. Initial Commentary

SPY exhibits a clear upward trend, as well as a frequent seasonal component. We can also see a single sharp change in behavior, which happened during the 2008 financial crisis. We proceed by applying transformations to the original time series.

2.1.3. Data Transformation

The chart shows exponential growth, so we will try to use a log and a boxcox transformation. After trying a boxcox transform on SPY, we get a lambda of 0, meaning it's best to try to use a log transformation here. Here's how the data looks when transformed:

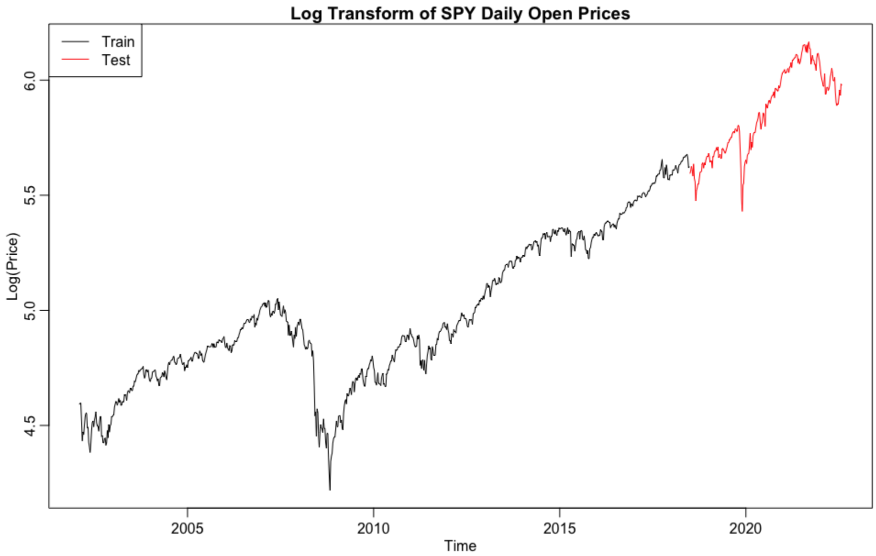


Figure 3. Log Transform of SPY Prices

We can now get more insight about the data by examining its ACF and PACF plots:

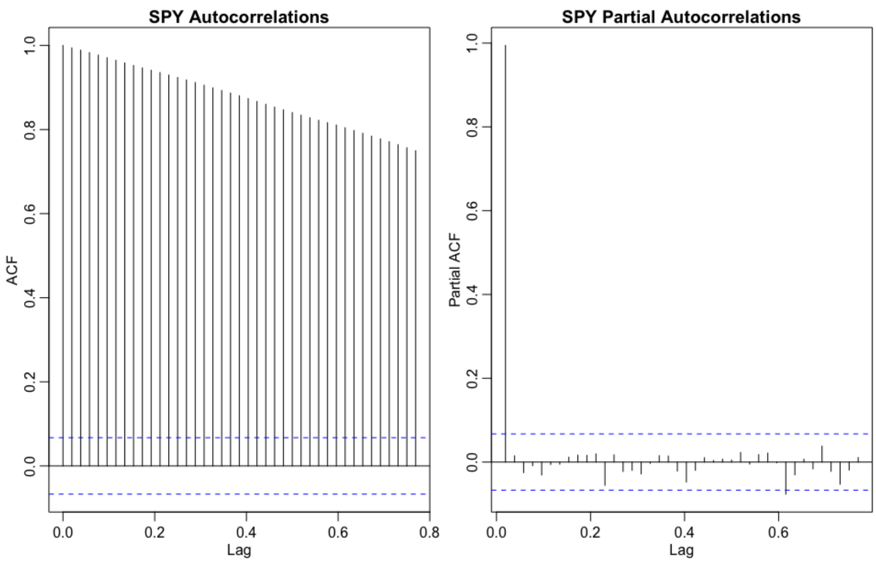


Figure 4. ACF and PACF of SPY

2.1.4. Making Data Stationary

The PACF chart hints that there is a seasonal period of 10 within our data, and that it may be useful to difference our data by order of 10 to remove seasonality. We will also apply a differencing of period 1 to remove trend from the chart. Before committing to the above strategy, we check whether a second differencing is necessary by looking at the variance of each time series:

```
> seaonal_term <- 10
> y10 <- diff(train_spy_log, seaonal_term)
> y10.2 <- diff(y10, seaonal_term)
> var(y10); var(y10.2)
[1] 0.004466301
[1] 0.009118835
>
> # difference to remove trend.
> par(mfrow=c(1,1))
> y1 <- diff(y10, 1)
> var(y10); var(y1)
[1] 0.004466301
[1] 0.001123792
```

Figure 5. Comparing Variances of De-trended Time Series

The output above shows that we get a smaller variance by doing a single differencing of order 10, followed by a differencing of order 1. The result is the following stationary time series:

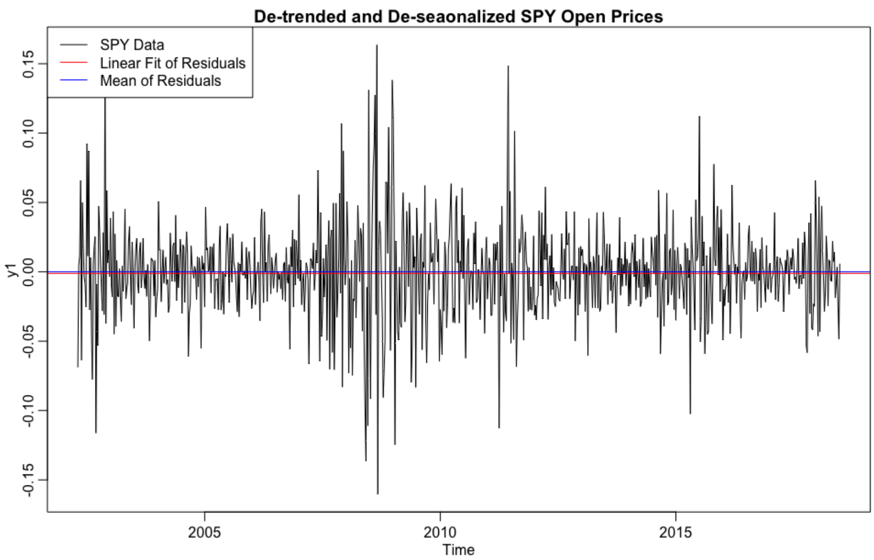


Figure 6. De-trended and De-seasonalized SPY Data

Since the linear fit line (red) in Figure 6 is horizontal, trend has been eliminated. We can see the mean is almost zero (as indicated by the blue horizontal line), and that, for the most part, the variance stays constant throughout. There are some fluctuations that have to do with volatile trading periods, but, for the most part, the data looks good to proceed. The next step is to identify candidate model parameters by inspecting the ACF and PACF plots of the de-trended time series.

