Title
Using Spectral Analysis and Autoregressive Moving Average Models to Identify Patterns in the Financial Markets

Author
Hsu, Rita

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Using Spectral Analysis and Autoregressive Moving Average Models to Identify Patterns in the Financial Markets

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Applied Statistics

by

Rita Hsu

2022

ABSTRACT OF THE THESIS

Using Spectral Analysis and Autoregressive Moving Average Models to Identify Patterns in the Financial Markets

by

Rita Hsu
Master of Applied Statistics in
University of California, Los Angeles, 2022
Professor Frederic R. Paik Schoenberg, Chair

It is clear that the intricacies of the stock market have prompted many to conduct research in this potentially lucrative topic of analysis. Some suggest that the stock market obeys the random walk hypothesis, some explore cyclical patterns, and others contemplate on the impact of macroeconomic variables on stock market performance. This study aims to investigate these questions by analyzing the Dow Jones Industrial Average (DJIA) data. Through autoregressive moving average (ARMA) modeling, spectral analysis and moving average filtering, we find evidence agreeing to the random walk hypothesis, uncover correlation between the macroeconomic environment and stock market returns, and encounter limits of the ARMA models in sustaining their predicting accuracy during times of uncertainty. In this paper, we present the evidence and meaningful findings of the study, and in addressing the limits we offer potential approach for extended research.

The thesis of Rita Hsu is approved.

Vivian Lew

Yingnian Wu

Frederic R. Paik Schoenberg, Committee Chair

University of California, Los Angeles

2022

To my mother, father, and sister . . .

who—among so many other things—

supported and advocated for me along my academic journey

as I clumsily worked my way through each step

in the pursuit of self-discovery

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank the Master of Applied Statistics program faculty and staff, for making time for MAS students in the evenings and being understanding on many fronts to accommodate part time students who work full time. It is with their continuous support that I could enjoy learning and finish the program smoothly.

Next, I would like to thank the members of my thesis committee who reviewed my paper and provided valuable feedback. Special thanks to Professor Schoenberg for the tremendous amount of help and guidance he provided me through numerous zoom meetings. I would also like to thank Vivian Lew for introducing me to the program and helping me get through hard times when I was close to giving up. I will forever be grateful for all you have done for me through my undergraduate and graduate years.

Lastly, I would like to acknowledge the UCLA Statistics department and community as the place where I found my passion for data after exploring different majors as an undergraduate student and then later deepened it as a graduate student. It has been such a blessing to get to pursue my studies here and be part of this community. I am grateful for everyone who I have crossed path with during my time here.

CHAPTER 1

Introduction

Predicting the future is no small task, but knowing the future can bring tremendous value. People put a lot of their spare income into the stock market, hoping that stock prices will rise over time, which would then increase their wealth. The potential financial gains and the complex market dynamics have made forecasting the stock prices a popular topic of discussion and analysis, because understanding the market and predicting it well would mean building wealth and reaching financial freedom that much faster.

Several past studies have hypothesized the notion that the stock market behaves like a random walk. Granger and Morgenstern found in their study on the New York Stock Exchange data that the short term movements in the market follow the random walk hypothesis [GM63][GGM64]. However, aside from patterns in day-to-day activities, historically there have been "black swan" events in the stock market, a term coined by Nassim Taleb as rare, out-of-the-ordinary events with extreme impact [Tal07]. Examples of such an event include the terrorist attack that occurred on September 11, 2001, and the 2008 financial crisis. There remains to be little research that explore how these black swan events may or may not affect the hypothesis that the stock market is similar to a random walk.

Another black swan event occurred in March 2020, when news of the COVID-19 outbreak around the world sent the global stock markets plummeting, bringing a large amount of uncertainty and unprecedented losses. The economic damage was reported to be greater than those from the Severe Acute Respiratory Syndrome (SARS) outbreak in 2003 and the Spanish flu of 1918 [TTK21]. The government subsequently carried out drastic interest rate changes and money supply measures to ease the economy from further damages, prompting the stock market to quickly recover and the crash to stop. In this sequence of events, we can clearly see that changes to the macroeconomic environment, brought by government policies in this case, appear to have an influence on investor sentiment in the stock market and therefore the stock market performance.

In the past, there have been research that investigated the relationship between stock market returns and numerous macroeconomic variables, with varying results. In their study on the U.S. stock market, Ratanapakorn and Sharma found positive relationships between stock prices and variables such as inflation and short-term interest rates, but a negative relationship between stock prices and long-term interest rates [RS07][Nai13]. Wongbampo and Sharma, in a study that looked at stock market data from five Asian countries, found the relationship between stock prices and interest rates to be negative in three countries but positive in the other two [WS02][Nai13]. There were also researchers whose studies concluded that the relationship between the macroeconomic environment and the stock market is not statistically significant [PK00]. All in all, past studies seem to suggest that the existence and the directions of relationships between macroeconomic variables and the stock market returns vary across different markets and time periods of research [Nai13]. Thus, this analysis attempts to dive into the relationship between some macroeconomic variables and the U.S. stock market returns, specifically in the time frame of the COVID-19 pandemic.

This leads us to ask several key questions of interest which will be addressed in the paper: (1) does the stock market behave like a random walk? (2) do macroeconomic variables correlate with the stock market returns? (3) how well do our models, trained on pre-pandemic data, predict stock market returns after the COVID-19 crash? We test the hypothesis that the stock market follows a random walk by performing spectral analysis and autoregressive moving average (ARMA) model fitting. We investigate the potential impact of macroeconomic variables by incorporating several of them as covariates in the model fitting process. Utilizing the ARMA models, we find that stock market returns fit closely to an ARMA(0,0). Through spectral analysis, we discover cycles that appear to improve the model fit but prove to be statistically insignificant. We also observe significant correlation between stock market returns and several macroeconomic variables, such as inflation and the U.S. Treasury 10-year yield. Moreover, after training our models on pre-pandemic data and testing their performance on post-pandemic data, we find that the predictions return to pre-pandemic levels of accuracy after suffering a short period of increased errors. Lastly, we conclude that our models do not predict stock market returns more accurately than a white noise model, agreeing to the theory that the stock market is similar to a random walk.

The rest of this paper is structured as follows. Chapter 2 elaborates on the methodology used in analyzing the data, including ARMA model fitting, investigation of cycles through spectral analysis and moving averages, and cycle incorporation through linear regression. In Chapter 3, we focus on interpreting the models through diagnosis plots and predictions made from each model, and making model recommendation based on measures of model fit and prediction performance. Chapter 4 summarizes main conclusions of the research, explains how they answer the key questions, and shares a few potential areas where there may be room for further research.

CHAPTER 2

Description of Data

The data we focus on for this study is the Dow Jones Industrial Average (DJIA) daily close price, which is obtained from Yahoo Finance. The full dataset has approximately five years of daily data (November 2016 $\sim$ November 2021), including the DJIA index's opening price, closing price, highest price and lowest price of each trading day. For model testing purposes, we split the dataset by assigning the earlier 60% of the data to training and the remaining 40% for testing. This translates to the first three years of the dataset being used for training the model, and the rest for testing purposes. Figure 2.1 shows the DJIA daily close price plotted across time for which the entire dataset spans. The colors differentiate whether the time period is used for training or testing, with the training period shown in black and testing period shown in blue.

Figure 2.1: DJIA daily close price

CHAPTER 3

Methods

To make the time series more stationary, we first transform the DJIA close price data into daily percentage returns. Figure 3.1 illustrates the DJIA daily percentage returns series moving around the value 0, marked in black during the training period and blue during the testing period. Trading days that had big movements can be clearly seen here. For example, we see the COVID-19 crash in March 2020 as well as the quick rebound afterwards. Also can be seen is the 4% drop in February 2018, primarily driven by to fear of inflation and rising interest rates at the time.

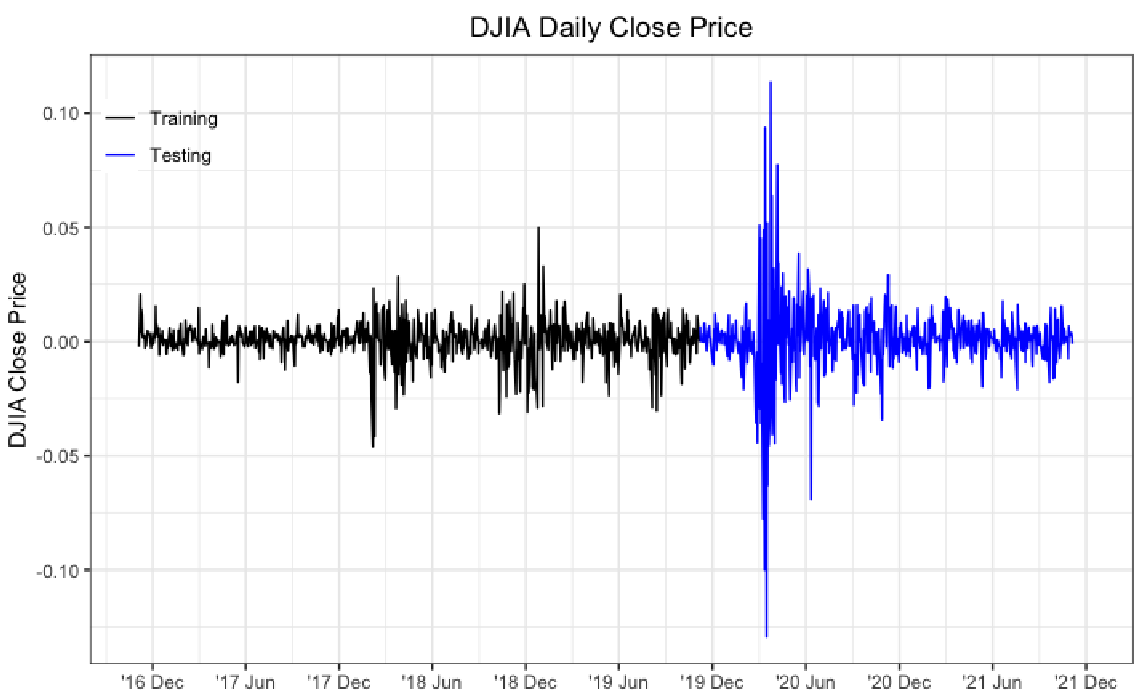


Figure 3.1: DJIA daily percentage returns

3.1 ACF and PACF

We then evaluate autocorrelation function (ACF) and partial autocorrelation function (PACF) charts for DJIA returns. The ACF function, defined in Equation 3.1, gives us correlation between values of a time series at two different time points, or at a lag, which can in turn help us understand the predictability of present value of the time series based on its past values [SS17]. The ACF chart shows us ACF values on the y-axis at different lags reflected on the x-axis. Using an ACF chart, we can find notable correlation within the series at different lags by observing where the ACF value peaks above the confidence interval of an uncorrelated series. A peak at lag 0 is normal as a series would be perfectly correlated with itself. A 1-day correlation would show up as a peak in lag 1, and a 5-day correlation would show up as a peak in lag 5. Figure 3.2 shows ACF values of the DJIA returns series, and it appears that the highest values of ACF occur at lags 14 and 24. This means that the DJIA series is most correlated with its value from 14 days ago and 24 days ago. However, even these peaks in ACF are barely outside the confidence interval for an uncorrelated series, which suggests that there is close to no statistically significant autocorrelation within the DJIA returns series.

