STAT 443 Group Project

Group 21

2022-12-05

Contents

Problem

For decades Microsoft Corporation has been a leading provider for computer software, consumer electronics, computers, and more. However, despite their size and strength, Microsoft was not immune to the array of impacts stemming from the onset of the COVID-19 pandemic. For instance, when stock markets around the world crashed on February 20th, 2020, Microsoft's stock (MSFT) fell with it and continued to descend and fluctuate for the short term thereafter. We thought it would be interesting to forecast what Microsoft's closing stock price could have looked like in the hypothetical situation in which the pandemic never occurred. Moreover, stock data is generally classified as a random walk. This assumes that the stock price is a result of random movements, independent of the past, and that reliable predictions are futile. This is why we are not attempting to forecast MSFT's price for the purpose of optimizing returns in trading. Rather, our goal is that the process of comparing a prediction based on the historical data versus what actually happened will highlight just how true the random walk argument is. This would also serve to demonstrate just how significant the pandemic was on financial markets, at least at first.

Plan

After finding an adequate dataset that includes a measure for Microsoft's stock price over time, we will implement various appropriate modeling techniques to fit our data and evaluate their prediction power. For each model, a training set will be used for fit and then we will compare the predictions this model makes for a chosen test set against the actual data over that time. Thus, the best model will be the one that minimizes the prediction mean squared error (MSE). Using this we can forecast the MSFT price after the market crash and compare this to what actually occurred.

Data

Dataset

We retrieved Microsoft's closing stock price from April 2015 through to March 2021 inclusive from kaggle.com. From this set we decided to focus on the data from 2019 to early 2020 to fit our models and evaluate prediction power. We decided that using a snippet of the data was sufficient since the movement in price (including any sources of non-stationarity) was similar in the data prior to 2019. This also enabled us to see more granularity in the movement of MSFT's closing price.

Training and Test Set

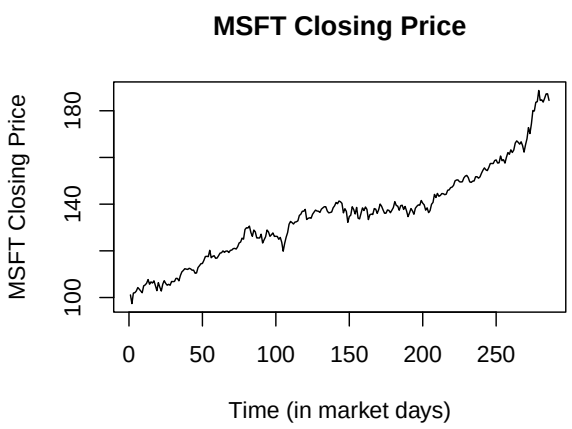
We used all of 2019 up until the month before the crash as our training set and the month between January 21, 2020 to February 20, 2020 as our test set. This translated to a training set of all days when the stock market was open between January 2nd, 2019 to January 17th, 2020 and a test set of all days when the stock market was open between January 21st, 2020 to February 20th, 2020.

Potential Problems

This time series consists of consecutive market days as opposed to calendar days. In other words, the stock market is not open on weekends or holidays and so the closing price for the MSFT stock can not be collected every day of the year. None of the market days within our time period of interest are missing values or contain errors that we can find, thus, this is not a problem. However, careful consideration of what calendar days our fitted values and predictions correspond to may be necessary.

Preliminary Analysis

To get a better sense of how the MSFT stock price behaves, we can look at a time series plot and an autocorrelation plot (ACF) of the data:



Over time it appears that the closing price for MSFT has been steadily increasing. Moreover, we do not see any obvious signs of seasonality. We can also look at the ACF for this data to verify this. The slow decay in the ACF confirms our suspicions that the MSFT closing price only has an increasing, non-periodic trend.

In regard to the variance, it looks roughly constant throughout time, however, there are some periods with larger/smaller variance that are slightly concerning. We can perform Fligner's test on a few different segmentations of the data to test this and determine if a transformation is necessary. In all cases, the Fligner's test produced a very small p-value suggesting that the variance was not constant. However, after trying various power transformations, none of them made significant improvements in the consistency of the variance. The results of these tests can be found in Exhibit 1 of the Appendix. Thus, we decided it was best to leave the data un-transformed, especially since at first glance of the time series we felt that the variance looked roughly constant.

Analysis

Regression

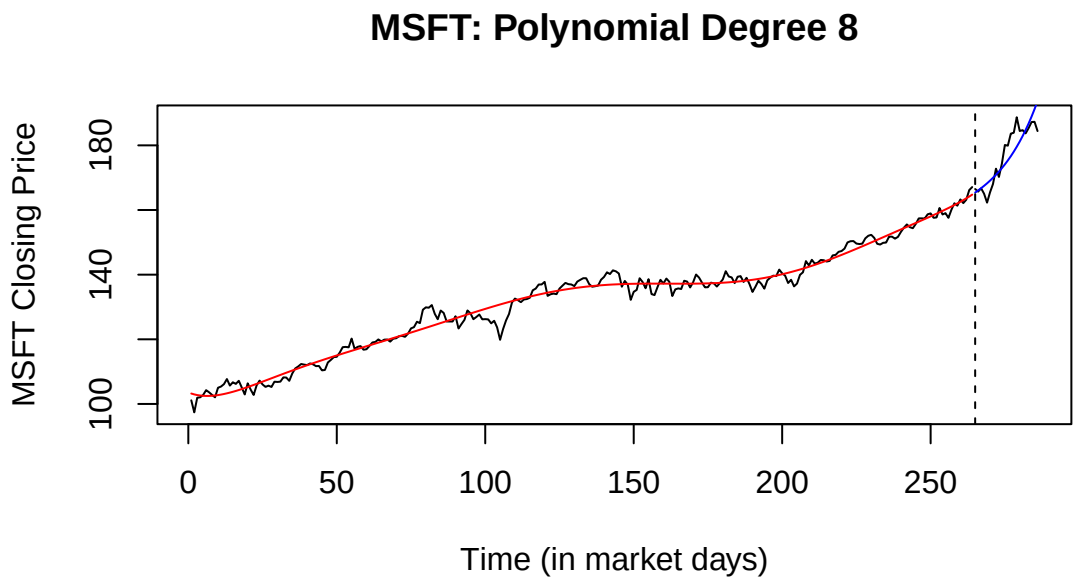
Un-regularized Regression

We started by trying various un-regularized regression models on the data. To avoid large predictors with high correlation, we used orthogonal polynomials. Polynomials of degree 1 to 10 were used to fit the data and chose the best model based on the prediction MSE.

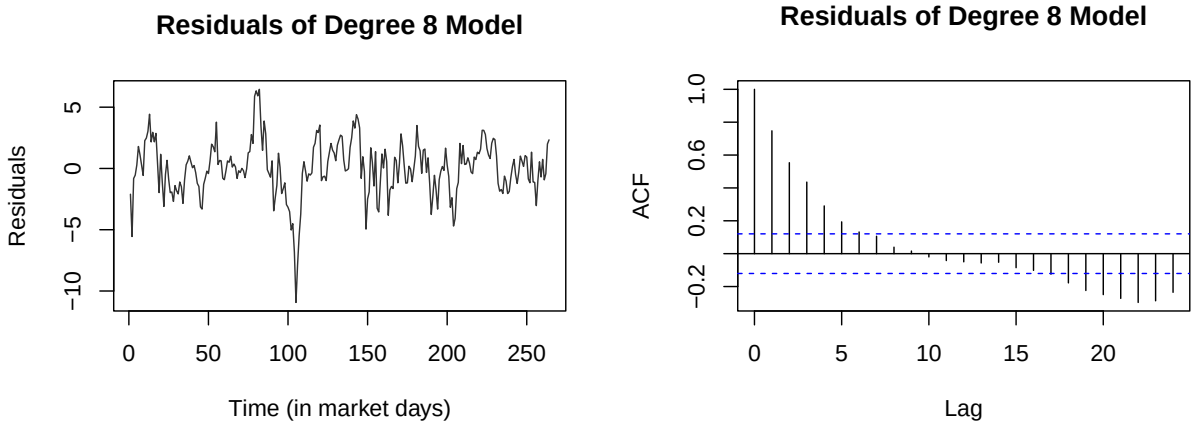
Table 1: MSE for Best Model by Degree

1	2	3	4	5	6	7	8	9	10
318.3771	427.153	106.9403	19.45023	35.13549	231.4563	401.5411	18.74654	3710.62	25.94357

As the table above shows, the test set's MSE was minimized when we used a polynomial of degree 8. We can judge how well this model did by visually checking the fit over the training and test sets as shown in the plot below.



As expected, the regression model fits the training data very well. Thankfully, the fit over the test data is also good. We can now perform model diagnostics by analyzing the residuals below.



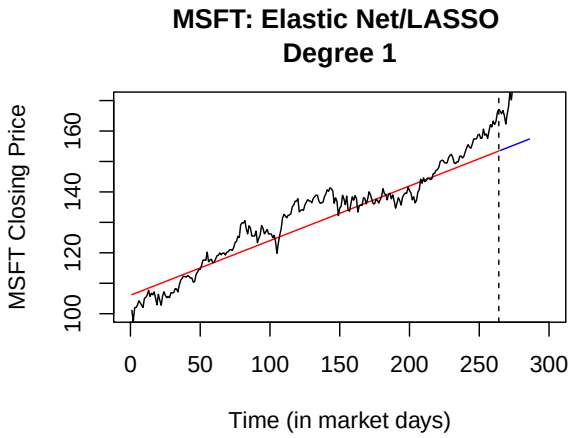
The slow decay in the ACF plot suggests that there is still some trend left in the residuals. It also appears that the residuals do not have constant variance. Thus, these residuals are not stationary.