2.1.5. Parameter Estimation

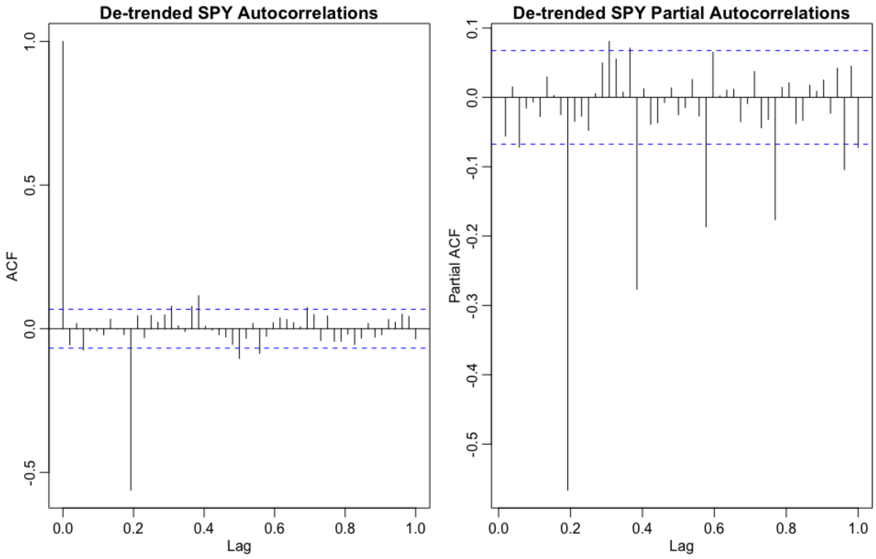


Figure 7. ACF and PACF of De-trended SPY Data

- The ACF chart on the left tells us potential values for q could be 0 or 3.
  - Here, $q = 3$ is somewhat unlikely because the ACF at lag 3 is barely significant, but it's worth taking note of the possible parameter.
- The same ACF chart shows that $Q = 1$ should be a parameter of our model.
- The PACF chart on the right shows that p could be 0 and 3.
  - Once again, $p = 3$ is unlikely to be a good parameter because of how little the PACF at lag 3 is significant by, but it's worth taking note of it.
- Finally, the PACF chart also shows that P could be 0, 1, 4, or 5.
  - We test $P = 4$ and $P = 5$ because the PACF at lag 5 is significant but not extremely decisve-looking.

We proceed by fitting all parameter combinations to our now-stationary data, and calculating their corrected Alkaline Information Criterion for comparison. We will try to fit the 3 models with the lowest AICc. Here's the table describing the parameters of those models:

p	q	P	Q	AICc
3	0	4	1	-3889.379
0	3	4	1	-3889.378
0	0	4	1	-3888.261

Table 1. Lowest AICc SARIMA Models for SPY

From the above table, we can see that the AICc metric agrees with our initial impression that $p = 3$ and $q = 3$ will not be too helpful in improving the model. The initial impression is that the simple SARIMA$(0, 1, 0)(4, 1, 1)_{10}$ will be the best model, since a simpler model is usually better than a more complex model when the results of the two models are similar.

2.1.6. Model Diagnostics

We will compare the success of the 3 models by performing tests for residual normality and independence. The first model we have is a SARIMA$(3,1,0)(4,1,1)_{10}$. First, we check the plot of the residuals for this model:

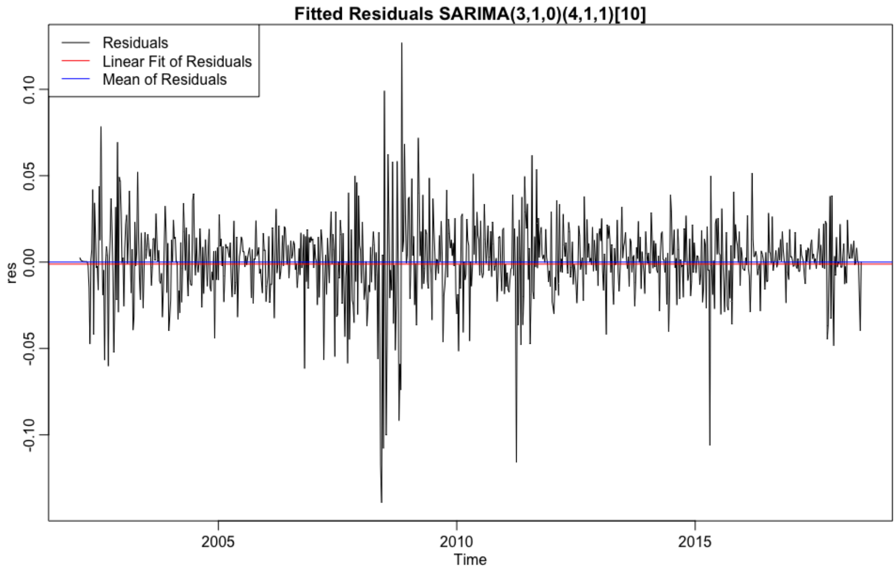
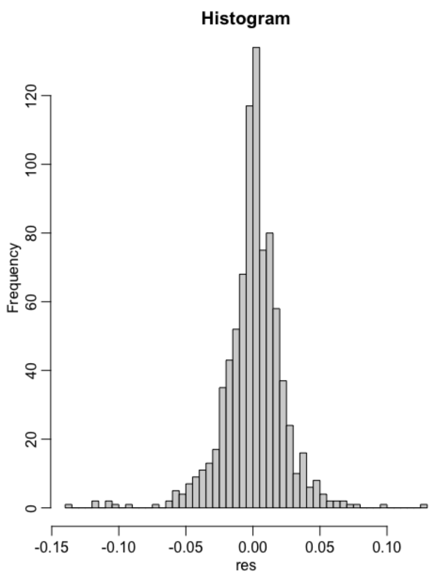