$$\rho(h) = \frac{\gamma(t+h,t)}{\sqrt{\gamma(t+h,t+h)\gamma(t,t)}} \tag{3.1}$$

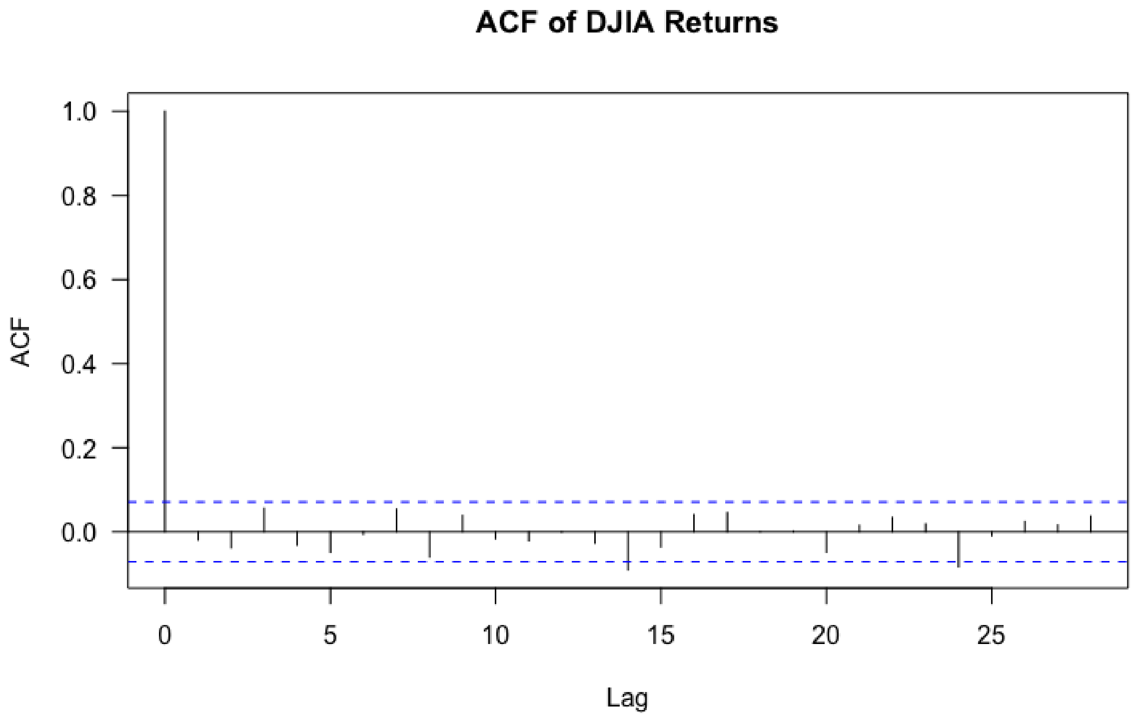


Figure 3.2: ACF of DJIA returns

The PACF function, defined in Equation 3.2, measures the correlation of the series and its past values, with the estimated impact of those past values removed. In other words, it gives us the correlation of the series and the residuals from regressing on its past values [SS17]. Using a PACF chart, we can identify at which lag(s) the series is most correlated with the residuals. Figure 3.3 shows that PACF values of the DJIA returns series similarly peak at lags 14 and 24.

$$\phi_{hh} = \begin{cases} corr(x_{t+1}, x_t) = \rho(1) & , h = 1 \\ corr(x_{t+h} - \hat{x_{t+h}}, x_t - \hat{x_t}) & , h >= 2. \end{cases} \tag{3.2}$$

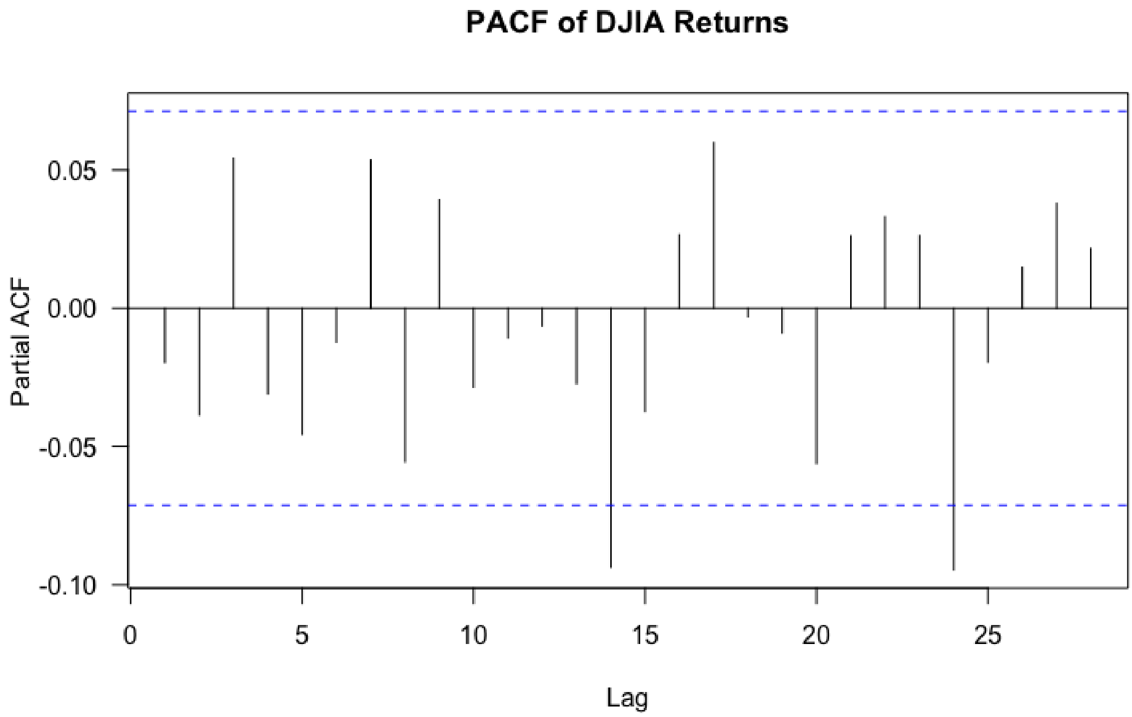


Figure 3.3: PACF of DJIA returns

3.2 ARMA

Next, we fit the first model on the DJIA returns series with an ARMA model. An ARMA model fits a series with a mixture of an autoregressive (AR) model and moving average (MA) model. An AR model regresses the series on its own past values, hence the name autoregressive model. In other words, the present value of the series x_t is fitted as a function of its values from the past, $x_{t-1}, ..., x_{x-p}$, with p representing the furthest back in time for which the model incorporates its history and the order of the AR model. The form of an AR(p) model is shown in Equation 3.3 [SS17].

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t \tag{3.3}$$

While an AR model regresses on past values of the series, an MA model regresses on past error terms of the series. Equation 3.4 shows the form of an MA(q) model, where q is the furthest back into the past for which the model incorporates error terms and therefore the order of the MA model [SS17].

$$x_t = w_t + \theta_1 w_{t-1} + ... + \theta_q w_{t-q} \tag{3.4}$$

As previously mentioned, an ARMA model is a mixture of an AR and MA model. By combining AR and MA, an ARMA model includes the impact of the series' past values (AR) as well as patterns in the residuals from regressing on the series' past values (MA). Equation 3.5 shows the form of an ARMA(p,q) model, in which p represents order of the AR portion and q represents order of the MA portion [SS17]. In this first model fit on the DJIA returns data, we find ARMA(0,0) with a mean of 0 to be the best fit, which indicates that the DJIA returns behave similarly to white noise.

$$x_t = \phi_1 x_{t-1} + ... + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + ... + \theta_q w_{t-q} \tag{3.5}$$

3.3 Spectral Analysis

It is common for time series data to have cyclical patterns in them, and we explore this by performing spectral analysis on the DJIA returns. We look at the spectral density of the DJIA returns at different frequencies. Figure 3.4 is a periodogram with spectral density on the y-axis and the range of frequencies on the x-axis. The red horizontal line marks the mean of the spectral densities, and the blue vertical line demonstrates the confidence band. Spectral densities that surpass the mean by at least the length of the confidence band (or length of the blue line) are considered statistically significant. The frequencies of these peaks in spectral density would then lead us to cyclical patterns that are considered statistically significant. In Figure 3.4, while there is a peak around frequency of 0.3, or a period of 3 days, it is not greater than the mean by enough to be considered statistically significant. Figure 3.5 shows the spectral density through an AR fit of the series. Similarly to the periodogram, any patterns would show as peaks in the spectral densities at their corresponding frequencies. With the values of spectral density showing as a flat line with no peaks, Figure 3.5 presents the same conclusion that there is no statistically significant cyclical patterns found.

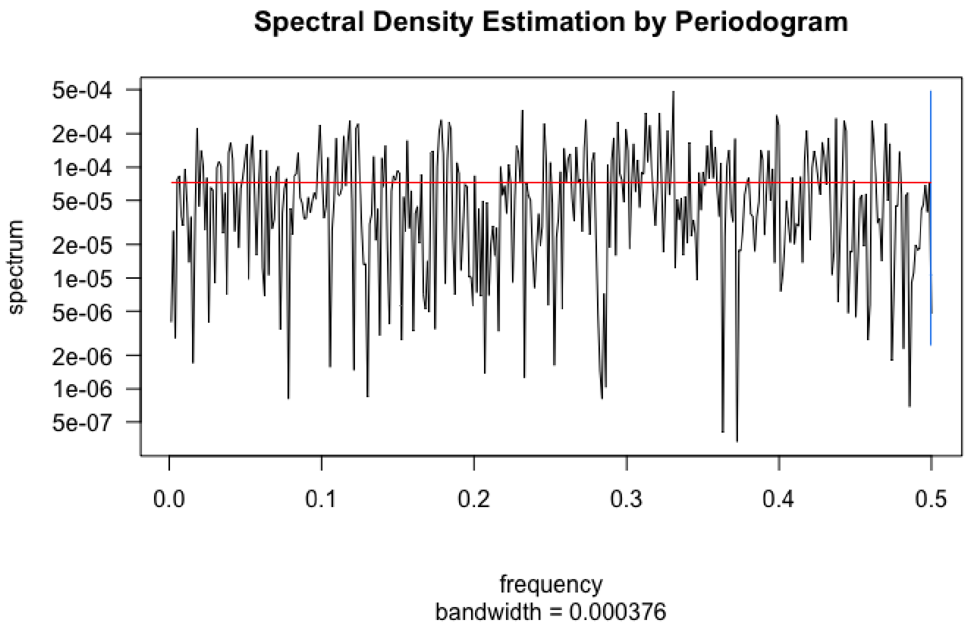


Figure 3.4: Estimated spectral density through a periodogram of DJIA returns with the mean and confidence intervals

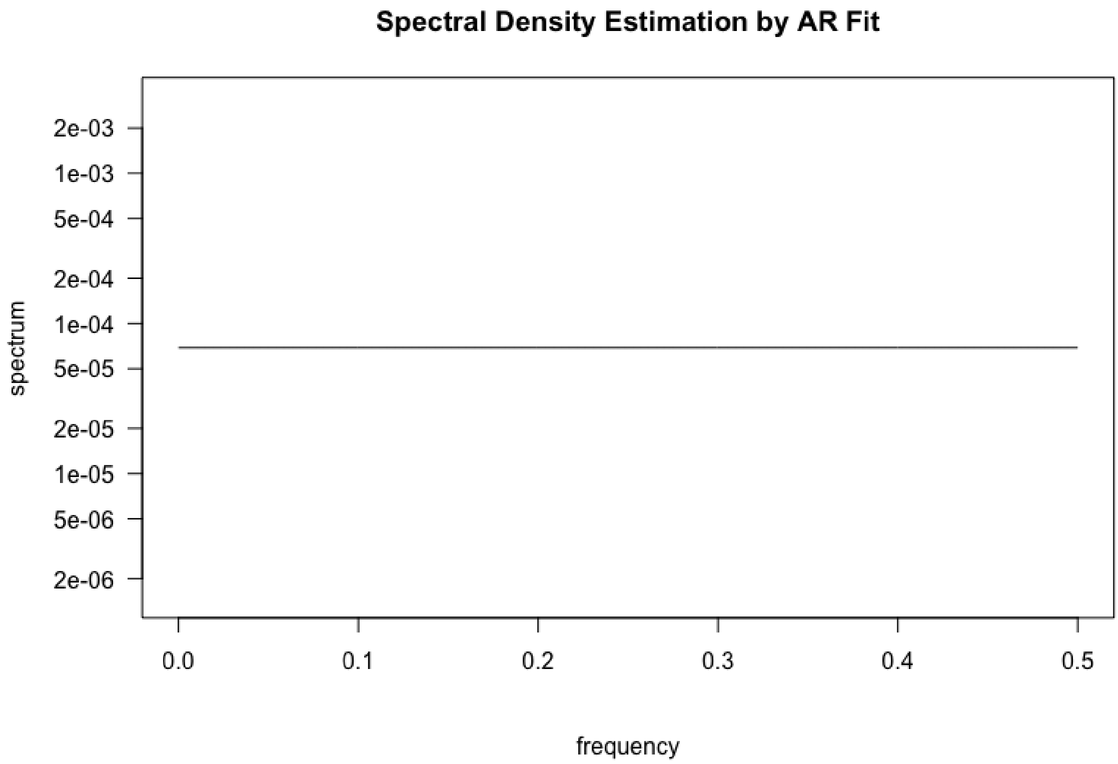


Figure 3.5: Estimated spectral density through an AR fit of DJIA returns

With the thought that cycles might be mixed in with noise due to the financial market closing on weekends and therefore blending the effect of beginnings and endings of a trading week, we apply a moving average filter on the DJIA returns data. Figure 3.6 shows the original DJIA returns series in black and the moving averages in blue. We can see that some of the movements have been smoothed out in the moving average series. By performing spectral analysis on the resulting moving averages instead of the original series, we hope to find cyclical patterns standing out more distinctly. Figure 3.7 is the periodogram showing spectral densities of the moving averages. Although we see different peaks from those in the original series, representing cycles with periods of 25 and 51 days, these cycles still do not prove statistically significant, because the spectral densities did not surpass the mean by the amount represented by the confidence band. Despite their statistical insignificance, these cycles appear to help the model fit. When we remove these cycles and model on the remaining data, an ARMA(2,3) model is found to be the best fit, and this model demonstrates an improvement in AIC.