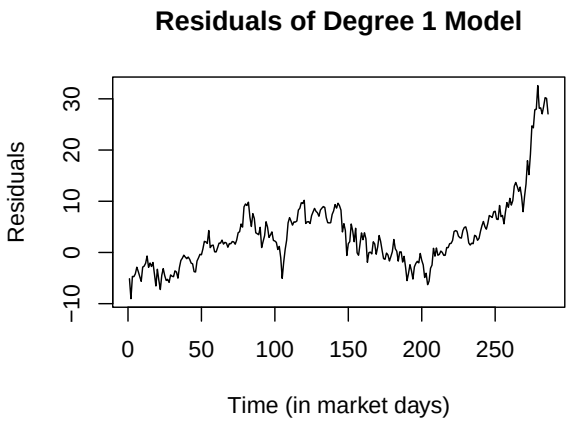
Elastic Net and LASSO Regression

Next, we used regularized polynomials. For each degree of polynomial (1 to 10), we fit a LASSO model and an Elastic Net model with a grid of alpha values by increments of 0.1. We chose the regularization parameter for each model with cross validation, using the value that minimized the cross validation error. For each degree of polynomial, the models were compared based on their prediction MSE. After choosing the best performing model within each degree, we chose the model among all degrees which had the smallest prediction MSE for the test set. In this case that was the degree 1 polynomial.

Table 2: MSE for Best Model by Degree

1	2	3	4	5	6	7	8	9	10
4082.799	4087.701	4180.446	4222.16	4233.172	4235.92	4232.886	4225.355	4241.901	4243.902





Looking at the above plot, regularized regression performs very poorly for the purpose of removing the trend from the data. This makes sense. Since we are shrinking the parameters, we introduce bias into the model in aims of reducing the variance. Because we shrink the parameters, they likely deviate more from the true values, and hence the model underestimates the true values. Moreover, we have created a model with orthogonal polynomials, so the predictors have no collinearity that regularization could help reduce. Also, we did fit the data with ridge regression too, however, the prediction was very poor even with the optimal model. This can be seen in Exhibit 2 of the Appendix. Overall, the use of regularized regression is a poor choice for this data set since it reaps none of the benefits that regularization can provide, but keeps its drawbacks.

Conclusions

Since regularized regression was not a good fit for this dataset, we will proceed by only considering unregularized regression, for which a degree 8 polynomial provided the best fit. However, considering the fact that polynomial fits are generally very poor at predicting outside the range of the training data (due to properties of polynomials), it would be wise to take any extrapolated predictions of the polynomial model with a grain of salt.

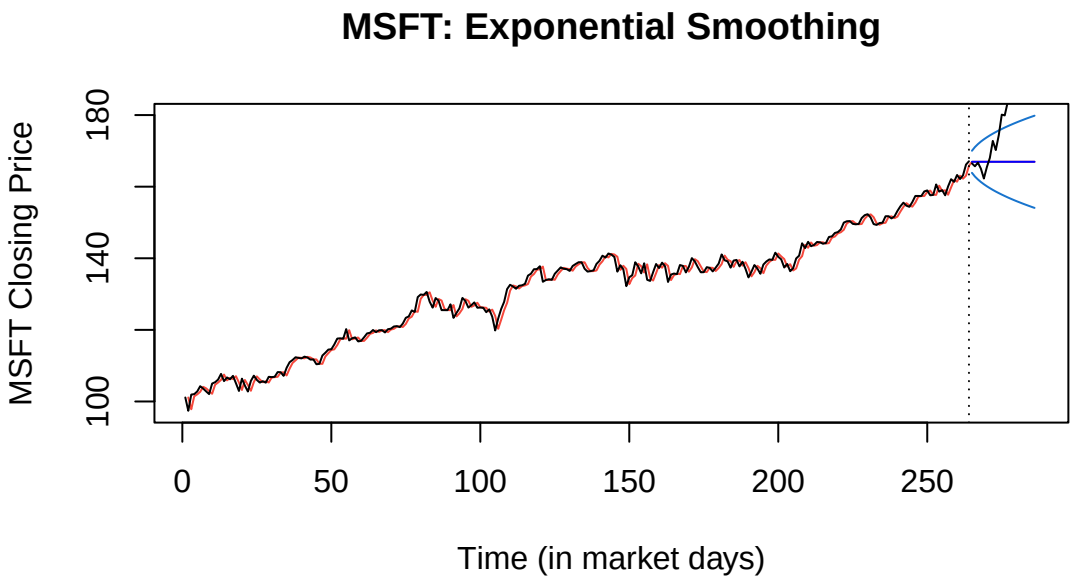
Smoothing Methods

Since we only have a non-periodic trend present, we tried exponential smoothing and double exponential smoothing with the Holt-Winters algorithm.

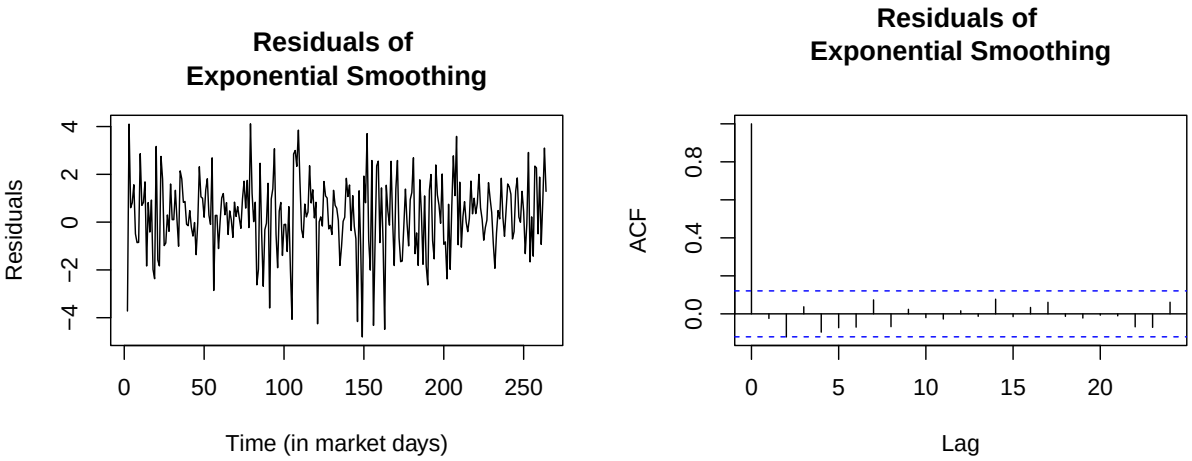
Exponential Smoothing

Under exponential smoothing the fitted values over the training set capture the movement in MSFT's price very well (as expected). However, since exponential smoothing is only able to give us constant predictions equal to the level of the most recent observation, the prediction over the test set does not capture the increasing trend, as seen in the plot below. Based on the results of the Holt-Winters algorithm, we get the

following formula for predictions: $\hat{X}_{t+h} = L_t = 166.95$. Using this to predict over the test set we get a prediction MSE of 176.35.



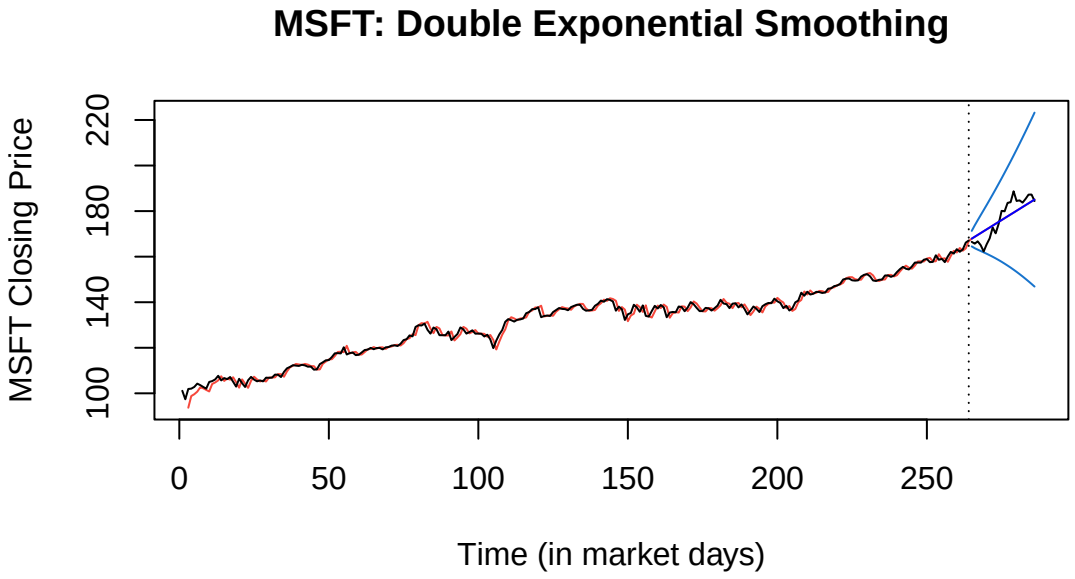
Importantly, when we check the residuals from this modeling technique we can see that the trend has been successfully eliminated. Perhaps the variance is not always constant, but we do not feel that it has gotten worse. We used Fligner's test to check. Using 8 segments of 32 observations (where the last one has 15 extra), under Fligner's test we get a p-value of 0.15. Thus, we have no evidence that the variance is not constant so we can argue that the residuals are stationary. There also does not appear to be any issues with collinearity.