Figure 8. SARIM$(3,1,0)(4,1,1)_{10}$ Residuals

We can check normality of the residuals by looking at its histogram and qq-plot, as well as performing a Shapiro-Wilk normality test:



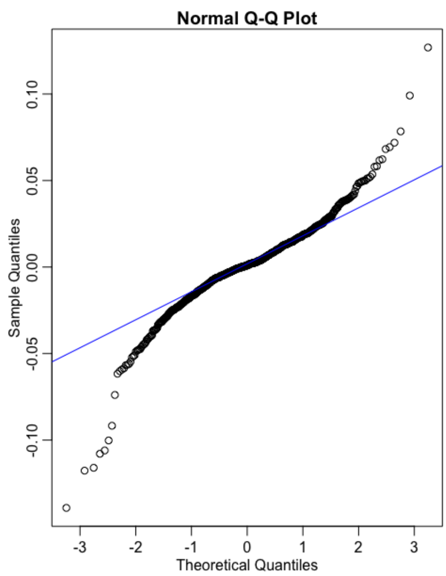


Figure 9. Residuals Histogram and QQ-Plot

Since the QQ-plot is curved on its ends, it is inconclusive in telling us whether the residuals are normal; most of the residuals are normal, but there are some outliers skewing the results. The histogram displays a good bell-shaped curve, but we can still see that the tails are different from that of a bell-shaped curve, as they become too small too quick and extend further than they should. Overall, most of the residuals seem to be normally-distributed. That being said, the model does not pass the Shapiro-Wilk normality test:

```
> shapiro.test(res)

        Shapiro-Wilk normality test

data:  res
W = 0.85905, p-value = 5.243e-12
```

Figure 10. Model 1 Shapiro Wilk Normality Test

With most datasets, this would lead to model re-evaluation, where we try different seasonality and differencing parameters; however, our data will naturally have many outliers during volatile trading periods, and thus this test will usually fail when working with stock market time series. In fact, all models we tried to fit to $SPY did not have normal residuals. Thus, we proceed with trying to use the model to forecast prices, keeping residual non-normality in mind. To check for independence of the residuals, we can look at their ACF and PACF plots:

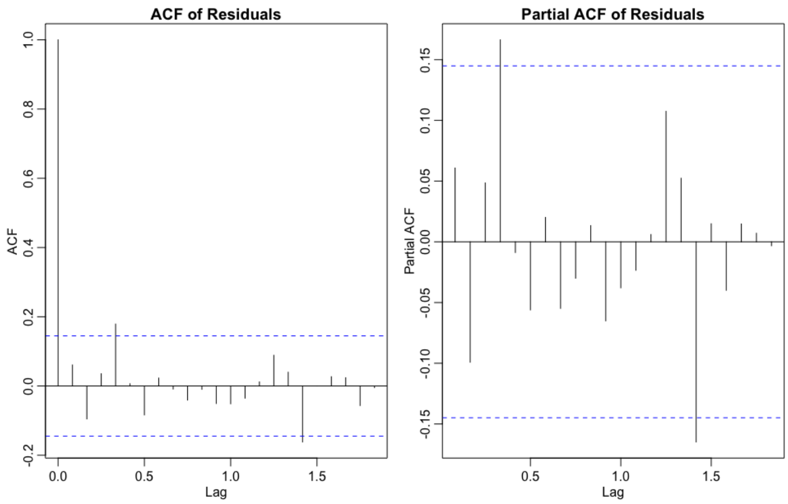


Figure 11. Residuals ACF and PACF Plots

For the most part, the ACF and PACF values are insignificant. To be sure of residual independence, we can perform the Box-Pierce and Box-Ljung tests, and see that they pass:

```
> Box.test(res, lag = floor(sqrt(length(res))), type = c("Box-Pierce"), fitdf = 3)

        Box-Pierce test

data:  res
X-squared = 10.867, df = 10, p-value = 0.368

> Box.test(res, lag = floor(sqrt(length(res))), type = c("Ljung-Box"), fitdf = 3)

        Box-Ljung test

data:  res
X-squared = 11.252, df = 10, p-value = 0.3383

> Box.test(res^2, lag = floor(sqrt(length(res))), type = c("Ljung-Box"), fitdf = 3)

        Box-Ljung test

data:  res^2
X-squared = 9.6401, df = 10, p-value = 0.4726
```

Figure 12. Residuals ACF and PACF Plots

We do the same model diagnostics on SARIMA$(0,1,3)(4,1,1)_{10}$ and SARIMA$(0,1,0)(4,1,1)_{10}$, and conclude that the residuals are both normal and independent for those 2 models as well. In fact, all plots for each of the 3 models were very similar, and led to similar conclusions. Because the model diagnostics did not look better for the more complex models, we choose our best model to be the simpler SARIMA$(0,1,0)(4,1,1)_{10}$. The model formula is:

$$(1+0.0284B^{10}-0.1055B^{20}-0.0107B^{30}+0.0468B^{40})Y_t=(1-B^{10})Z_t, \quad Z_t\sim\mathcal{N}(0,0.0005422)$$

2.1.7. Forecasting Prices

We now use the R predict function to get the sought-after forecast values, and transform them back using the exp function. First, we try to make a short-term forecast – here's the resulting plot:

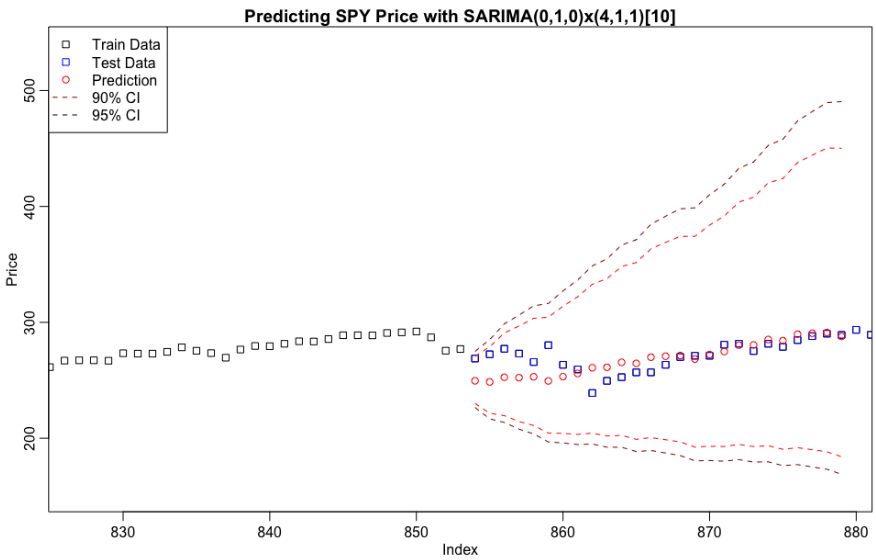


Figure 13. Short-Term (6 Months) Forecast of SPY Prices

Although the confidence intervals are too wide to use this model for actual trading, the forecast results are quite good. Let us now attempt to make a longer forecast, to see if the model adapts to the chart properly:

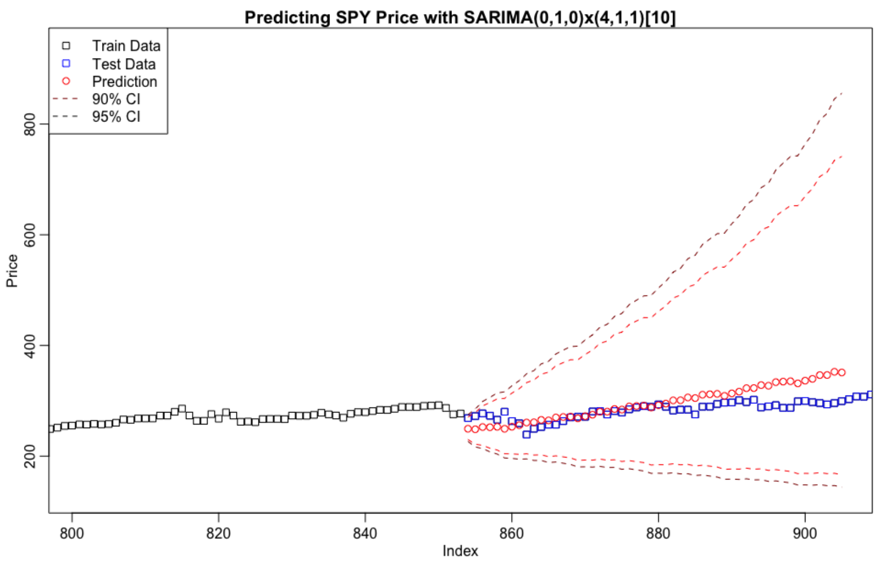


Figure 14. Long-Term (2 Years) Forecast of SPY Prices

At longer time-frames, the confidence interval becomes much too wide for the model to be of any use. Not only is it too wide, it also favors the upside more than the downside, which can lead to heavy losses if the interval was used for trading and managing risk. Furthermore, we can see the prediction starting to deviate from the true prices as the forecast interval becomes larger. In any case, since the short-term forecast seems to be accurate, we can use the model to forecast the next couple months of SPY prices:

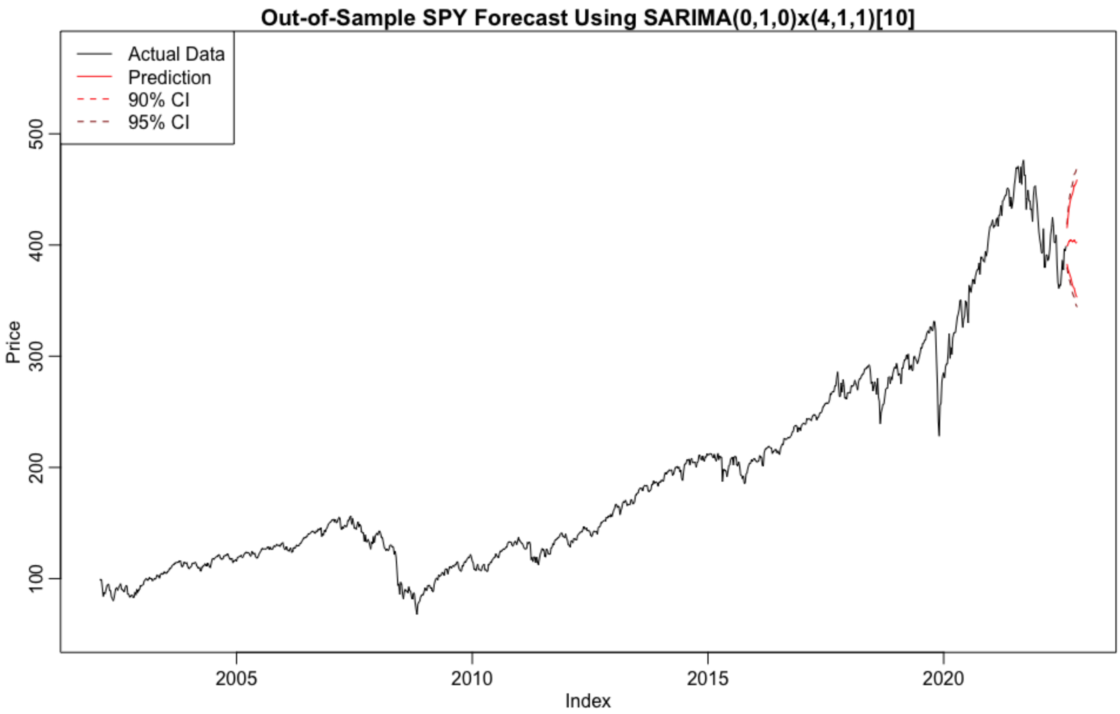


Figure 15. Short-Term Out-of-Sample Forecast of SPY Prices

2.1.8. Concluding Thoughts on SPY

Although some deviations were present, overall, Seasonal ARIMA was a decent model to use to make a near-term forecast of SPY prices. One problem to note was our inability to find a model with normally-distributed residuals, mainly due to the fact that the chart has a couple spots where sharp changes in behavior occur (2008, 2020, etc.). While the confidence interval is too wide to use the model to trade SPY in the market, the model can be used to give a good approximate price for the coming week(s); the model confidence interval can, also, be used to manage risk – one can stay on the safer side of their trades by taking into account the possibility of the lower or upper end of the confidence interval being hit by the actual price. The overall conclusion for SPY is that we could not follow all guidelines of time series analysis, but the best model we achieved fit quite well. The model's failure to predict volatile moments during which price movement behavior changes makes sense, since price (not volatility) is the only input we begin with. As a result, using the model to trade is a poor idea; however, the model can be improved by dynamically adapting the standard deviation of a fitted SARIMA model's parameters to changes in implied volatility.

We will now perform the same steps as above to try to fit a model to $GOOG, $AAPL, and $TSLA, and compare how well each model behaves relative to its test set. Since we already went into detail of how a model is fit to data when working with $SPY, we will fast-forward to the resulting models of the other stocks, noting any differences in model-fitting and model diagnostics between the tickers.

2.2. GOOG

Since Google recently underwent a 21:1 share split, we first adjust google prices by converting the post-split prices to pre-split prices by multiplying them by a factor of 21. The main thing to note about Google's chart, is that, unlike $SPY's chart, there are no sharp changes in price movement. This may result in a more accurate forecast moving forward. Looking at the $GOOG price chart below, we see a clear trend, and we speculate that there might be a seaonal component present on some level:

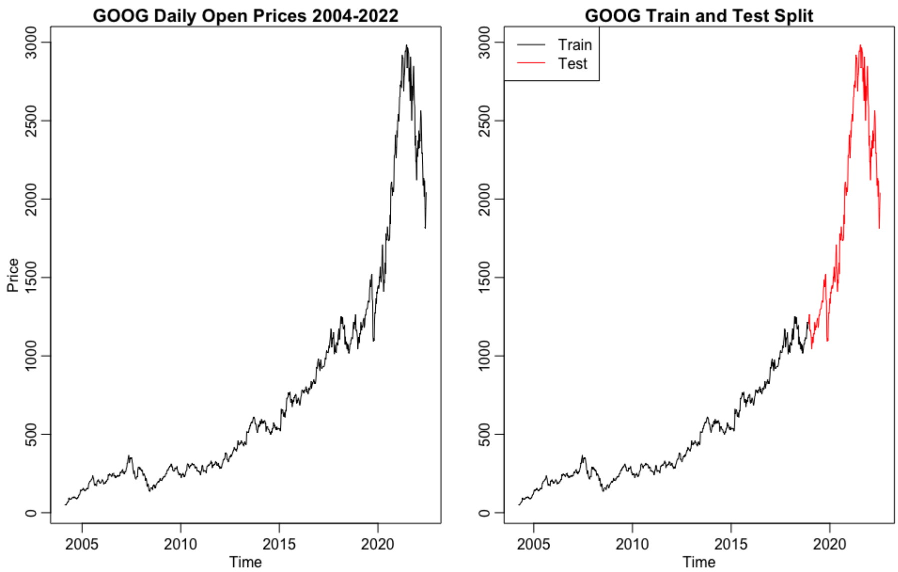


Figure 16. Google Train-Test Split

One lesson to learn from the last chart is that it was difficult to see the seaonality parameter when using weekly prices. Therefore, we first change google prices to be monthly, moving on to analysis after that (the shape of the chart and its behavior remain the same after the re-frequencing). Performing the same time series analysis techniques on this chart as we did with $SPY, we get a different SARIMA model from the one we got for S&P500, but one that seems to work very well:

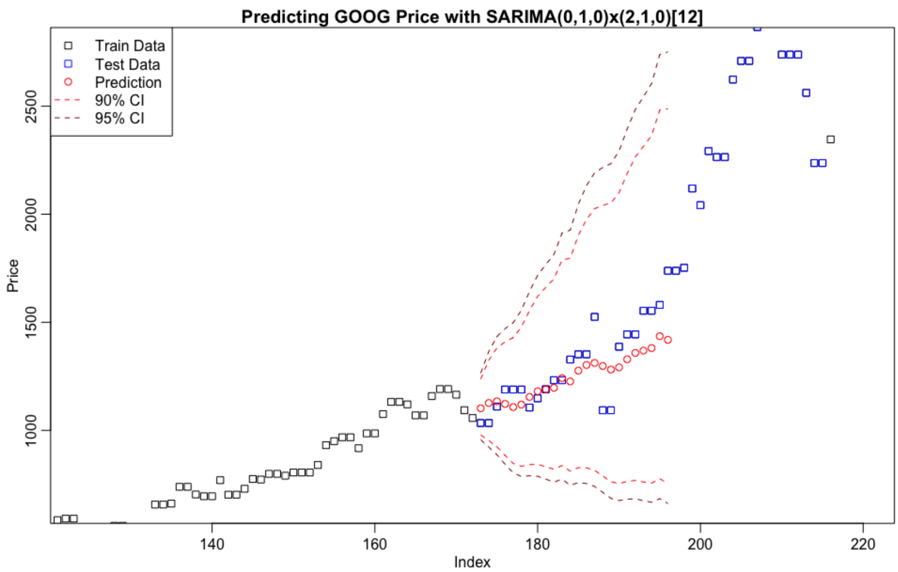


Figure 17. $GOOG Price Forecast Using SARIMA$(0, 1, 0)(2, 1, 1)_{12}$

Here's the SARIMA$(0,1,0)(2,1,1)_{12}$ model equation that was used to forecast Google prices:

$$(1 - 0.0739B^{12} - 0.0307B^{24})Y_t = (1 - 0.9089B^{12})Z_t, \quad Z_t \sim \mathcal{N}(0, 0.08235)$$