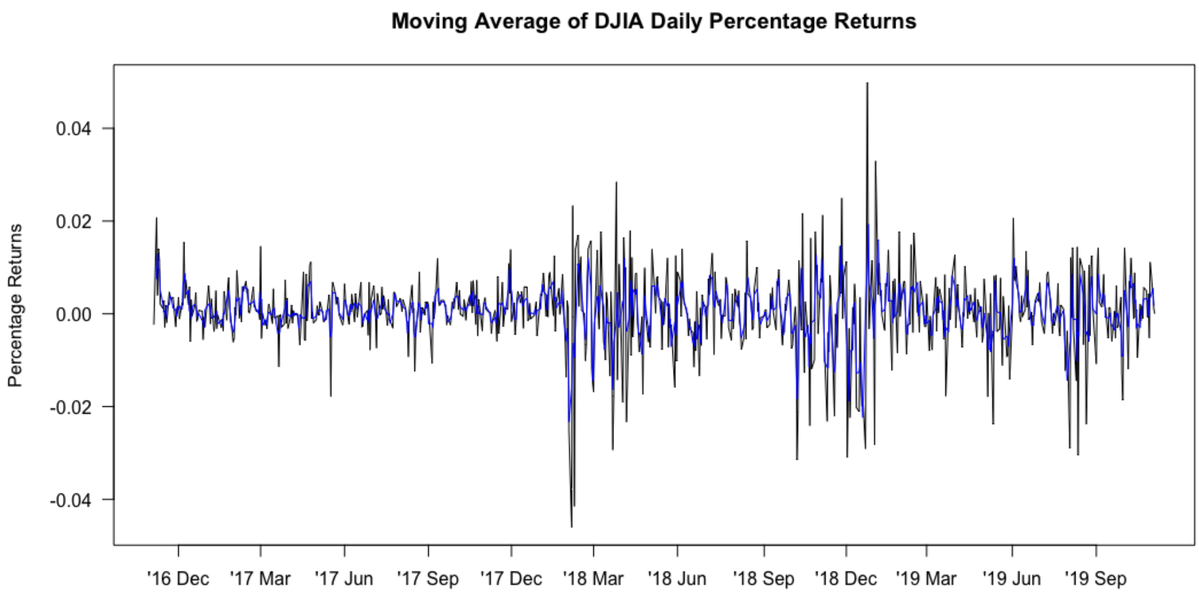


Figure 3.6: DJIA daily percentage returns with moving averages shown in blue

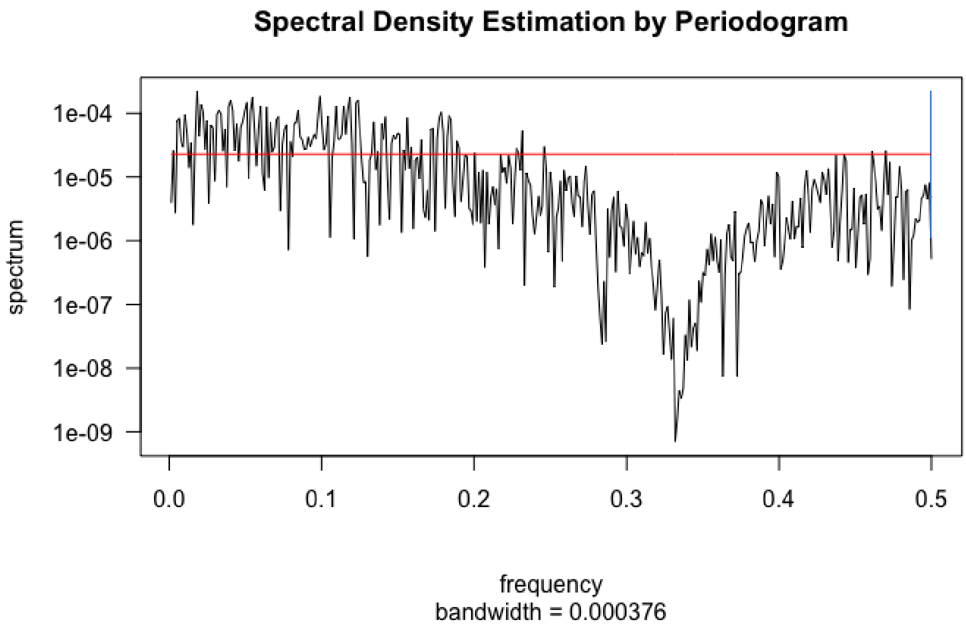


Figure 3.7: Estimated spectral density through a periodogram of moving averages of DJIA returns with the mean and confidence intervals

3.4 Covariates

Another perspective we explore is the potential impact of covariates that might be present in the data. We first take a look at the U.S. Treasury 10-year yield which is known as a data point that the investor population tends to watch closely. Figure 3.8 shows historical data of the DJIA close price and the U.S. Treasury 10-year yield. We transform the covariate series into daily percentage change and create scatterplots of DJIA returns and the covariate's percentage change to investigate potential correlation. Figure 3.9 is a set of 9 scatterplots between the DJIA returns and the percentage change in U.S. Treasury 10-year yield at lags 0 to 8. From the figure, we find a 38% correlation between the DJIA returns and the covariate's percentage change at lag 0.

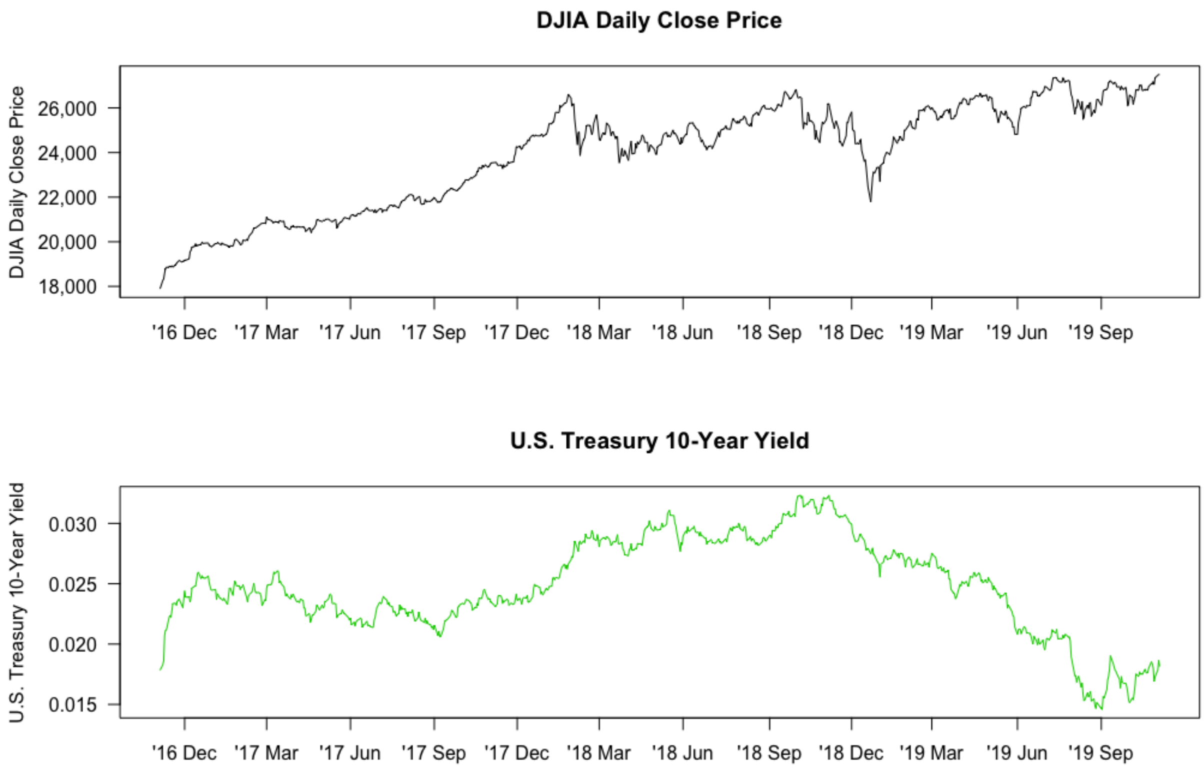


Figure 3.8: Trending of DJIA close price and the U.S. Treasury 10-year yield

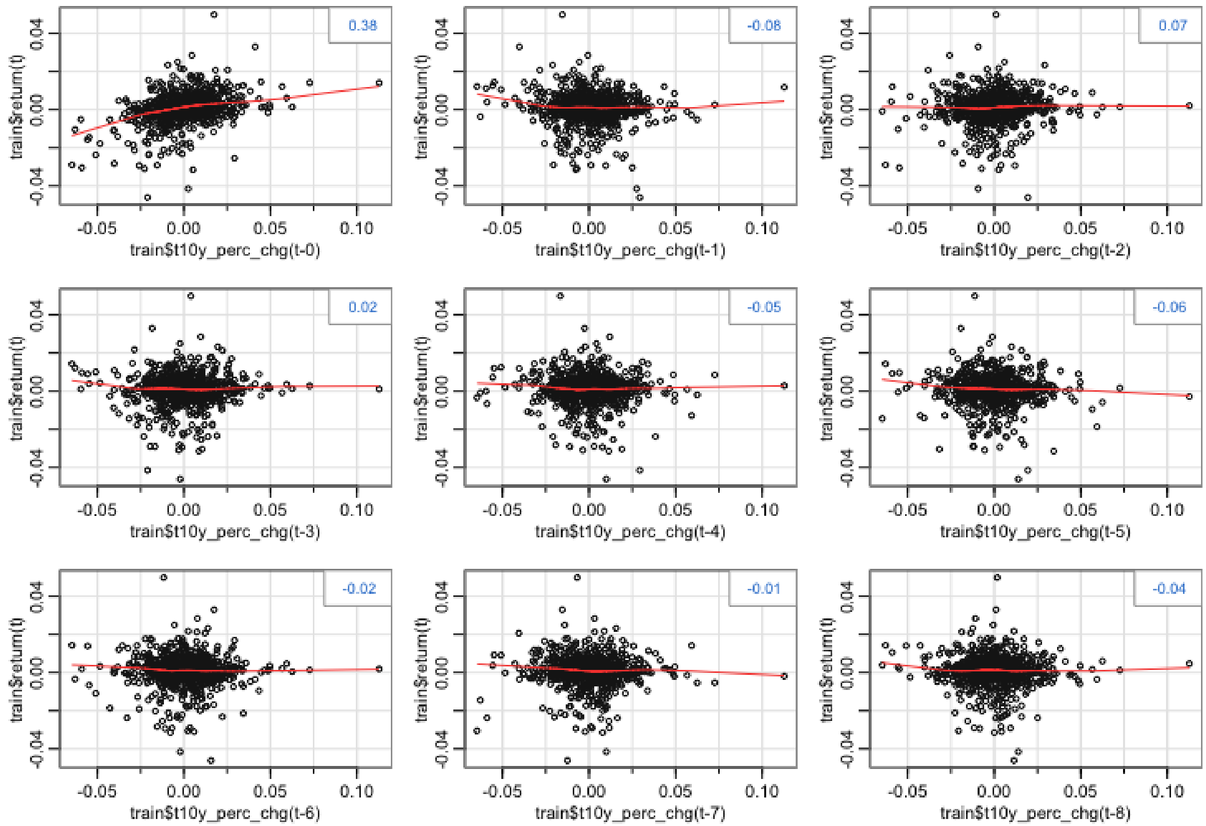


Figure 3.9: Scatterplots of DJIA returns and U.S. Treasury 10-year yield for lags 0 to 8

We then fit a linear regression of DJIA returns on the percentage change of U.S. Treasury 10-year yield. Table 3.1 is a summary showing the linear regression function and a few key metrics for the linear regression model. Looking at the coefficient for the covariate term, we can observe a positive correlation between DJIA returns and the percentage change in the U.S. Treasury 10-year yield. Since the p-value for the coefficient is less than 5%, we conclude that the percentage change of U.S. Treasury 10-year yield is statistically significant in predicting the DJIA returns.