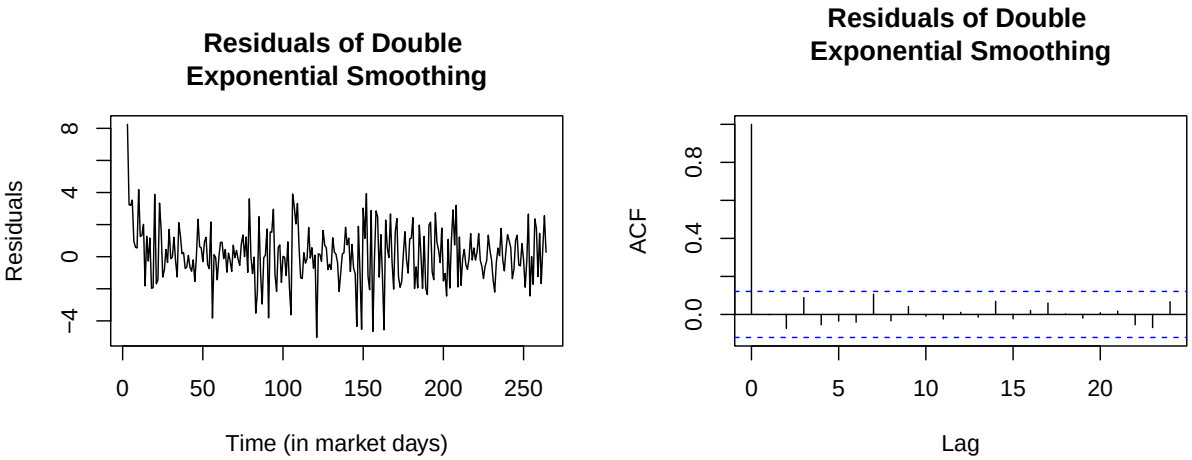
Double Exponential Smoothing

Similar to exponential smoothing, under double exponential smoothing the fitted values over the training set capture the trend and variability of the data well (as expected). However, the prediction over the test

set has certainly improved, as seen in the plot below, and is now able to capture the increasing trend over this period. Based on the results of the Holt-Winters algorithm, we get the following formula for predictions: $\hat{X}_{t+h} = L_t + hT_t = 167.09 + h(0.82)$. Using this to predict over the test set we get a prediction MSE of 20.71.



We can also analyze the residuals from this model to check their stationarity. We get the below results. The trend has been successfully removed and the variance looks relatively constant most of the time, however, it is perhaps not perfect. Since we are not sure about how constant the variance is with the plot alone, let's use Fligner's test to check. Using 8 segments of 32 observations (where the last one has 14 extra), under Fligner's test we get a p-value of 0.03. Thus, we have moderate evidence against constant variance, however, this is a significant improvement compared to the original data so we are satisfied. Thus, we feel that the residuals are stationary. There also does not appear to be any issues with collinearity.

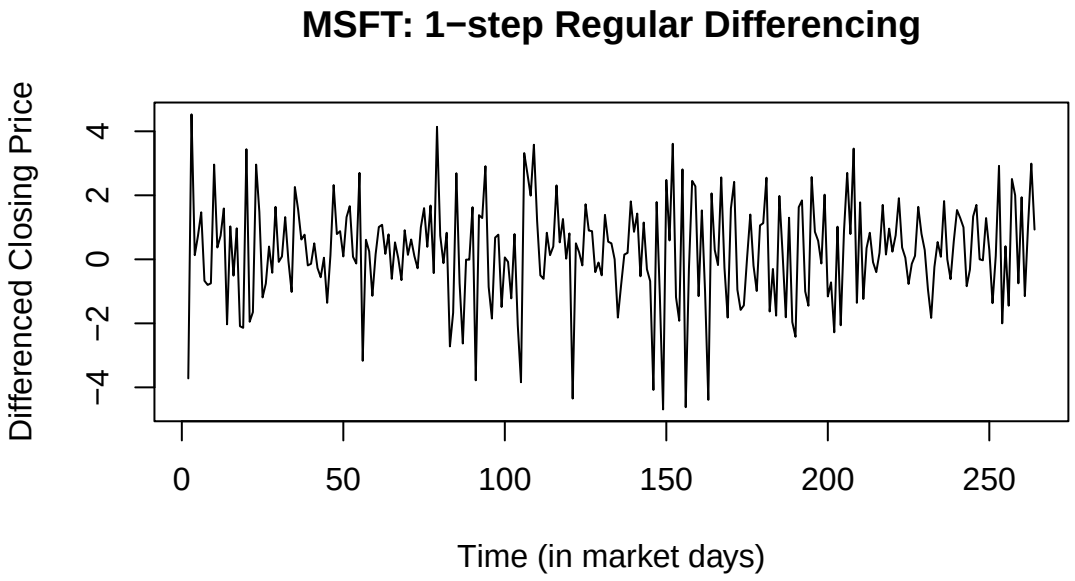


Conclusions

Out of the smoothing methods discussed here, the prediction MSE was minimized with double exponential smoothing. In both cases the results from the residual analysis were nearly identical, thus, we think that double exponential smoothing is the optimal smoothing method.

Box-Jenkin Models

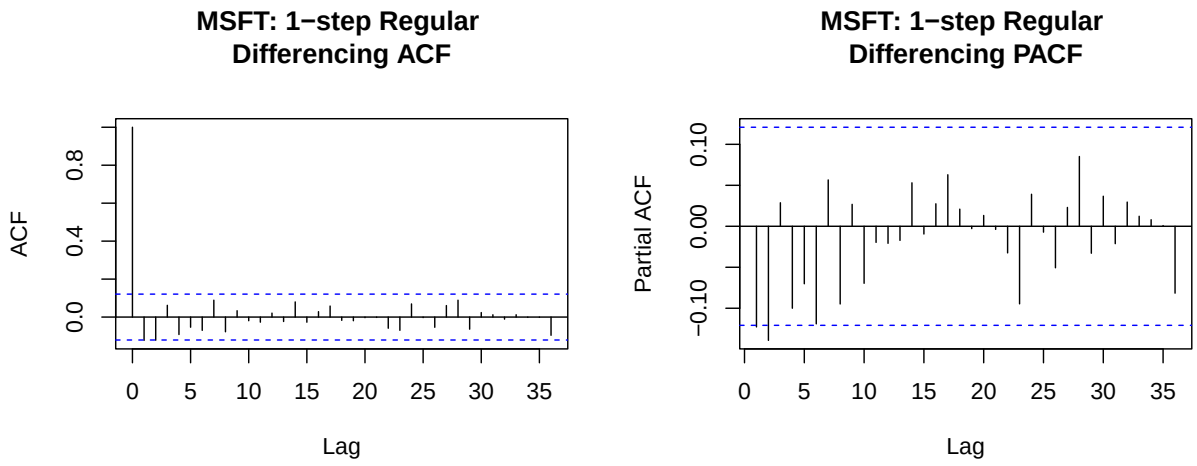
As seen earlier, no transformations were able to achieve constant variance based off of the Fligner's test, so we will be using the original data without any transformations. This is clearly not optimal. Recall that in the original plots, we saw that there was an increasing trend in the data and the ACF confirmed this. In order to eliminate this trend and make the data as stationary as possible, we used one step of regular differencing. A plot of the differenced data is shown below.



We can see that the trend has been removed from the data, and we are ready to propose potential models. However, the variance is perhaps not constant and this may be troublesome.

Proposing Models

Using the stationary data we have achieved, we want to fit a few Box-Jenkins models on the data. In order to find suitable models to propose, we look at the ACF and PACF of the data.



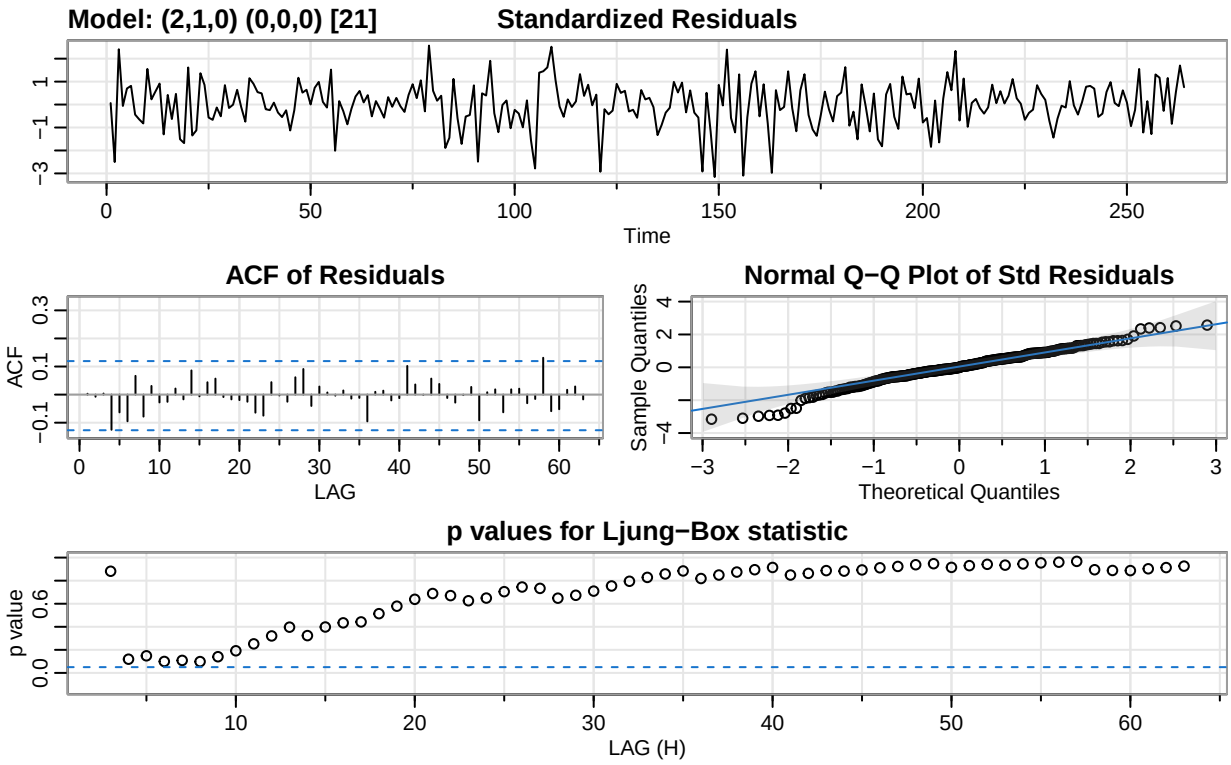
Thus, we propose the following models:

$$\begin{aligned}
&\text{Model 1} : \text{AR}(2)\\
&\text{Model 2} : \text{ARMA}(2,1)\\
&\text{Model 3} : \text{ARMA}(1,1)\\
&\text{Model 4} : \text{AR}(1)\\
&\text{Model 5} : \text{MA}(1)\\
&\text{Model 6} : \text{ARMA}(1,2)\\
&\text{Model 7} : \text{ARMA}(2,2)
\end{aligned}$$