We note that we were, once again, unable to make residuals normal for any models fit to Google data. As with $SPY, we can see that the short-term prediction looks quite good, but the forecast starts to deviate from true prices for longer forecast timeframes. Nevertheless, the true prices remain within our confidence intervals, which is good to see. Similar to what we did with $SPY, we now retrain the model on the full dataset, and use it to make out-of-sample price predictions for $GOOG monthly prices:

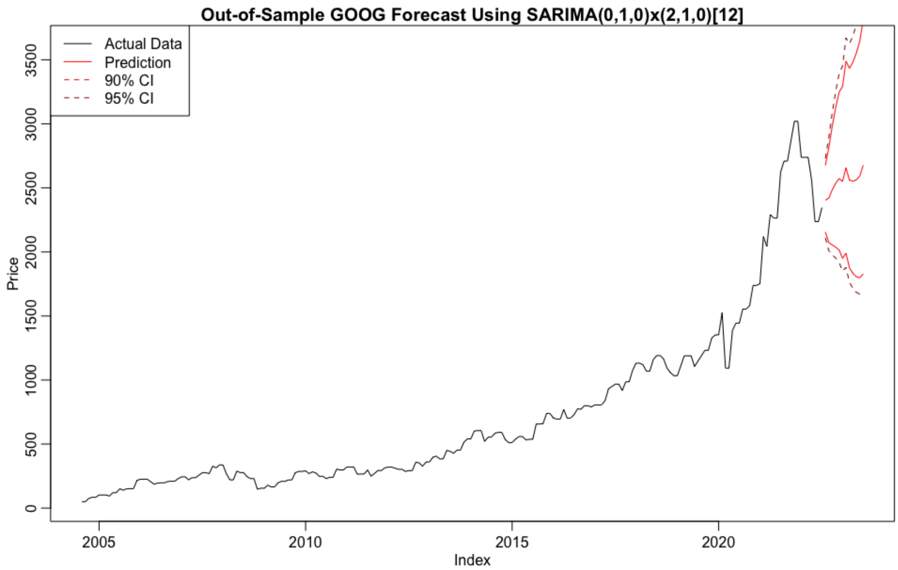


Figure 18. Out-of-Sample $GOOG Price Forecast Using SARIMA$(0,1,0)(2,1,1)_{12}$

To conclude the analysis of the Google chart, we note that the process of analyzing this time series and the previous one were nearly identical, with residual independence tests all passing, and residual normality tests all failing. We can start to see that although the SARIMA model may predict short-term prices well, it is not a great model for price forecasting in the stock market.

2.3. AAPL

As was the case with $GOOG, we can see that the $AAPL chart below does not have any sharp changes in behavior, and therefore it might be more reasonable to use a SARIMA model on this chart than the $SPY one:

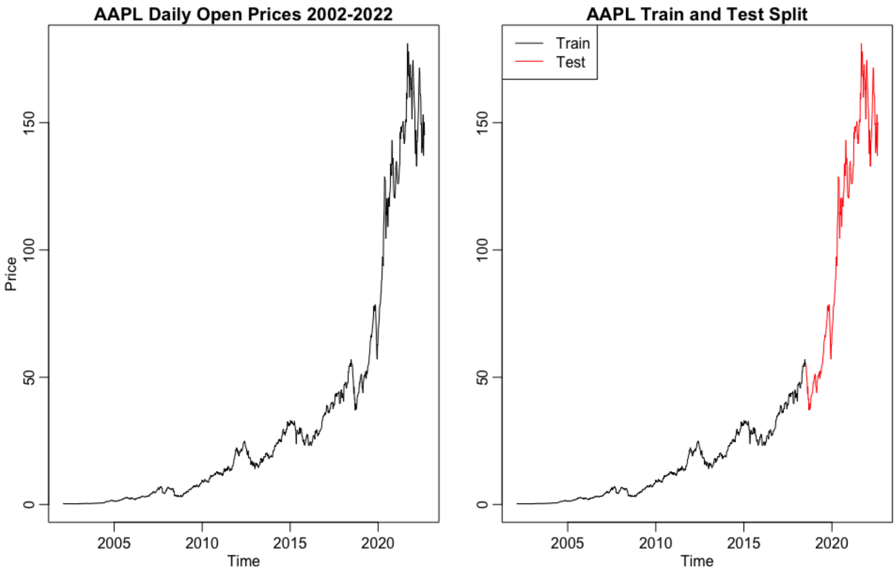


Figure 19. Apple Train-Test Split

The re-frequencing technique we used with Google seemed to help us when dealing with the seasonality parameter, so we will do the same for Apple, predicting its monthly (rather than weekly) prices. After analyzing the $AAPL chart, we come up with a SARIMA$(2,1,2)(1,1,1)_{10}$ model with the following equation:

$$(1+0.1772^{10})(1+0.0172B+0.7262B^2)Y_t = (1-0.9140^{10})(1+0.0016B+0.9720B^2)Z_t, \quad Z_t \sim \mathcal{N}(0, 0.06693)$$

The above equation is then used to make the following short-term forecast of Apple monthly prices:

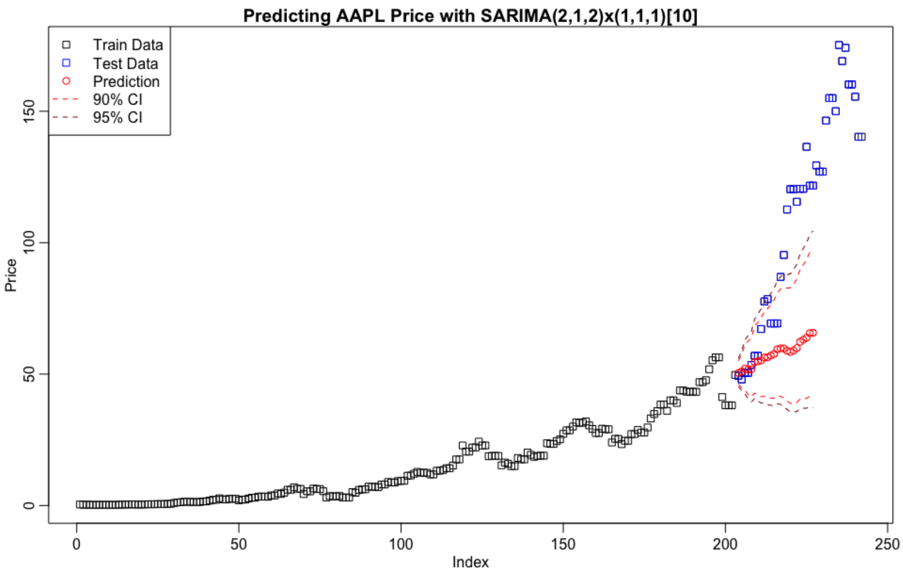


Figure 20. Apple Price Forecast with SARIMA$(2,1,2)(1,1,1)_{10}$

From the above, we can see that this forecast's confidence interval does not fully capture the true price movements ahead. We can zoom in on the chart to get a better idea of how well the prediction was:

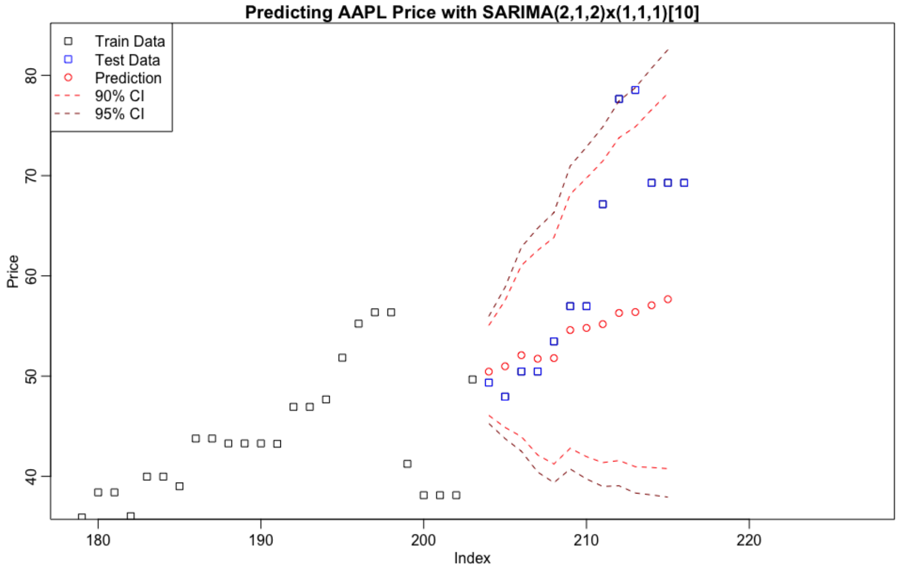


Figure 21. Apple Price Forecast with SARIMA$(2,1,2)(1,1,1)_{10}$

Zooming in on this prediction, we can see that the first 6 months of the forecast give good results, but the model starts to deviate from true prices as the forecast becomes more long-term; with Apple, the forecast seems to be valid for a smaller amount of time as compared to the previous two. Since the first couple months seem to be predicted fairly well, we will still make an out-of-sample forecast for Apple, although one must be weary of this result especially, as compared to previous forecasts:

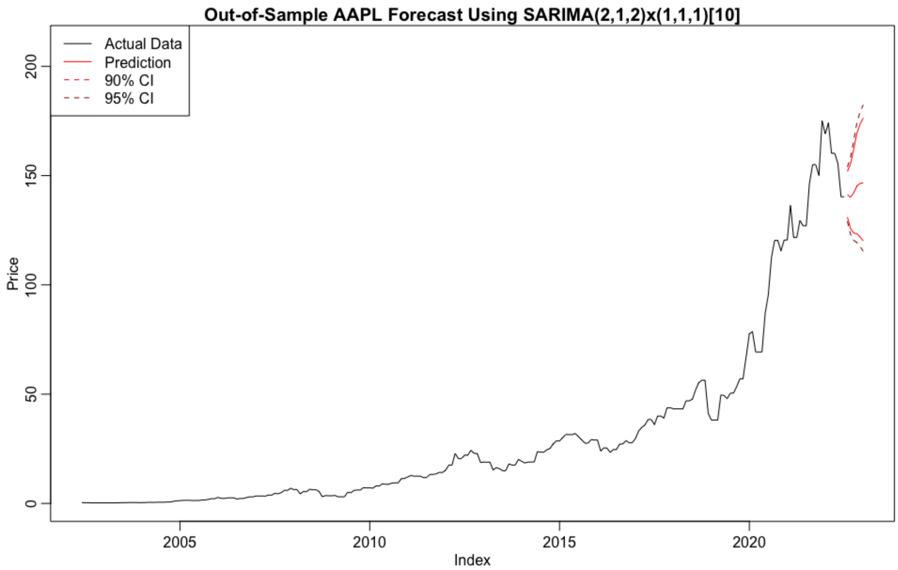


Figure 22. Out-of-Sample Apple Price Forecast with SARIMA$(2,1,2)(1,1,1)_{10}$

Next, we will try to fit a model to the Tesla chart, and see whether its obscure-looking price action can be modeled in any way.