	Coefficient	Std Error	t value	p value
Intercept	0.00057	0.00028	2.03994	0.0417
U.S. Treasury 10Y yield percentage change	0.18396	0.01626	11.31436	<2e-16

Table 3.1: Summary for the linear regression of DJIA returns on U.S. Treasury 10-year yield percentage change

Figure 3.10 shows four diagnosis plots for the linear regression model. We can see that residuals appear close to random and normal, and the leverage plot suggests that there are no influential outliers outside of Cook's distance. To remove the effect of the U.S. Treasury 10-year yield, we take the residuals of the fitted model. In fitting an ARMA model to the residuals, we find that the best ARMA model fit appears to be an ARMA(3,0). Next, we perform spectral analysis, leading us to cycles at periods of 19, 10, and 25 days. We then remove these cycles and find the best fit for the residuals after removing cycles to be an ARMA(1,2).

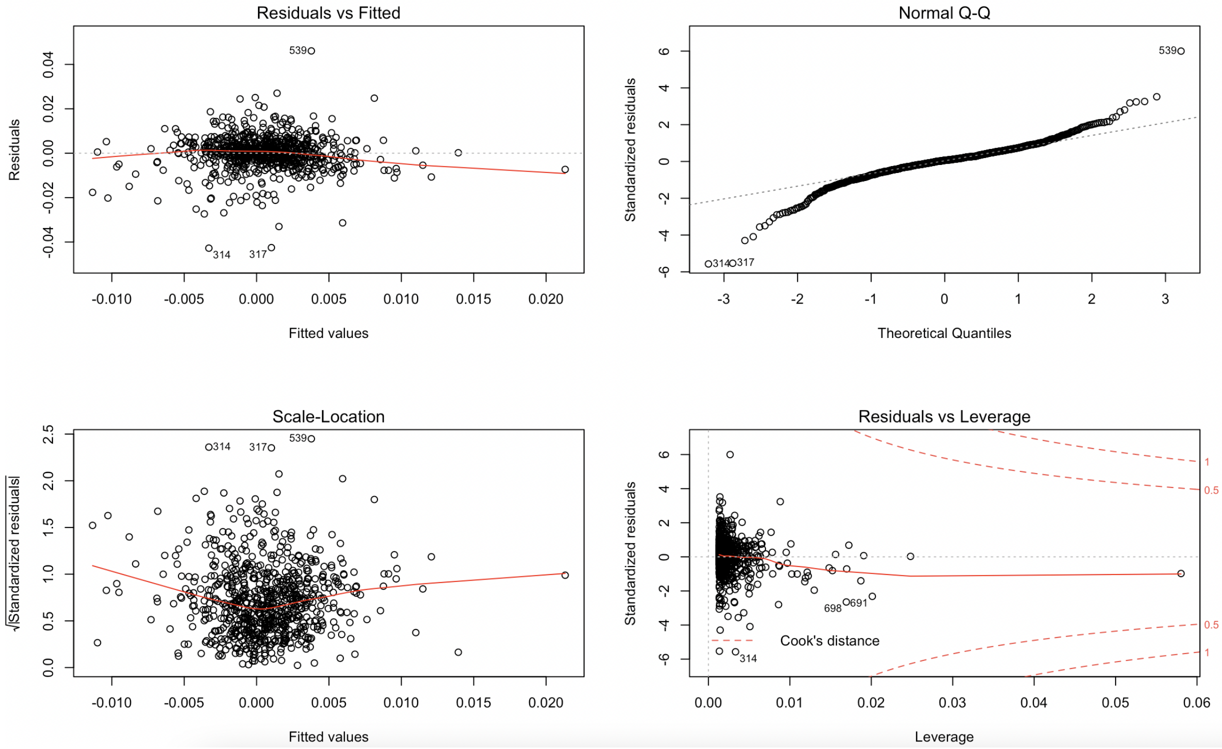


Figure 3.10: Diagnosis plots for the linear regression of DJIA returns on U.S. Treasury 10-year yield percentage change

Other covariates we look into include the unemployment rate and inflation rate in the U.S. We take the similar steps of transforming the covariates into percentage change form and investigating correlation through their lagged scatterplots with DJIA returns. While there is no notable correlation found with percentage change in unemployment rate, we find a correlation between DJIA returns and percentage change in inflation rate. Figure 3.11 shows the DJIA close price and the U.S. inflation rate, and Figure 3.12 shows a set of 9 scatterplots between DJIA returns and the percentage change in U.S. inflation rate at lags 0 to 8. From the scatterplots, we are able to find a correlation at lag 0.

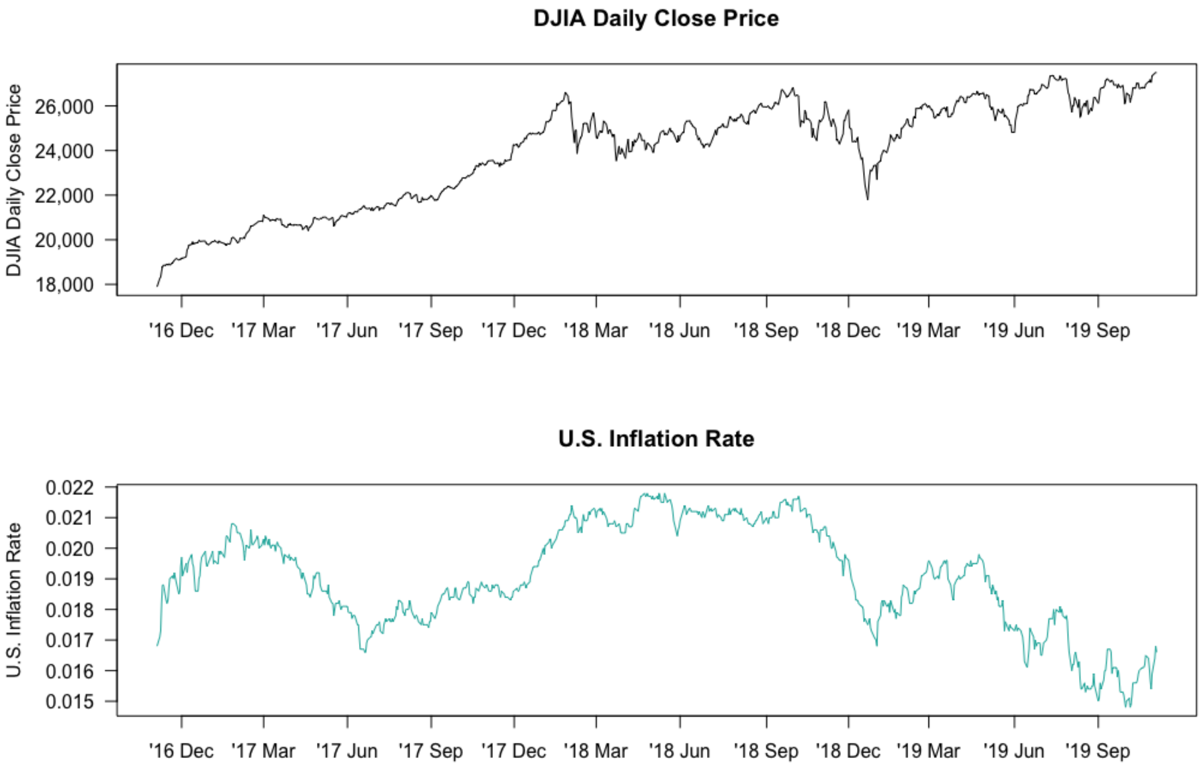


Figure 3.11: Trending of DJIA close price and the U.S. Inflation Rate

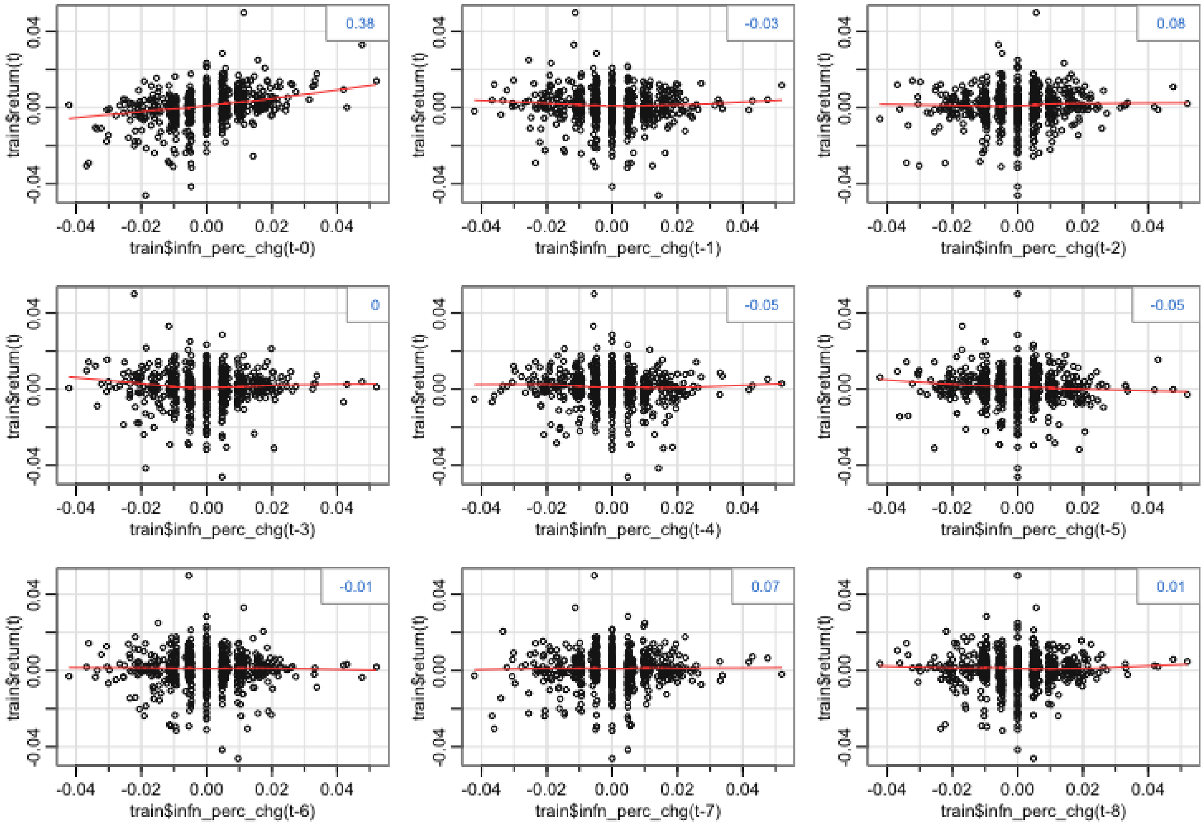


Figure 3.12: Scatterplots of DJIA returns and U.S. inflation rate for lags 0 to 8

Another linear regression for the DJIA returns is fitted on the percentage change of both U.S. Treasury 10-year yield and inflation rate. Figures 3.2 is a summary showing the resulting linear regression function and key metrics of the model. Coefficients for both covariate terms are positive, indicating a positive relationship between the DJIA returns and both covariates. P-values for both coefficients are less than 5%, so the impact of both covariates in predicting the DJIA returns is considered statistically significant.