For the purpose of conciseness, we will be focusing only on model 1 and 2 (the two that satisfied assumptions and had the lowest MSE). See Exhibit 3 of the Appendix for details on all 7 of the models that we proposed and fit.

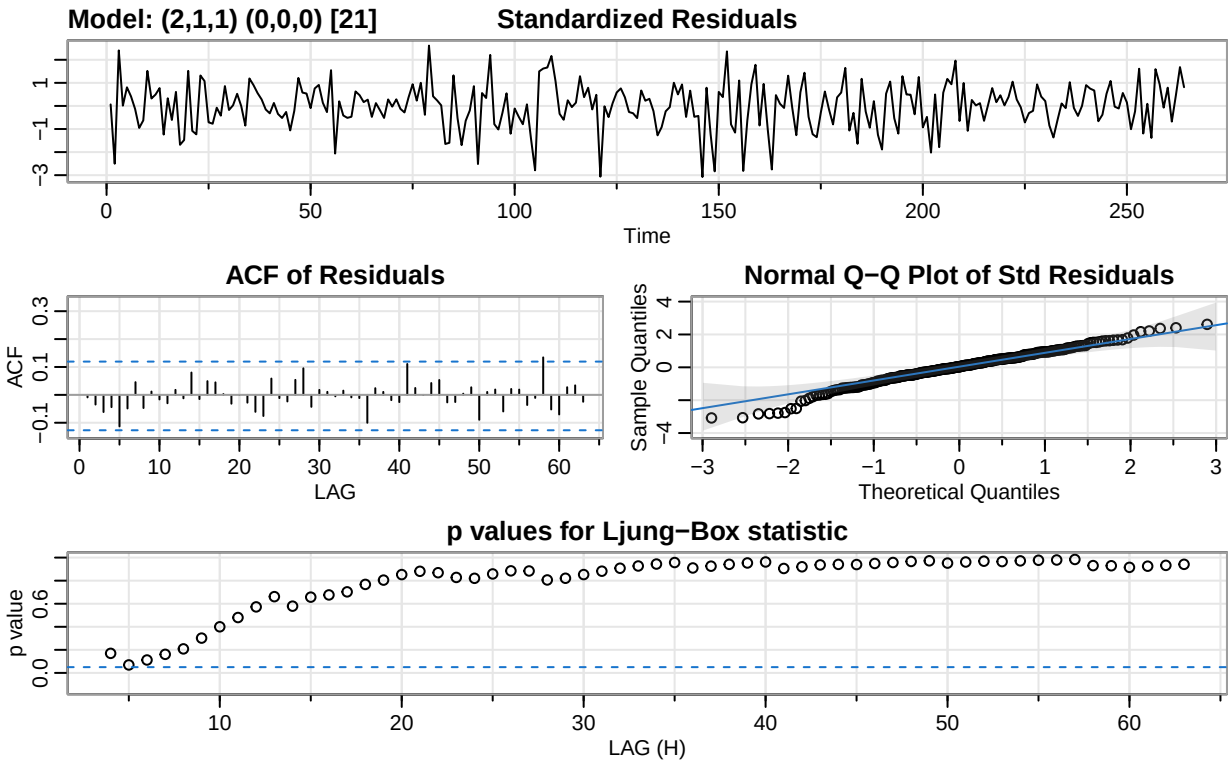
Fitting Models

AR(2): We proposed this model since it can be argued that the PACF cuts off after lag 2 and that the ACF experiences very quick exponential decay. Fitting the $\text{AR}(2) = \text{SARIMA}(2,1,0) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are a few LAG values on the border, however, none seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model.

ARMA(2,1): We proposed this model since it can be argued that the PACF and the ACF experience very quick exponential decay. Then we chose to start with 4 simple ARMA models. Fitting the ARMA$(2,1) =$ SARIMA$(2,1,1) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are no LAG that seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model too.

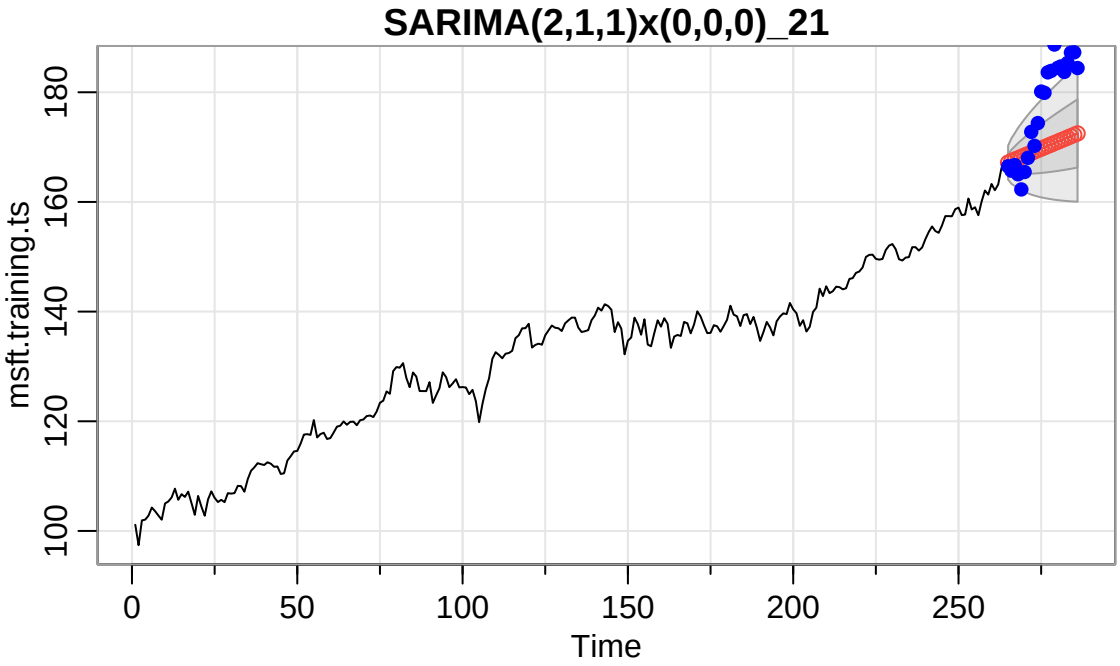
Choosing the best model

We can compute the prediction MSE for the two models above to compare their prediction power over the test set:

Table 3: Prediction MSE by Model

Model Name	Prediction MSE
AR(2)	108.58
ARMA(2,1)	103.90

Thus, the best model is ARMA(2,1) since it minimizes the prediction MSE. We can use this and look at the plot of its predicted values against the test set's true values.



Conclusions

We see that even our best model, ARMA(2,1), does not do well in predicting the test set. The regression model and Double exponential smoothing models did a much better job. This might be due to the fact that constant variance could not be achieved.

Conclusions

Statistical Conclusions

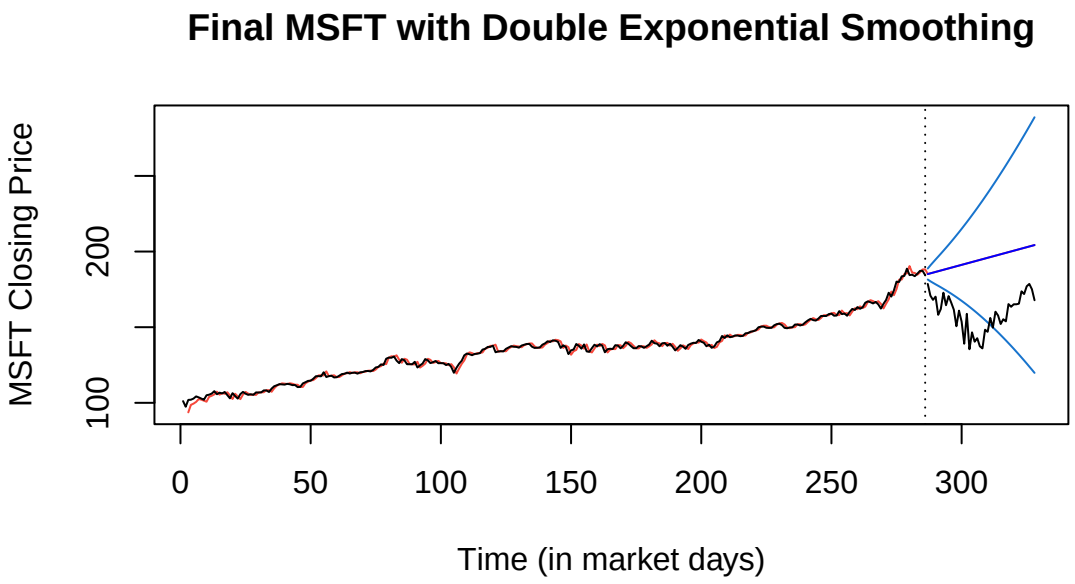
Since our focus is on comparing modeling techniques with respect to their prediction power over the test set, we will compare the prediction MSE for our best models from each section of the analysis:

Table 4: Prediction MSE by Model

Model Name	Prediction MSE
Unregularized Degree 8 Regression Polynomial	18.75
Double Exponential Smoothing	20.71
ARMA(2,1)	103.90

When we looked at these models in more depth above, we analyzed their residuals. For the un-regularized polynomial of degree 8, the residuals had evidence of a non-periodic trend still present and were, thus, not stationary. Both double exponential smoothing and ARMA(2,1) successfully removed the trend from the data and the residuals looked similar in terms of variance. They were arguably not perfectly constant, however, out of all the models that we tested this was as good as we could get. It is important to note that we had difficulty stabilizing the variance in our data before implementing these techniques, and since non-constant variance is not a problem for the Holt-winters algorithm this leads us to favour double exponential smoothing. Additionally, while the polynomial technically minimized the prediction MSE, it was very close to the MSE for double exponential smoothing.