2.4. TSLA

First, we can take a look at the Tesla chart itself:

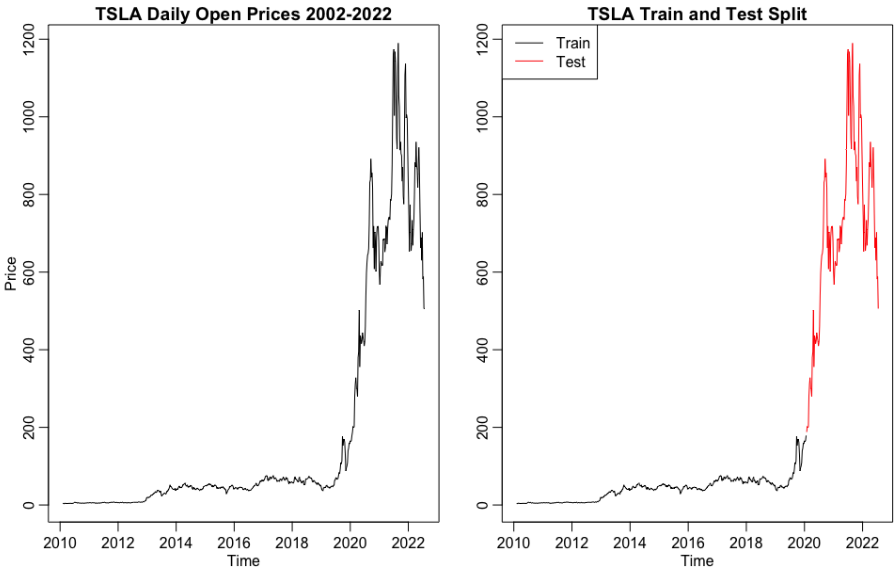


Figure 23. Tesla Train-Test Split

After applying the same time series analysis techniques as before, we get the following SARIMA$(0,1,0)(1,1,1)_{12}$ model as a result:

$$(1 - 0.0406B^{12})Y_t = (1 - 0.7905B^{12})Z_t$$

This model seems to completely miss the target prices with its forecast:

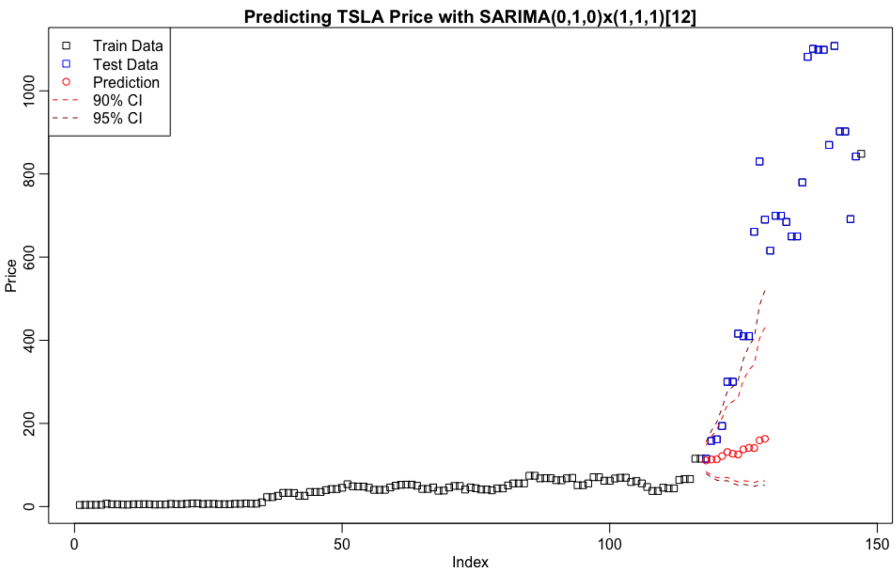


Figure 24. Tesla Price Forecast with SARIMA$(0,1,0)(1,1,1)_{12}$

The reason the SARIMA model did not work for this chart is that the training set happened to be the (mostly flat) price action before the sharp rise of Tesla. Naturally, the forecast is not aware of the testing set's sharp change in behavior, and thus the forecast is completely off from the beginning. This chart reminds us that although the first 3 charts had decent short-term forecasts, a sharp change in the behavior of any stock is possible at any moment, and the predictions are thus not good to use for trading purposes. As stated previously, adding volatility into this model would help with catching the change-in-behavior moments of each chart, and could thus be used to improve the model.

3. Conclusion

The original goal of this paper was to determine whether Seasonal ARIMA is useful in modeling stock prices for both common and non-common-looking charts. We were successful in determining a good proper model for $SPY and $GOOG, and an alright-working model for $AAPL, but we failed to fit a good SARIMA model to the $TSLA chart. It is important to note that normality of residuals was not able to be established for any of the models or charts, which implies that Seaonal ARIMA may not be a good model to use for predicting any stock prices. Besides this assumption, the paper does demonstrate how to successfully make (risky) out-of-sample forecasts of stock price action based on the time series techniques outlined above.

We once again want to reiterate the models we successfully fit to $SPY and $GOOG, so here they are, in that respective order:

$$(1 + 0.0284B^{10} - 0.1055B^{20} - 0.0107B^{30} + 0.0468B^{40})Y_t = (1 - B^{10})Z_t, \quad Z_t \sim \mathcal{N}(0, 0.0005422),$$

and

$$(1 - 0.0739B^{12} - 0.0307B^{24})Y_t = (1 - 0.9089B^{12})Z_t, \quad Z_t \sim \mathcal{N}(0, 0.08235).$$

3.0.1. Work Still To Be Done

With that being said, the overall conclusion of the paper is that SARIMA is only good at forecasting stock prices in certain cases, and more work has to be done to determine for which charts it works and for which it does not. The paper found some significant examples of charts where a Seaonal ARIMA model works, and some where it greatly fails to deliver results. Research should be done into the shapes of each chart, and the reasons for why some charts were able to be fit well, but not all. When testing the success of each model, back-testing could be implemented to check whether a dynamically-determined SARIMA model (adapting daily or weekly as new prices are recorded) can successfully catch the changes in trend of each stock. Volatility information for each chart could also be collected and used for several reasons:

1. Variance of each parameter/forecast can be scaled using volatility at that time to give a more accurate assessment of price risk.

2. Volatility could be used separately from the SARIMA model to highlight the times when the SARIMA model is useful (low volatility) and when it is not.

3. Prices could be scaled using volatility, and the new time series can then be tackled via Seaonal ARIMA time series analysis.

4. References

1. Introduction to Time Series and Forecasting, by P. Brockwell and R. Davis, Springer.

    - This textbook was the only source consulted throughout the paper.

5. Appendix

In this section we include the R code used to construct the report above. Because different models were continuously being changed and tested, and because we did it for 4 separate charts, the code was written separately for each ticker analyzed (as opposed to making some functions to work on any data). For this reason, the code is lengthy, and is thus included as a link to a github repository, rather than being pasted into the document itself.