	Coefficient	Std Error	t value	p value
Intercept	0.00057	0.00027	2.08935	0.03701
U.S. Treasury 10Y yield percentage change	0.11766	0.01976	5.95567	3.97e-09
U.S. inflation percentage change	0.16823	0.02964	5.67594	1.97e-08

Table 3.2: Summary for the linear regression of DJIA returns on U.S. Treasury 10-year yield percentage change and inflation percentage change

Figure 3.13 shows four diagnosis plots for the linear regression model. From the diagnosis plots, we can observe that residuals are close to random and normal, and there are no influential outliers outside of Cook's Distance. We then remove the impact of both covariates from the DJIA returns series and find that the best model fit to the residuals is an ARMA(0,0). After removing cycles identified based on spectral analysis and modeling on the remaining data, we again find the best fitting model to be an ARMA(0,0).

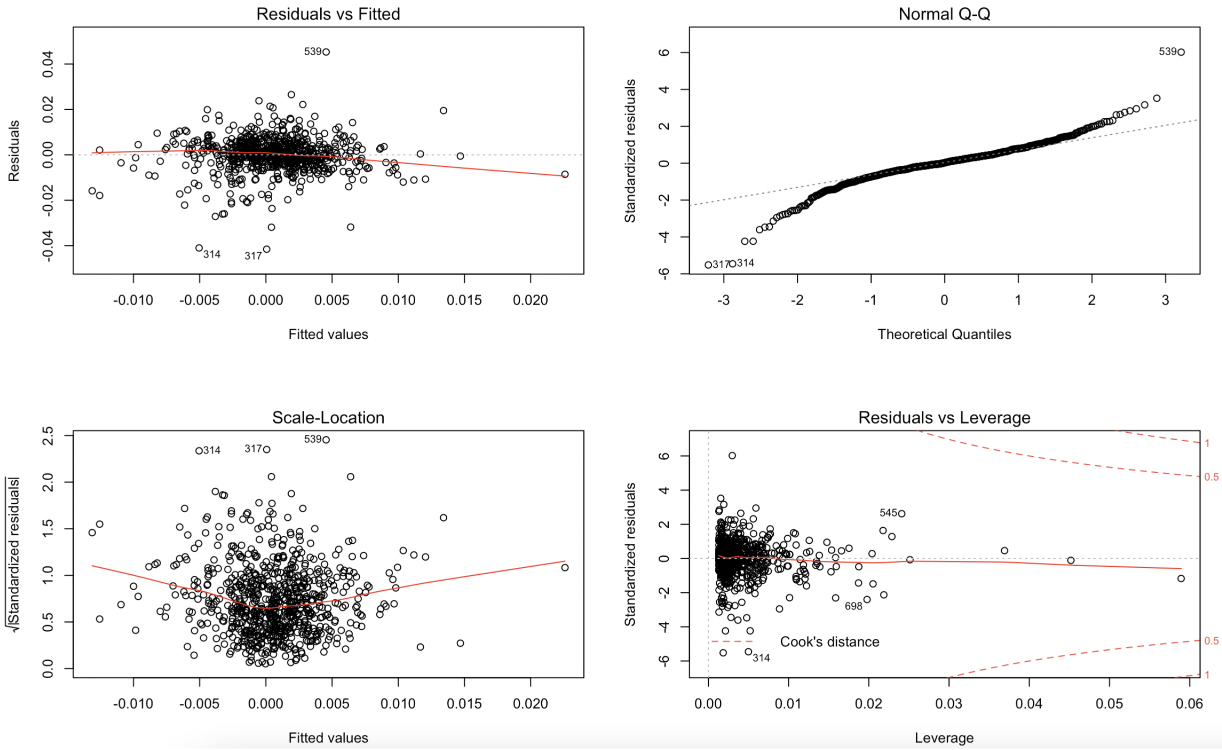


Figure 3.13: Diagnosis plots for the linear regression of DJIA returns on U.S. Treasury 10-year yield percentage change and inflation percentage change

3.5 Description of Models

Table 3.3 summarizes the models that are built in this study and description of the data each model is fitted on. The next section will go into metrics we gather on each model.

	Model	Description
1.1	ARMA(0,0)	DJIA Returns
1.2	ARMA(2,3)	DJIA Returns, cycles removed
2.1	ARMA(3,0)	DJIA Returns, U.S. Treasury 10-year yield percentage change removed
2.2	ARMA(1,2)	DJIA Returns, U.S. Treasury 10-year yield percentage change, and cycles removed
3.1	ARMA(0,0)	DJIA Returns, U.S. Treasury 10-year yield and inflation percentage change removed
3.2	ARMA(0,0)	DJIA Returns, U.S. Treasury 10-year yield and inflation percentage change, and cycles removed

Table 3.3: Summary of All Models

CHAPTER 4

Results

In this section we will examine the six models by looking at their model diagnoses and measuring their predicting abilities to help decide which model(s) might be the best at capturing the patterns within the DJIA index data.

4.1 Diagnosis

Generally, there is a progression of improving AIC as we go from Model 1.1 to Model 3.2, with Model 3.2 having the best AIC. From the perspective of model residuals, Models 1.1, 1.2, 2.1 and 2.2 pass the Ljung-Box test for residuals, with their p-values above 5%. However, Models 3.1 and 3.2, which have the impact of inflation incorporated, do not pass the Ljung-Box test. Figures 4.1 and 4.2 are the diagnosis charts of Models 3.1 and 3.2, respectively. The histograms show that the residuals are close to normal. However, in their residual ACF charts, there are peaks at lag 2, suggesting that there might be patterns in the residuals that remain to be captured. We explore this by trying models such as AR(2), MA(2) and ARMA(2,2). While they flatten out the peak at lag 2 in the ACF chart, they do not provide improvement from Models 3.1 and 3.2 in terms of AIC and prediction errors, which indicates that incorporating the autocorrelation at lag 2 could be overfitting.

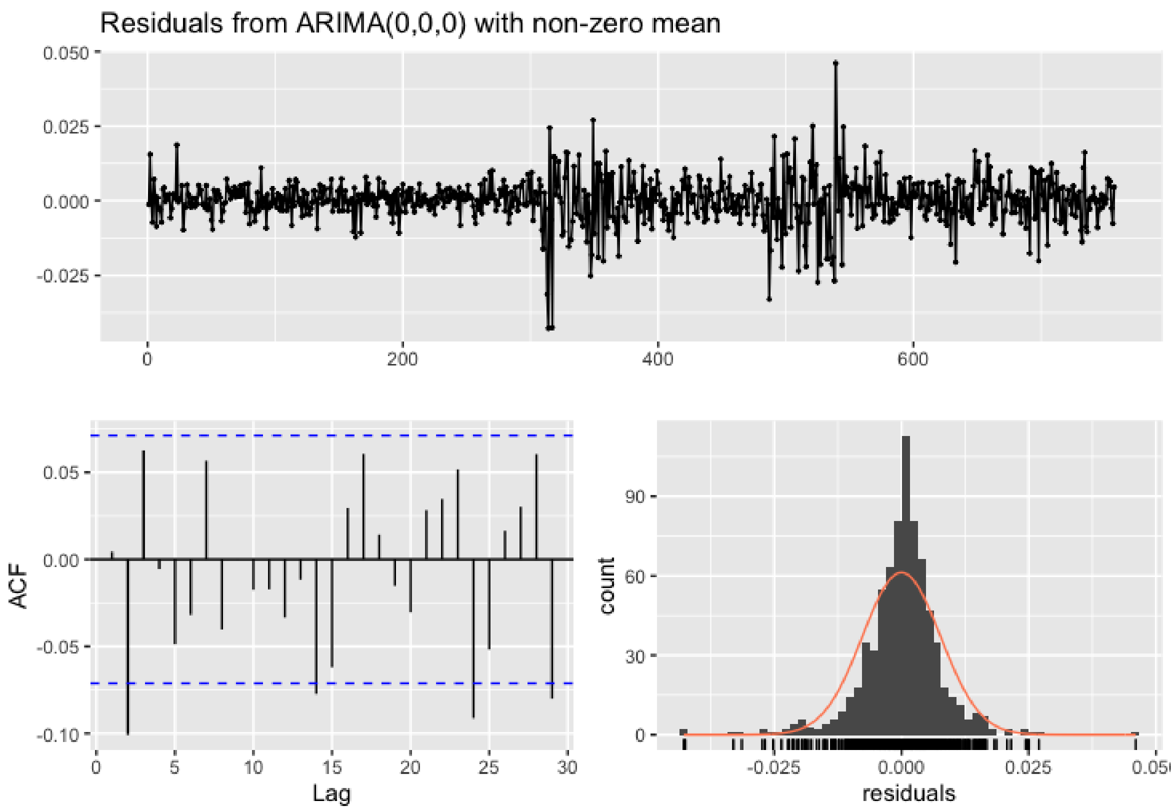


Figure 4.1: Diagnosis charts for model 3.1 (DJIA returns with U.S. Treasury 10-year yield and inflation percentage change removed)

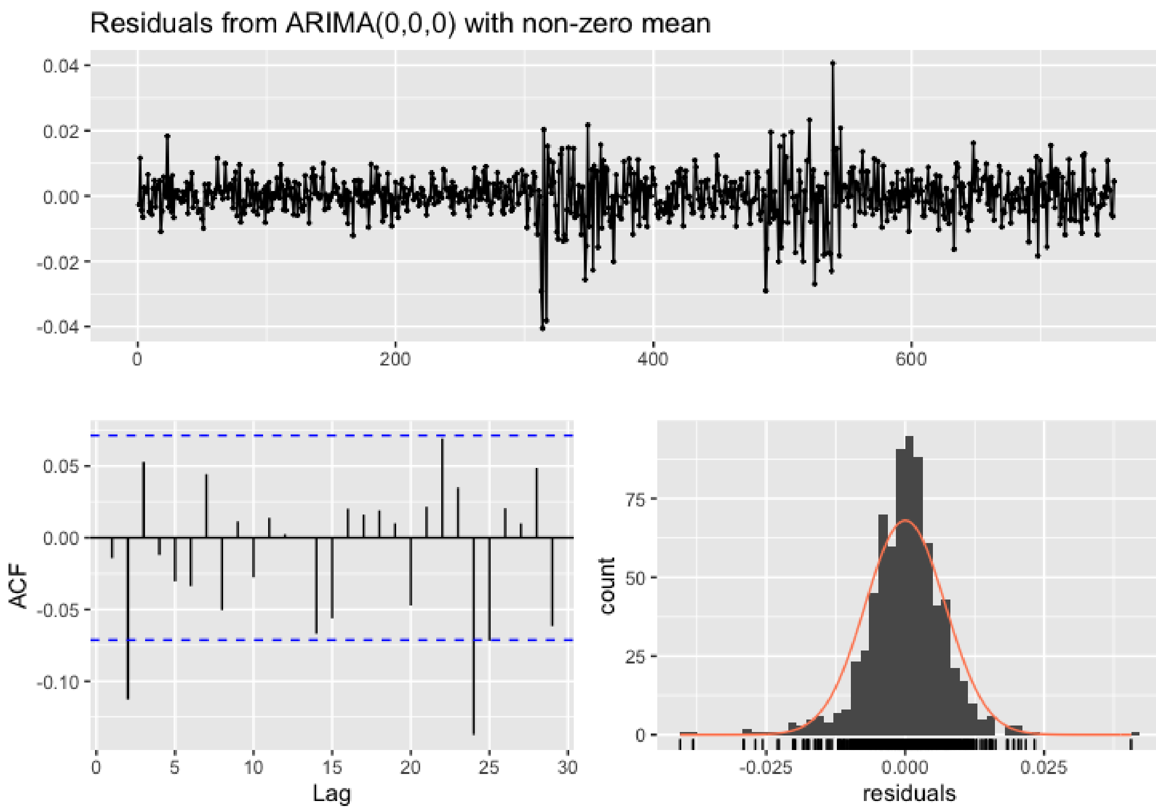


Figure 4.2: Diagnosis charts for model 3.2 (DJIA returns with U.S. Treasury 10-year yield and inflation percentage change, and cycles removed)

4.2 Making Predictions

Additionally, we utilize the models to create predictions of the DJIA close price. For each day in the testing period, we make a one-day prediction with the latest information, add in necessary elements such as cycles and covariates to arrive at the predicted DJIA returns for the day, and then multiply it by DJIA close price on the previous day to produce the predicted close price for the day. Figures 4.3, 4.4 and 4.5 show the predicted DJIA close prices for each model, against actual prices the DJIA index closed at. Actual DJIA close prices are shown in black, and the predicted DJIA close prices are shown in each model's own color. The predictions in general follow actual prices closely and look relatively similar, so we look at prediction errors next.