Thus, the final model that we will use for forecasting is the double exponential smoothing model since it did a good job satisfying stationarity assumptions and minimized the prediction MSE. Now we can fit the entire data (training and test set) to this model and derive our final forecast over our period of interest. Since we are interested in predicting 2 months after the stock market crash this translates to a prediction for all market days between February 21st, 2020 through to April 21st, 2020 inclusive.



We can immediately see that the prediction does not accurately capture what happened over these two months. In fact, over half of the predictions are not even covered by the 95% prediction interval. It seems that the stock market crash was a change point that resulted in changes to the non-periodic trend and a short-term increase in variance. Clearly this data had a problem with extrapolation since we had to assume that the pattern observed in 2019 up until the crash was going to project outside the range of data used to fit the model. We know now that this assumption was wrong.

Contextual Conclusions

Overall, it is fair to say that our forecast for the first two months after the 2020 stock market crash on MSFT's closing price deviates quite drastically from what actually happened. Prior to the crash, the price was consistently increasing and our modeling techniques, including double exponential smoothing, assumed that the future would look like this too. This was clearly not the case. Additionally, we can see that for the first few weeks after the crash the stock price continued to fall, however, it began to recover very quickly and was more or less back on track by the end of April 2020. If an investor were to have used our forecasts to make a profit on trading, this may have led them astray. For example, if someone purchased stock in early February 2020, expecting the upwards trend to continue as the model expects, they may have been panicked by the stock crashing and panic sold.

This demonstrates just how useful understanding stock data as a random walk is. The stock price was strongly influenced by a random event, in this case it was the onset of the COVID-19 pandemic. The resulting global lock downs and risk of supply chain issues led to a great deal of uncertainty in the market. No one could have constructed a model to accurately forecast this event and its effect on MSFT's stock. This is why it is commonly agreed that one cannot successfully "game" the stock market. In final consideration, while forecasting has a plethora of real world uses it is always important to remember that real data may behave drastically different than we, and our models, are expecting

Appendix

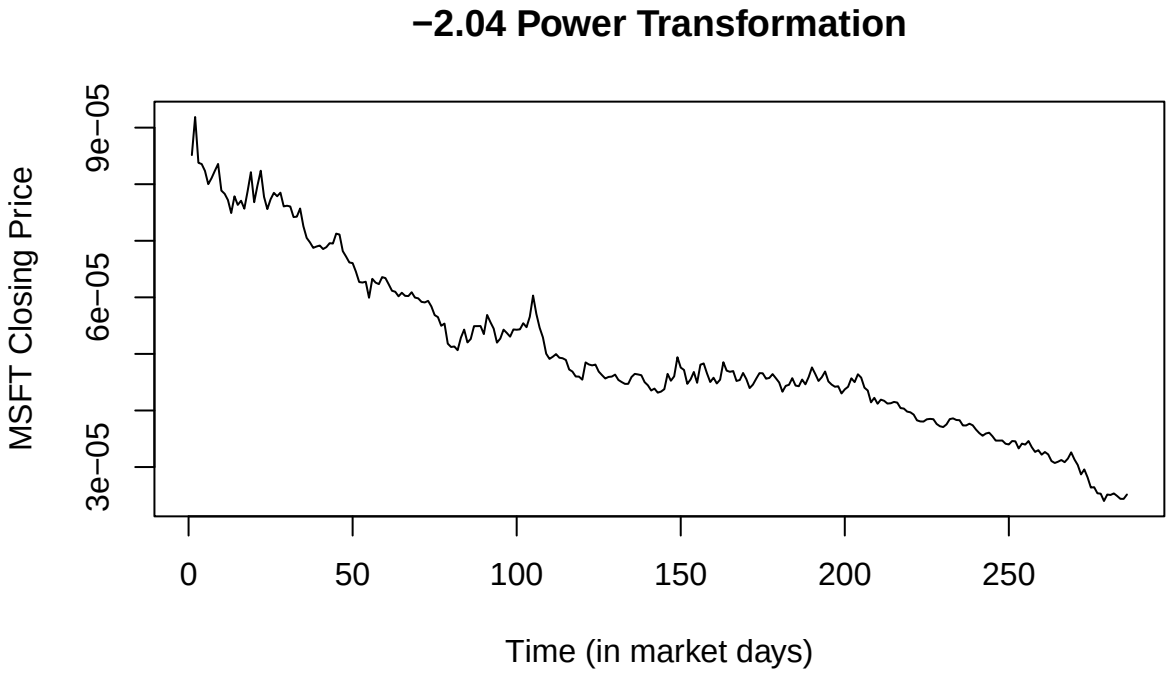
Exhibit 1

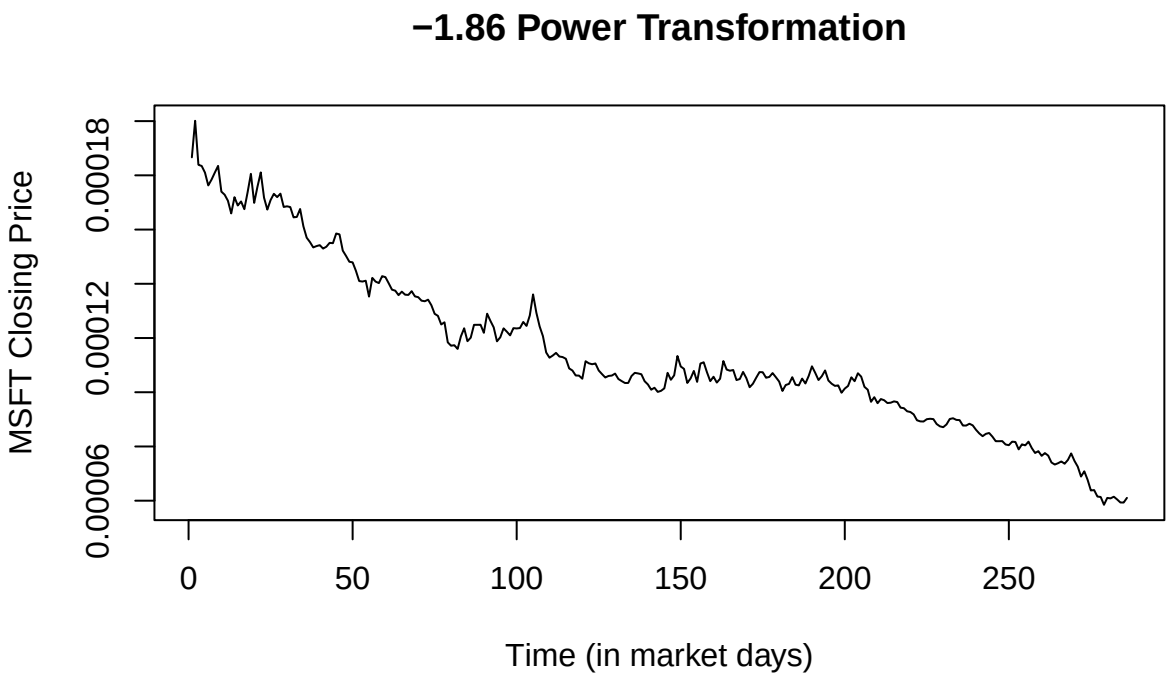
Results from Fligner's test on the data with 3 separate segmentations:

Table 5: Results from Fligner's Test

Segmentation	Original P-value	Optimal Power Transformation	New P-value
11 segments of 26	1.434201e-18	-2.03	3.445292e-9
7 segments of 35, 1 of 41	3.214942e-14	-1.86	9.095686e-8
14 segments of 19, 1 of 20	1.535037e-8	-0.67	2.316742e-5

P-values for power transformations from -5 to 5 at two decimal places of granularity were tried for each segment. None of them made significant improvements in maximizing the p-value. We can also plot these to visually demonstrate that the variance was not improved:




−1.86 Power Transformation
MSFT Closing Price
0.00006
0.00012
0.00018
0
50
100
150
200
250
Time (in market days)

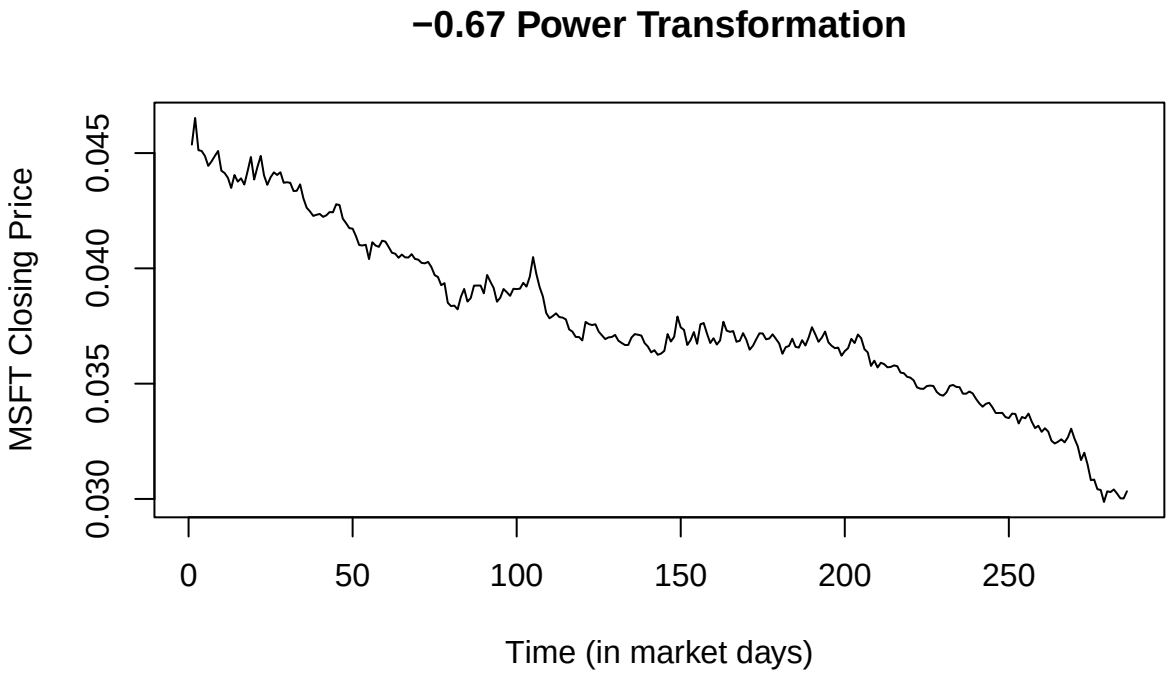
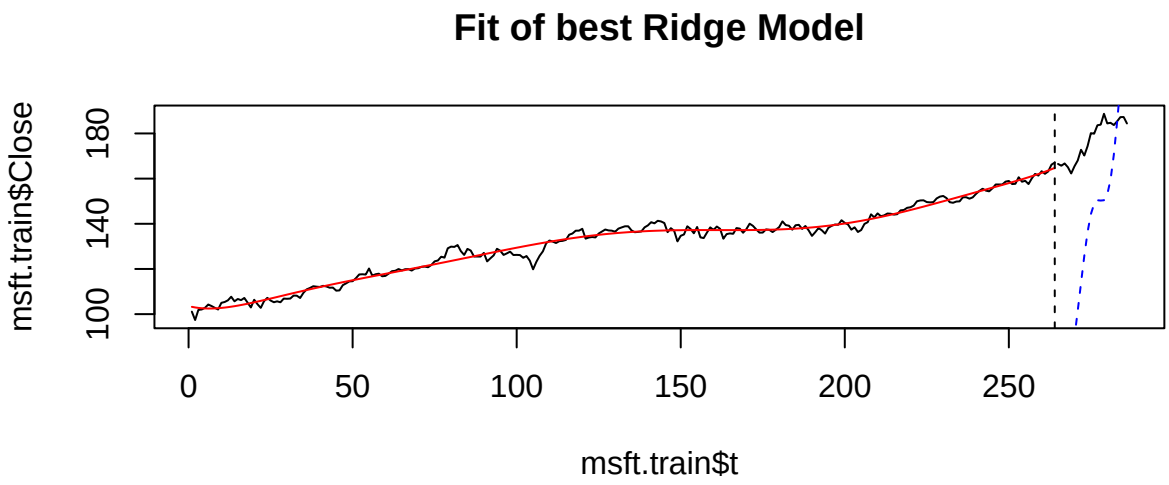

−0.67 Power Transformation
MSFT Closing Price
0.030
0.035
0.040
0.045
0
50
100
150
200
250
Time (in market days)

Exhibit 2

Below is a plot for the fit and test set prediction from the optimal model with ridge regression. These were excluded in the body of the report since they did not add anything (regularized polynomials performed very poorly).



Looking at the predicted values, it is immediately obvious why we felt that ridge regression did a very poor job even on its optimal degree model.

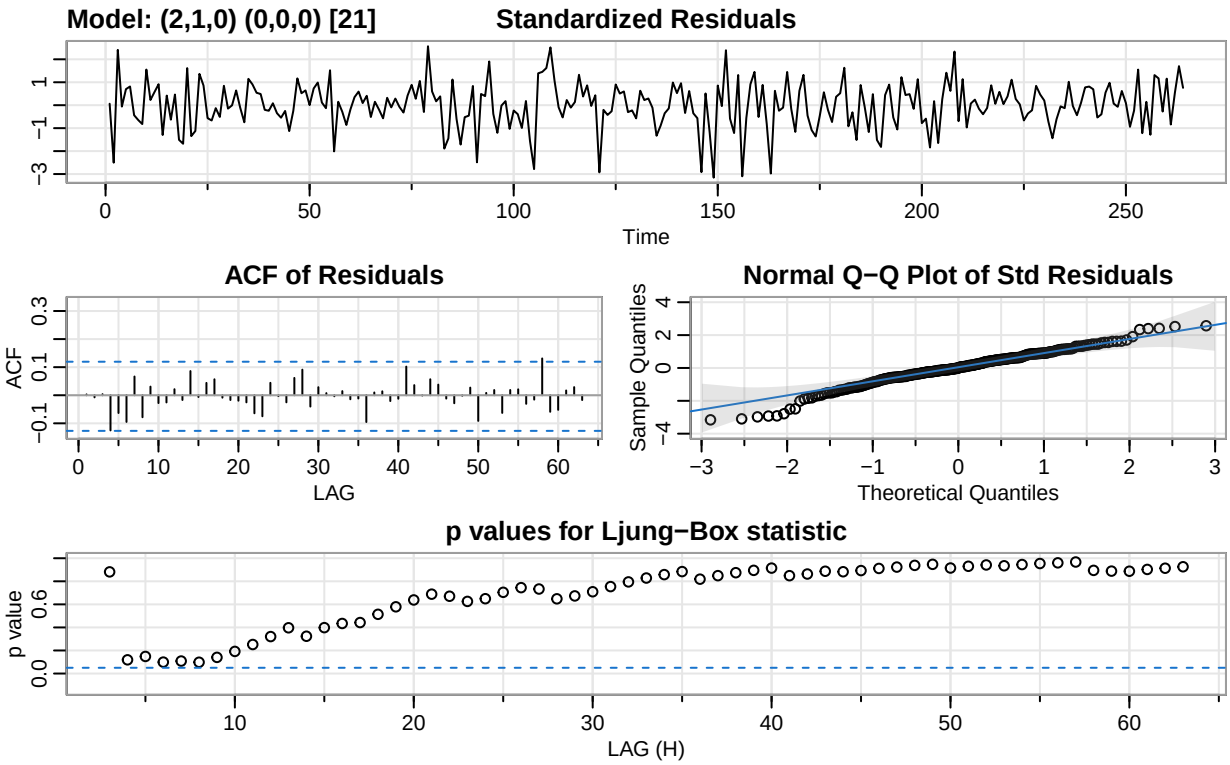
Exhibit 3

Proposed Models, their justification, and their SARIMA Residual Analysis:

Model 1 : AR(2)
Model 2 : ARMA(2, 1)
Model 3 : ARMA(1, 1)
Model 4 : AR(1)
Model 5 : MA(1)
Model 6 : ARMA(1, 2)
Model 7 : ARMA(2, 2)

AR(2)

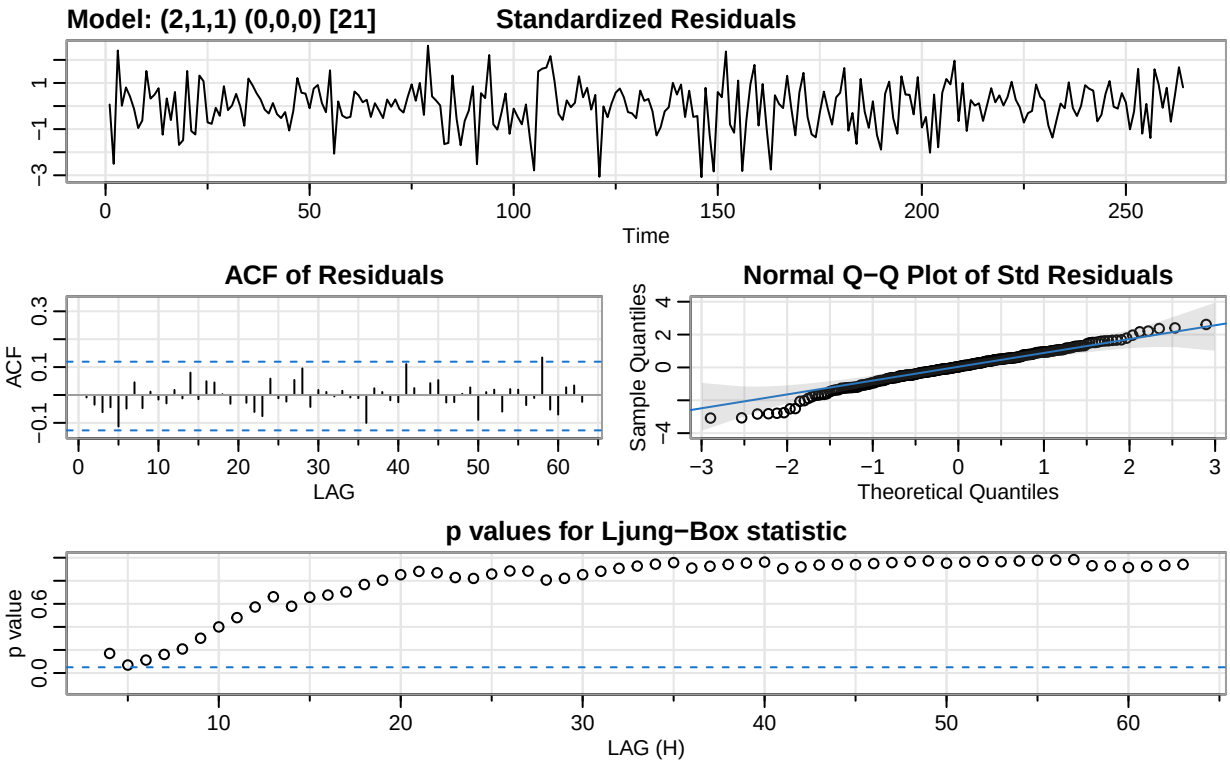
We proposed this model since it can be argued that the PACF cuts off after lag 2 and that the ACF experiences very quick exponential decay. Fitting the AR(2) = SARIMA$(2, 1, 0) \times (0, 0, 0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are a few LAG values on the border, however, none seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model.