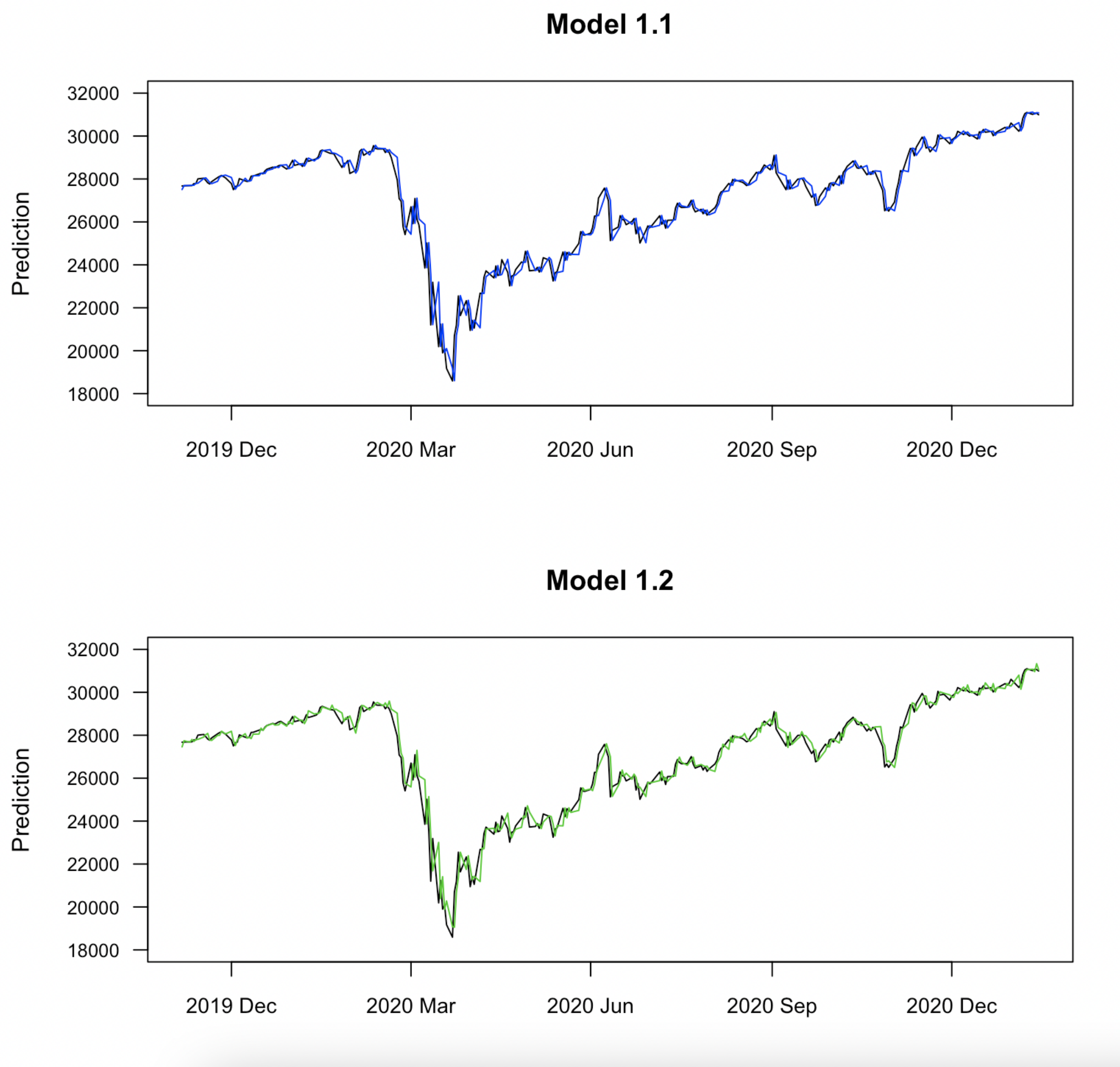


Figure 4.3: Models 1.1 and 1.2: predicted DJIA close prices vs actual DJIA close prices

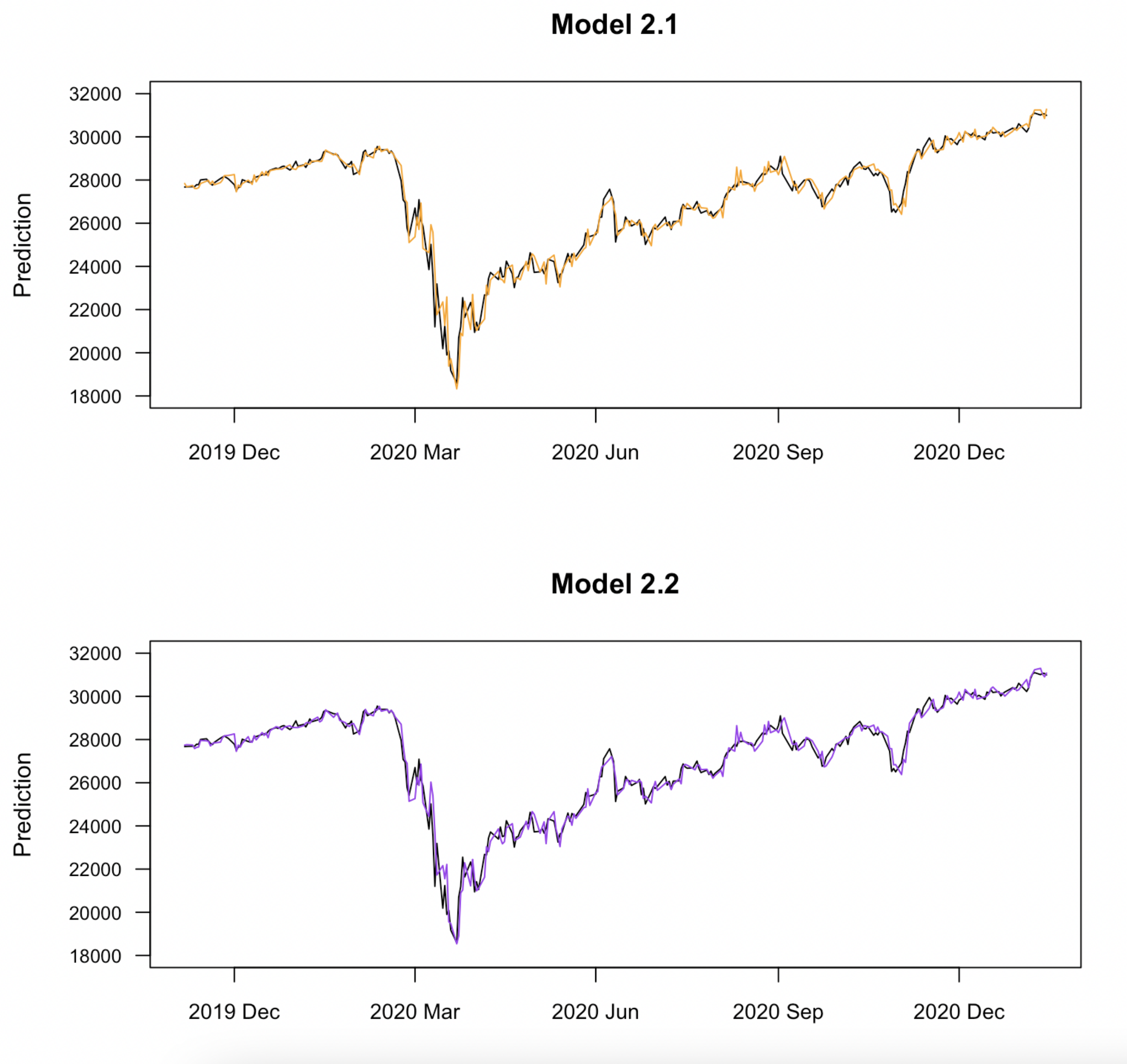


Figure 4.4: Models 2.1 and 2.2: predicted DJIA close prices vs actual DJIA close prices

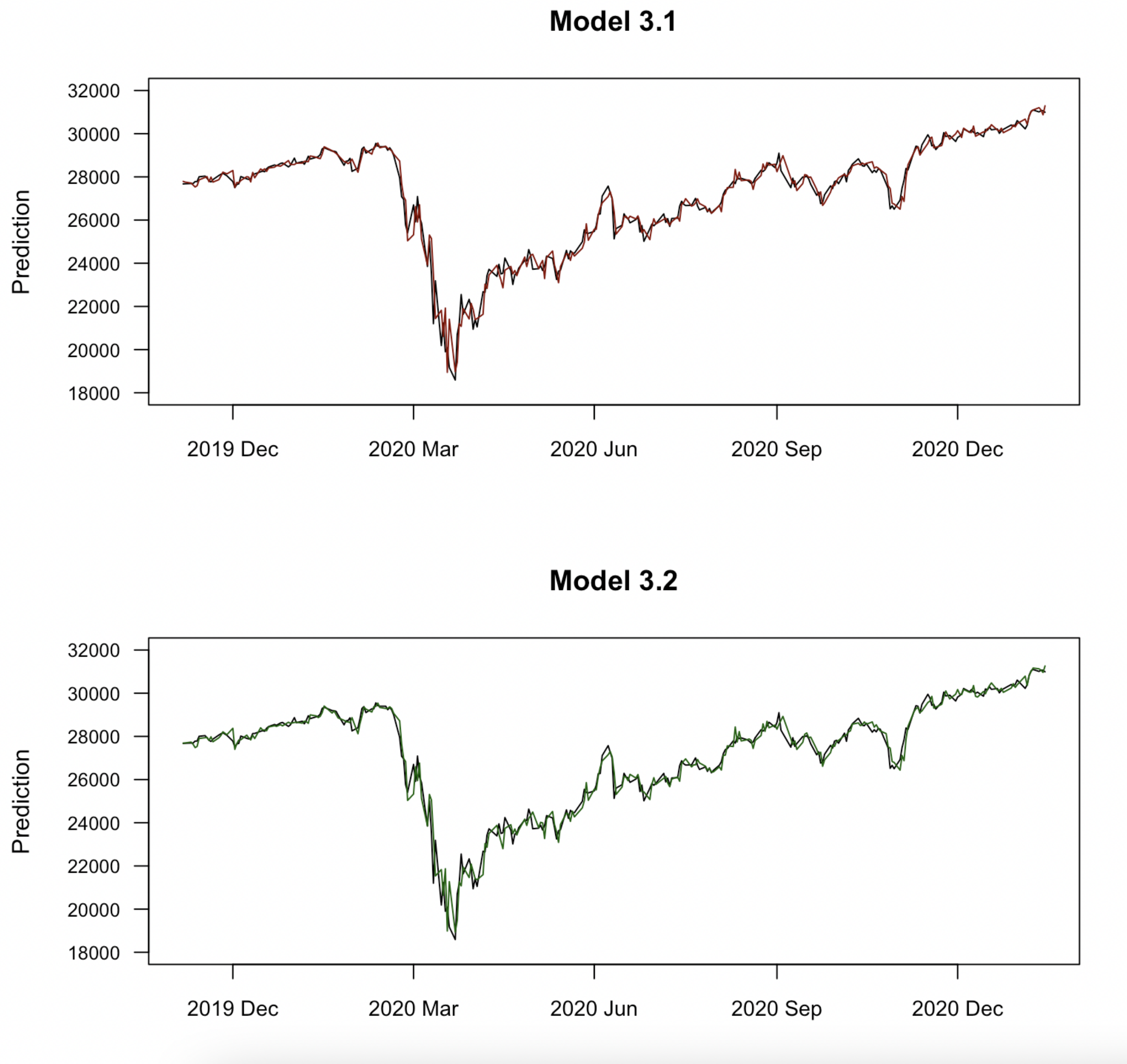


Figure 4.5: Models 3.1 and 3.2: predicted DJIA close prices vs actual DJIA close prices

We calculate the prediction errors by taking the difference between predicted and actual DJIA close prices for each model. Then, from the prediction errors we calculate the root mean squared errors (rMSE) to represent each model's predicting accuracy. Figures 4.6, 4.7 and 4.8 show the prediction errors across the testing period for each model. The rMSE values are also noted on the corresponding graphs. Generally, we observe that the errors appear to be at a relatively consistent level until March 2020 when the markets crashed because of news on the COVID-19 pandemic. The model predictions suffer increased errors during the COVID-19 crash, which phase out gradually and return to pre-pandemic levels towards the second half of 2020.