ARMA(2,1)

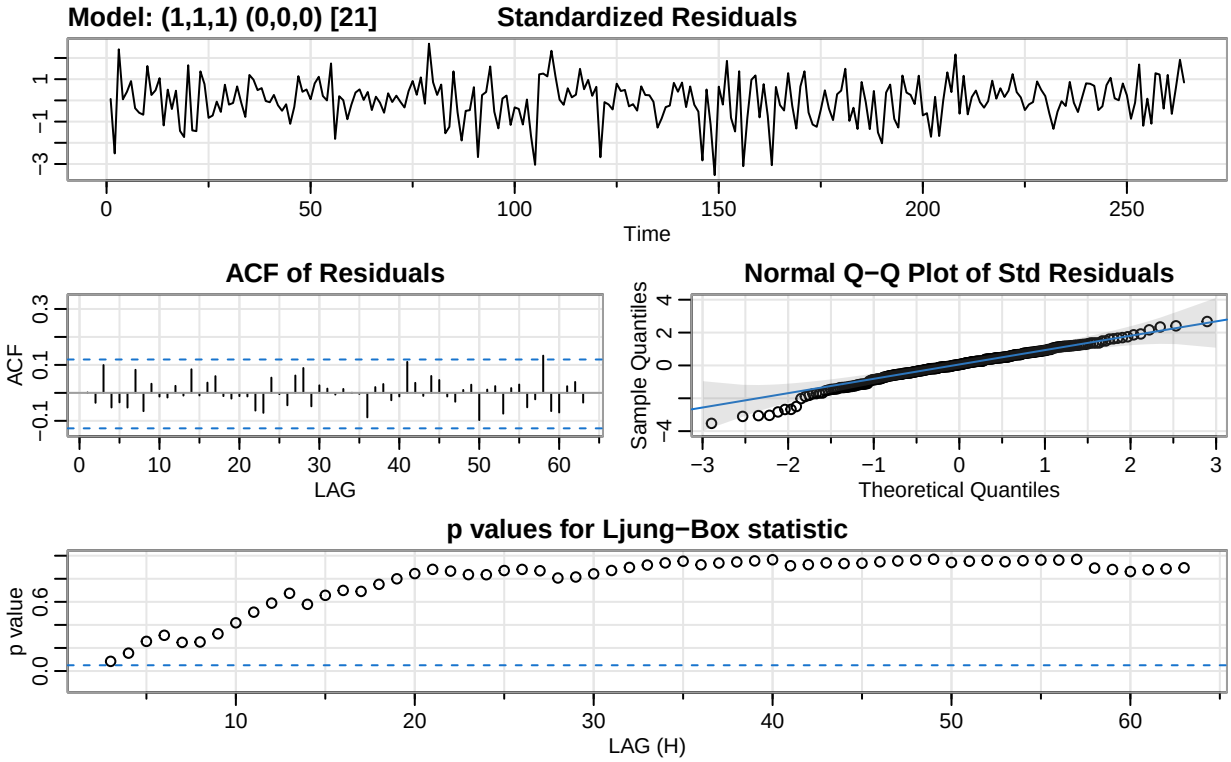
We proposed this model since it can be argued that the PACF and the ACF experience very quick exponential decay. Then we chose to start with 4 simple ARMA models. Fitting the $\text{ARMA}(2,1) = \text{SARIMA}(2,1,1) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are no LAG values that seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model too.

ARMA(1,1)

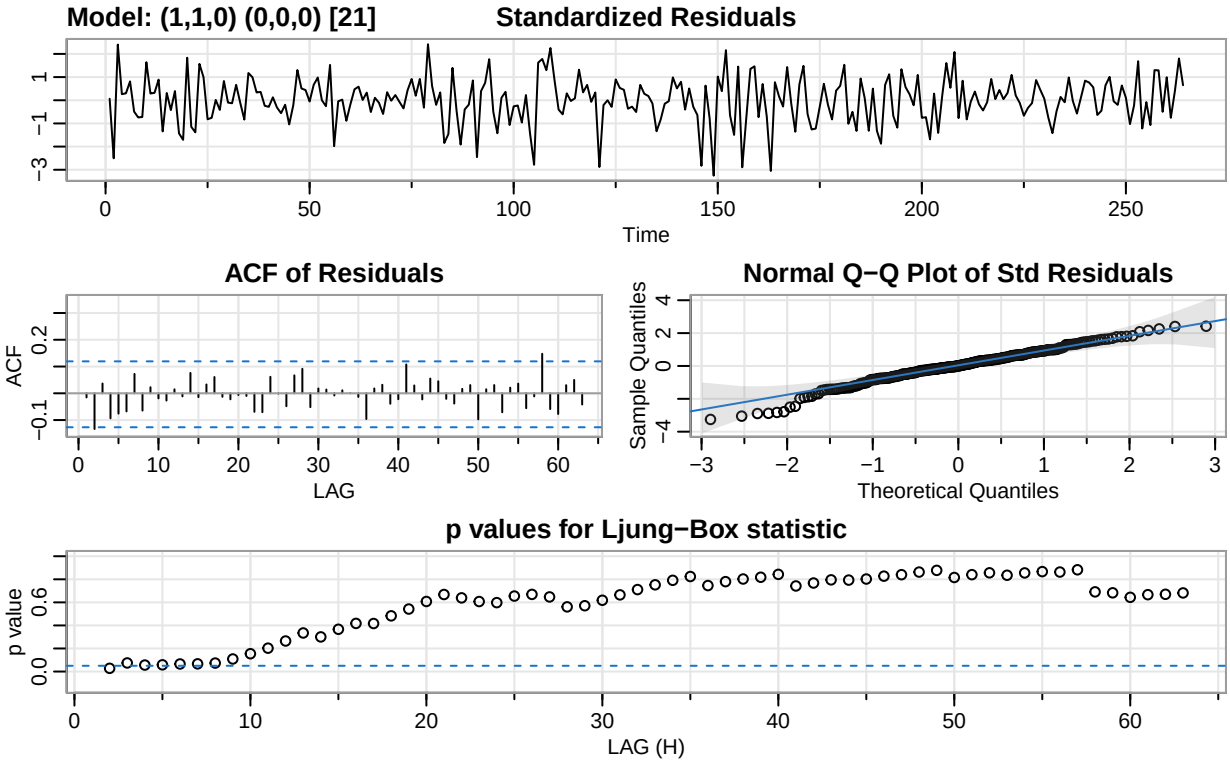
We proposed this model since it can be argued that the PACF and the ACF experience very quick exponential decay. Then we chose to start with 4 simple ARMA models. Fitting the ARMA$(1,1)$ = SARIMA$(1,1,1) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are no LAG values that seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model too.

AR(1)