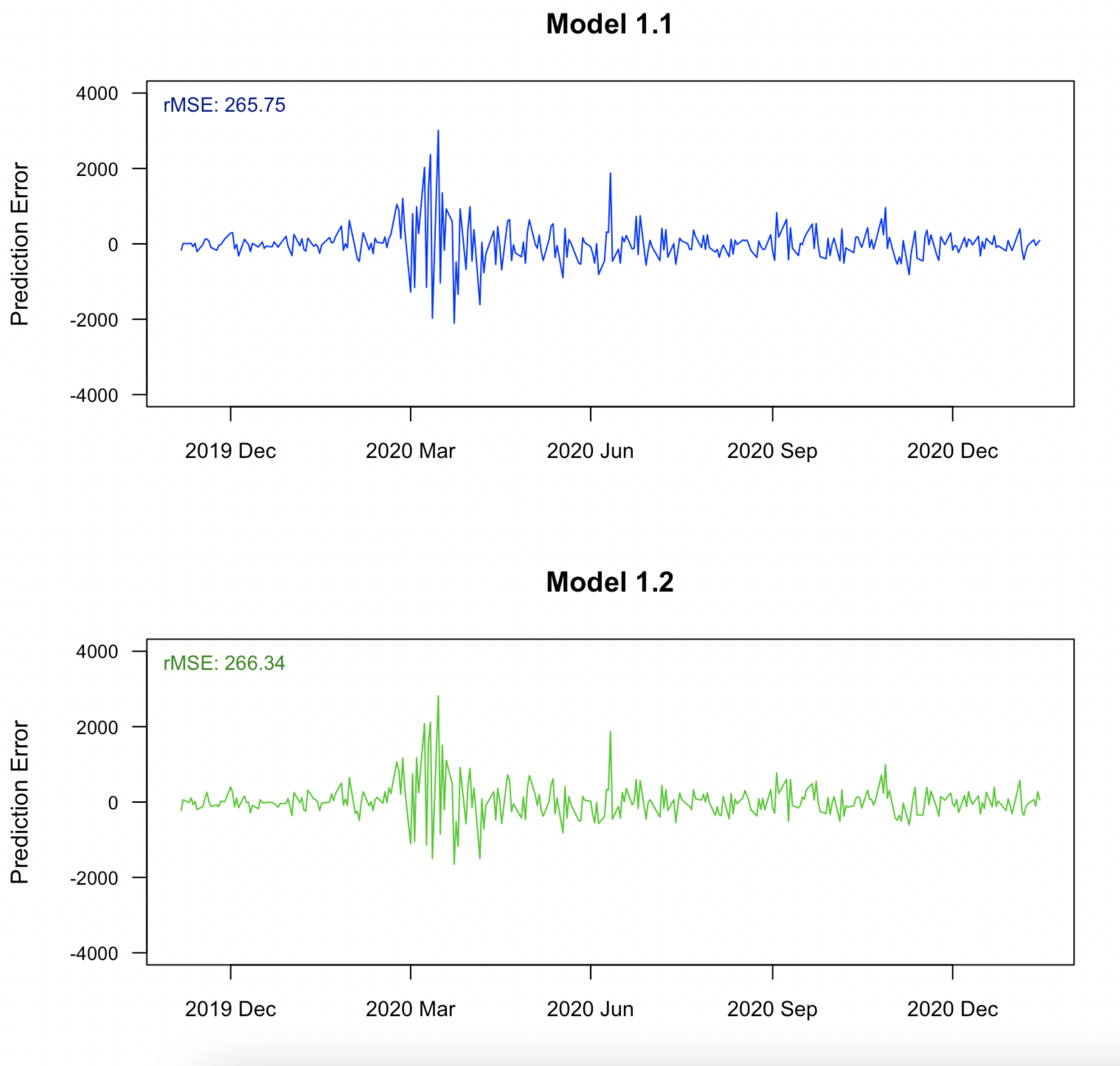


Figure 4.6: Models 1.1 and 1.2: prediction errors

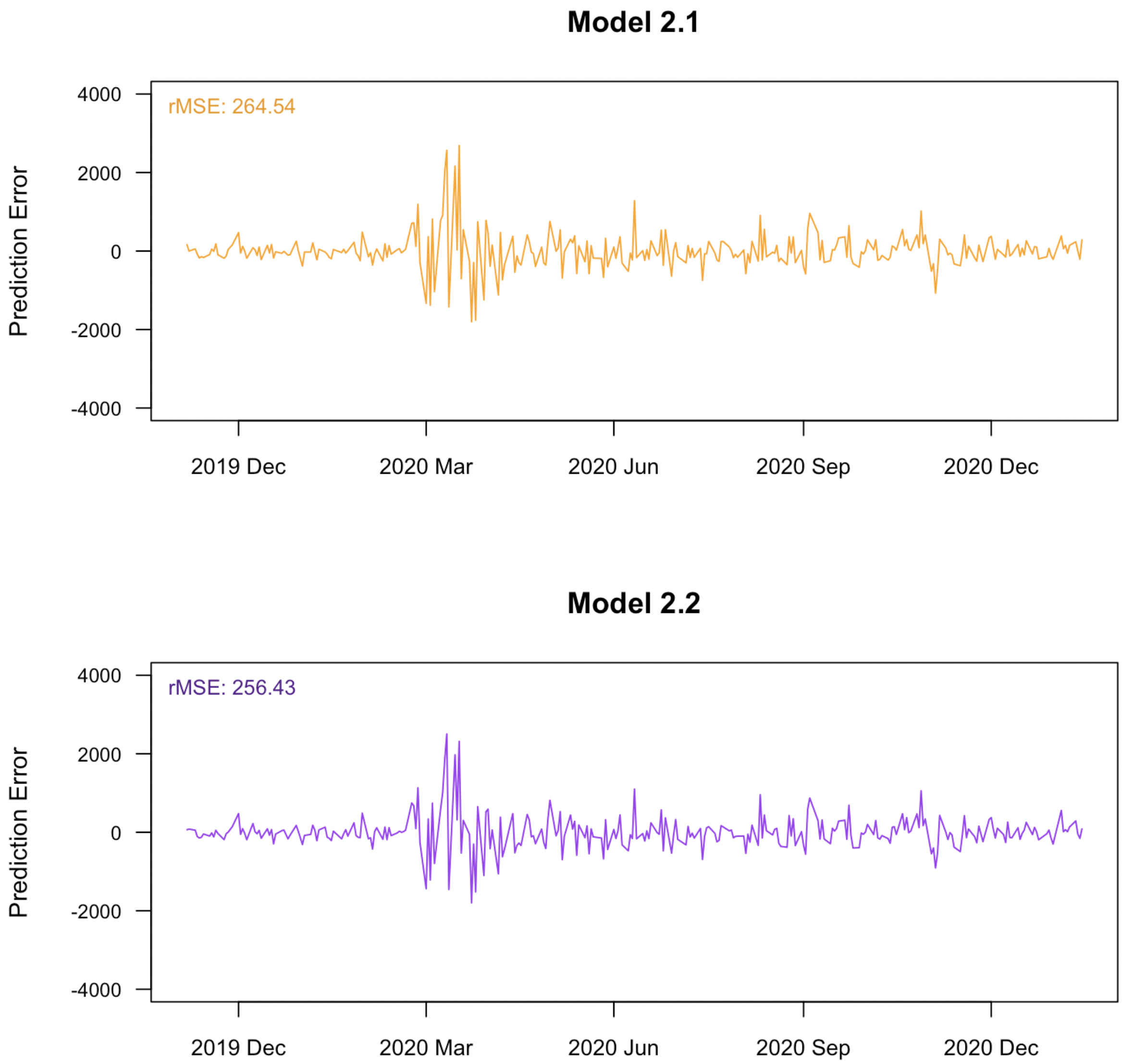


Figure 4.7: Models 2.1 and 2.2: prediction errors

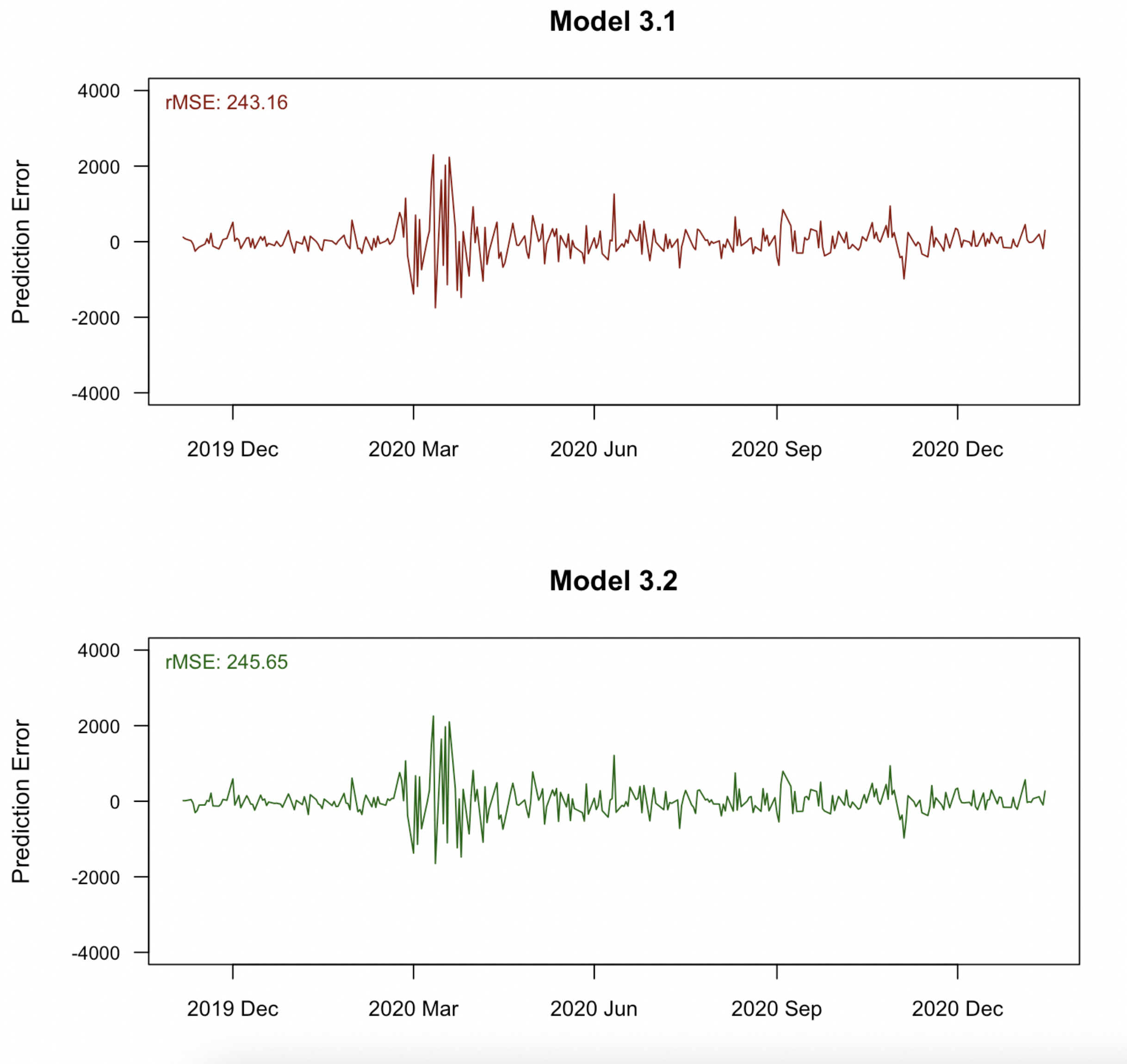


Figure 4.8: Models 3.1 and 3.2: prediction errors

4.2.1 Predictions During the COVID-19 Crash

In the period of increased errors, we want to further understand how the models perform relatively to each other. We do this by taking a more focused look at the period of the COVID-19 crash between February 25, 2020 and March 24, 2020. Figures 4.9, 4.10 and 4.11 show for each model the predicted DJIA close prices against actual DJIA close prices during the stated COVID-19 crash period, including approximately one month leading up to it and one month afterwards. The sets of two red dotted lines in the graphs mark the stated period of COVID-19 crash. We can observe the gap between predicted and actual values widening during the crash for all six models.