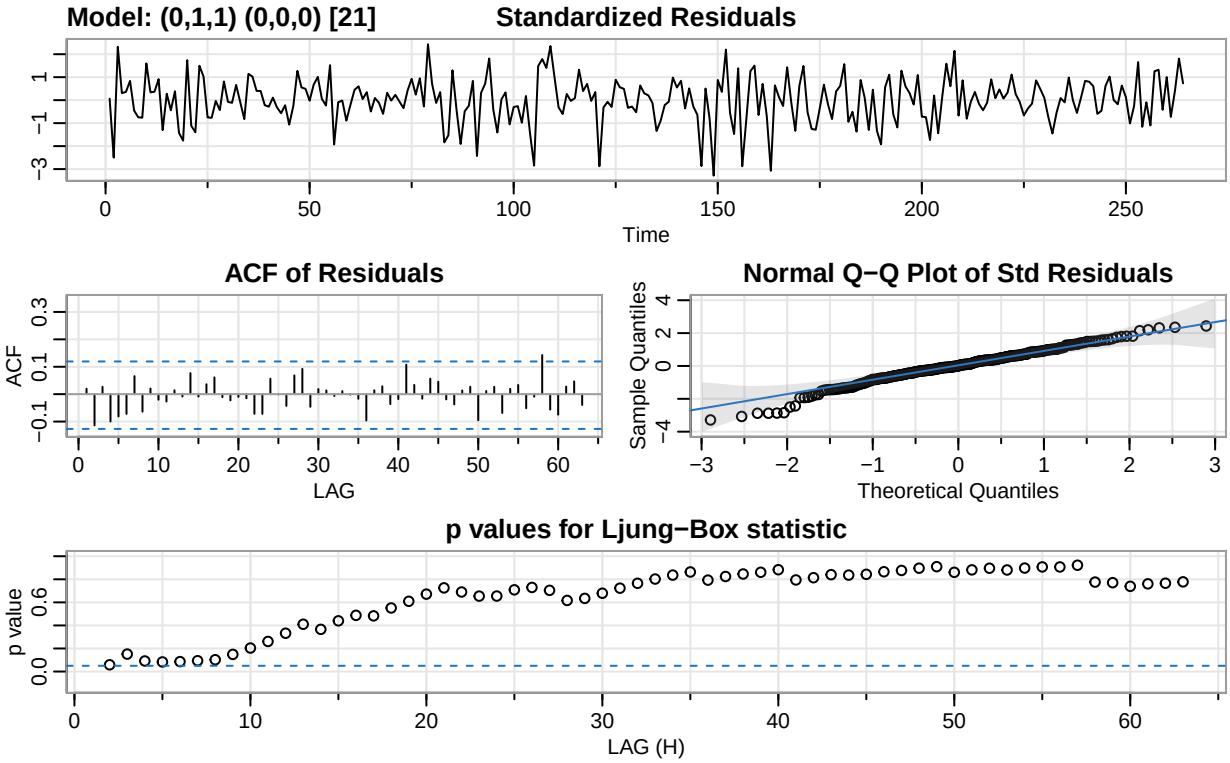
We proposed this model since it can be argued that the PACF cuts off after lag 1 (lag 2 is random) and that the ACF experiences very quick exponential decay. Fitting the AR(1) = SARIMA$(1, 1, 0) \times (0, 0, 0)$ model to the training data, the residuals are as follows:



The normality looks okay (there is a slight deviation, but it is at the end points), there are no collinearity issues, and the residuals look stationary. However, there are quite a few points on the border of failing the Ljung-Box Statistic test and one that looks like it did fail. Therefore, since we have better models we will not consider this model.

MA(1)

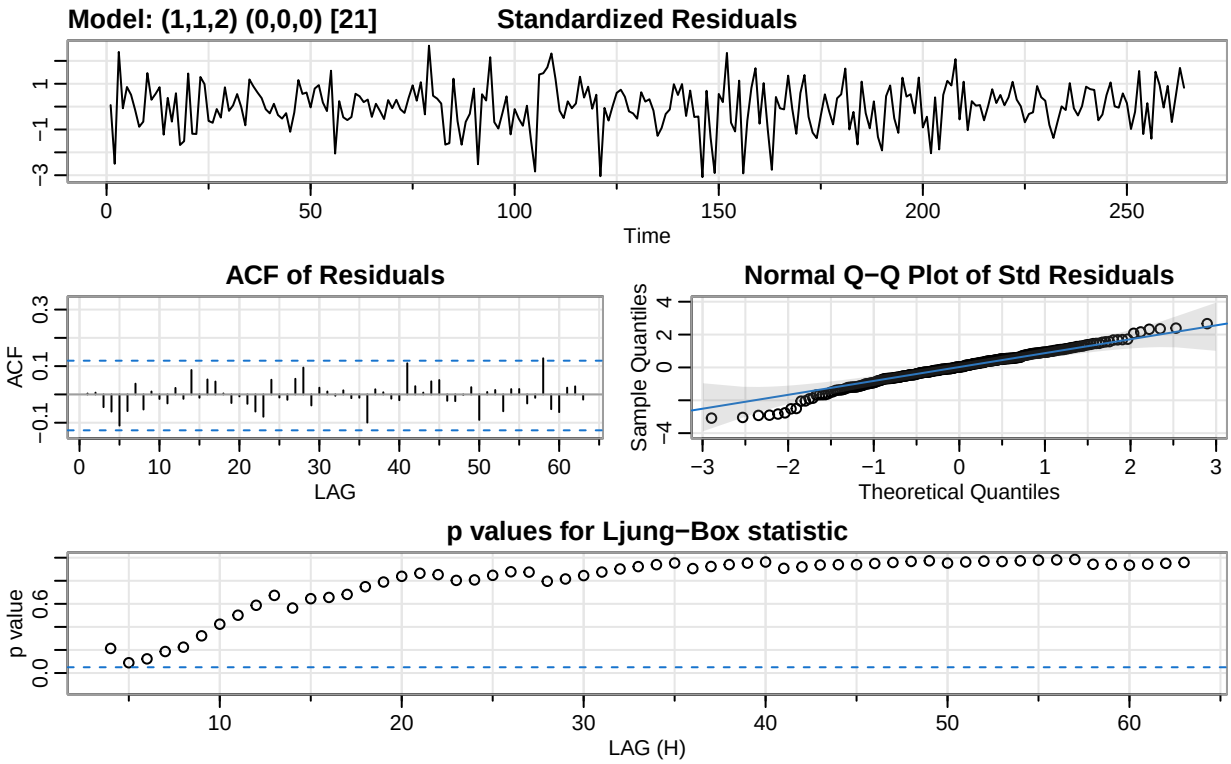
We proposed this model since it can be argued that the ACF cuts off after lag 1 and that the PACF experiences very quick exponential decay. Fitting the MA(1) = SARIMA$(0,1,1) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are a few LAG values on the border, however, none seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model..

ARMA(1,2)

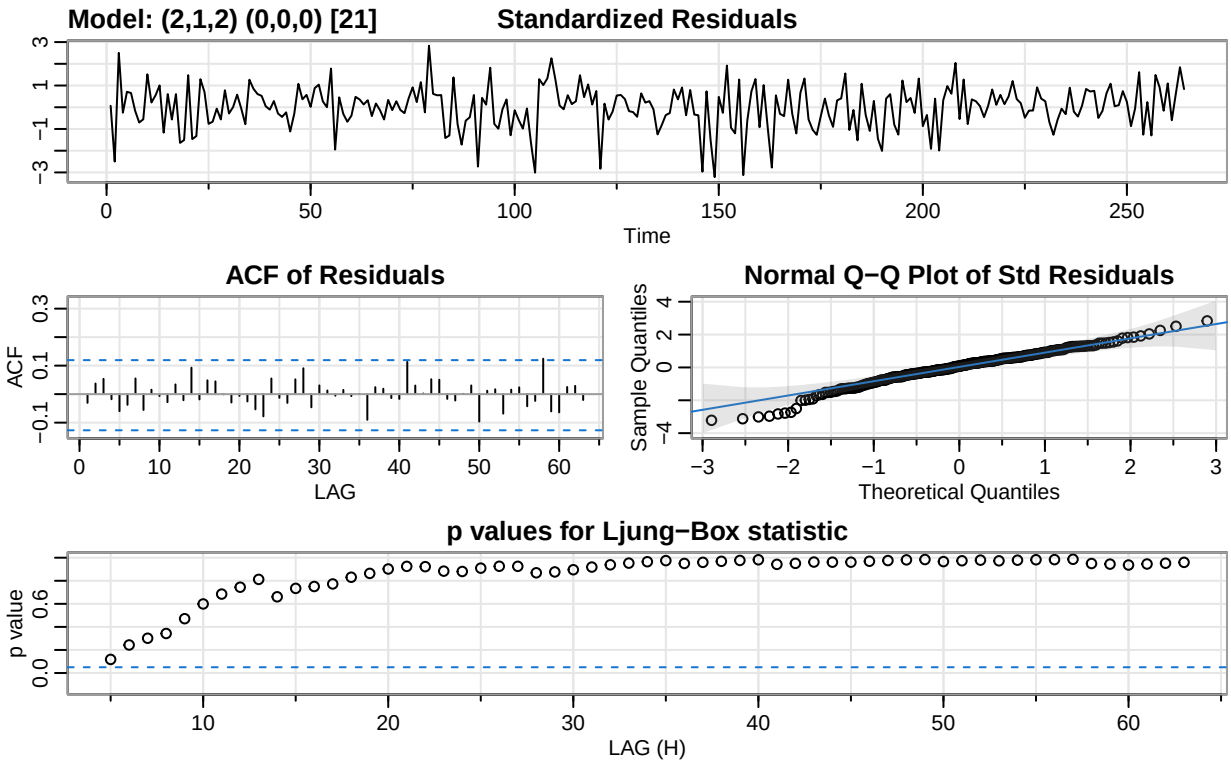
We proposed this model since it can be argued that the PACF and the ACF experience very quick exponential decay. Then we chose to start with 4 simple ARMA models. Fitting the $\text{ARMA}(1,2) = \text{SARIMA}(1,1,2) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are no LAG values that seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model too.

ARMA(2,2)

We proposed this model since it can be argued that the PACF and the ACF experience very quick exponential decay. Then we chose to start with 4 simple ARMA models. Fitting the ARMA$(2,2)$ = SARIMA$(2,1,2) \times (0,0,0)$ model to the training data, the residuals are as follows:



These diagnostic plots look good. There are no LAG values that seem to fail the Ljung-Box Statistic test. The normality also seems okay (there is a slight deviation, but it is at the end points) and there are no collinearity issues. And importantly, the residuals look stationary. Therefore, we will consider this model too.

We can then compare the Prediction MSEs over the test set for each of the selected models:

Table 6: Prediction MSE by Model

Model Name	Prediction MSE
AR(2)	108.58
ARMA(2,1)	103.90
ARMA(1,1)	132.53
MA(1)	104.35
ARMA(1,2)	107.87
ARMA(2,2)	130.37

Thus, out of these selected models, ARMA(2,1) minimizes the prediction MSE and is our optimal model.