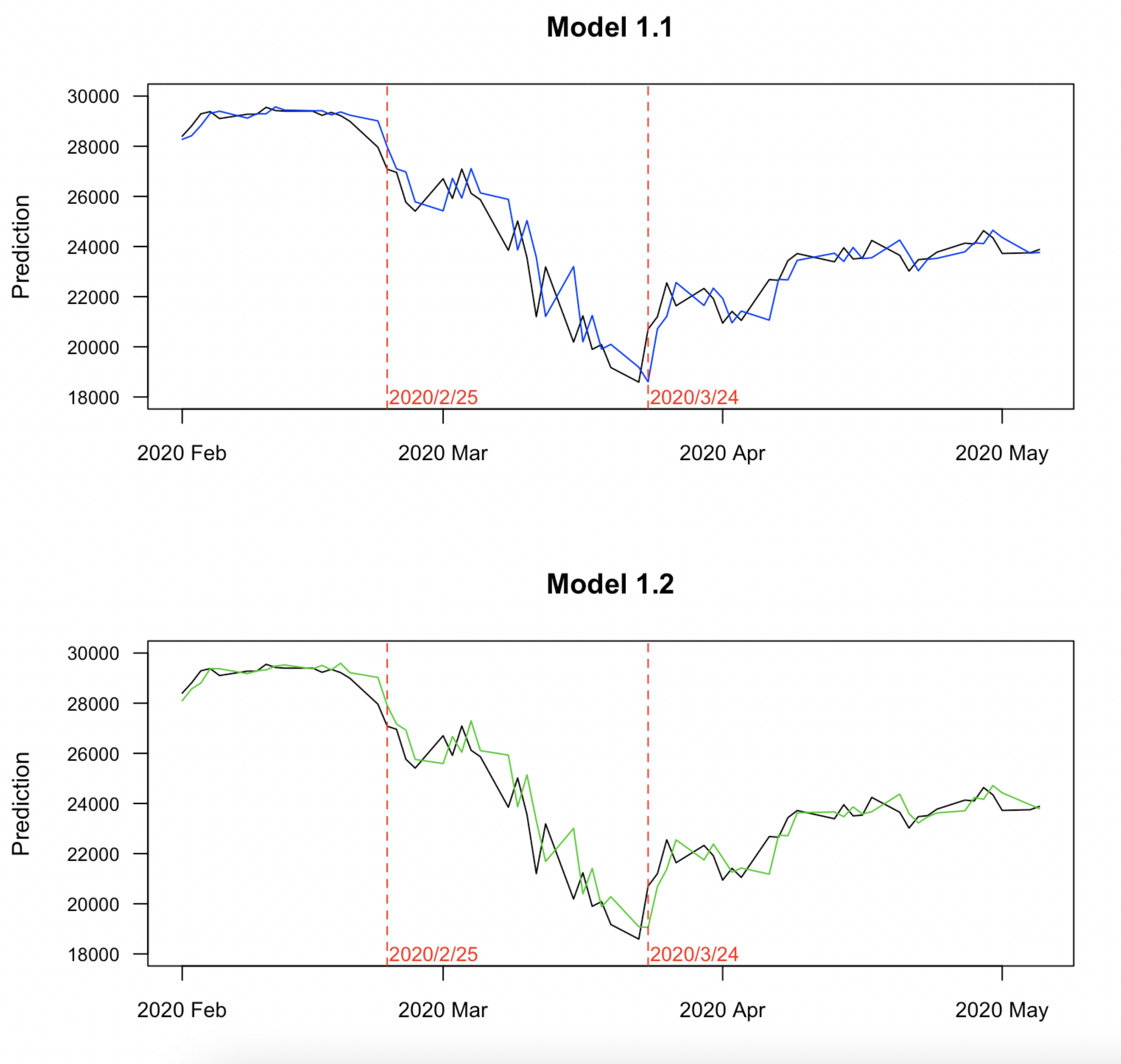


Figure 4.9: Models 1.1 and 1.2: predicted DJIA close prices vs actual DJIA close prices, during COVID-19 crash

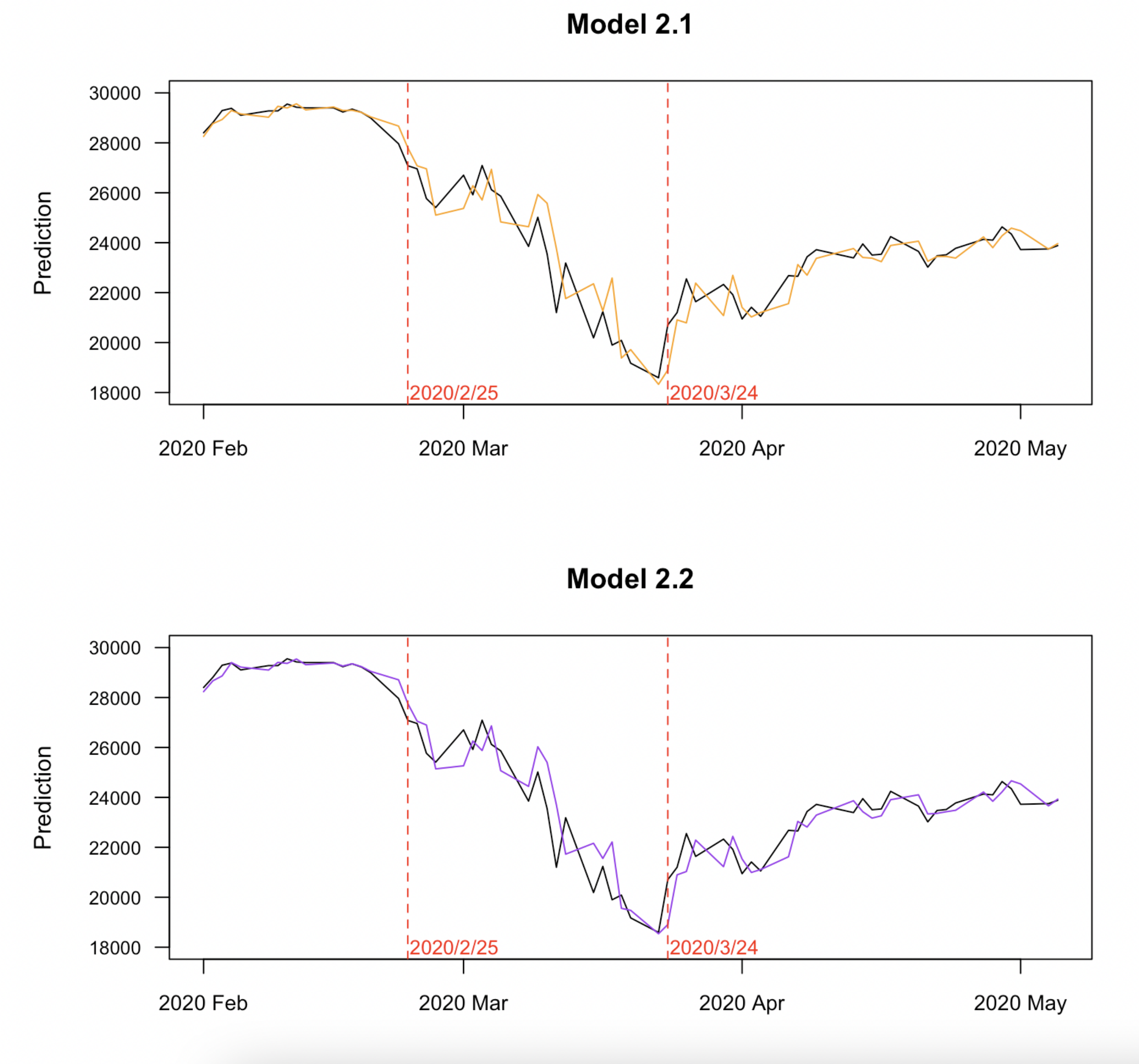
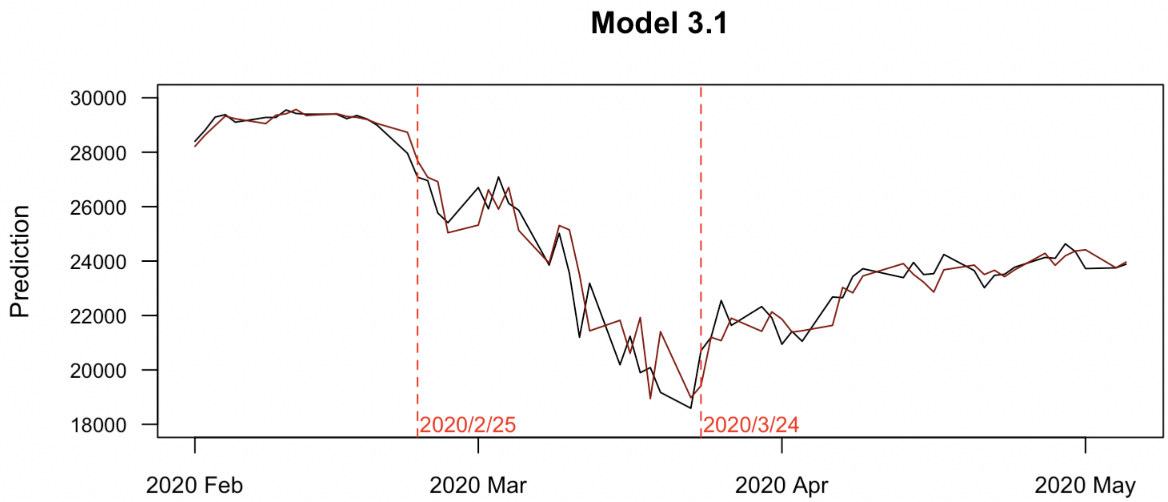


Figure 4.10: Models 2.1 and 2.2: predicted DJIA close prices vs actual DJIA close prices, during COVID-19 crash



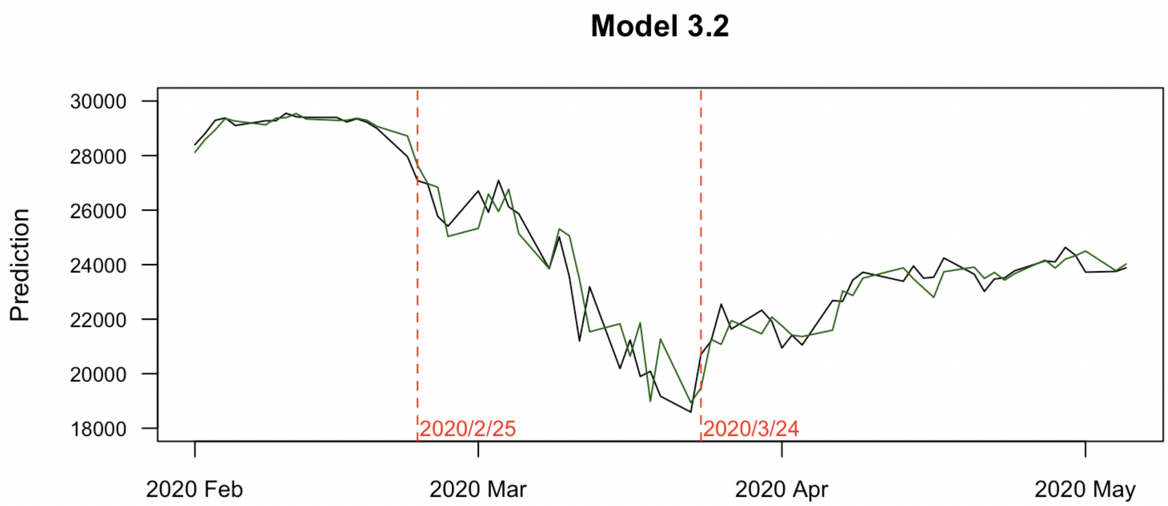


Figure 4.11: Models 3.1 and 3.2: predicted DJIA close prices vs actual DJIA close prices, during COVID-19 crash

Figures 4.12, 4.13 and 4.14 illustrate the prediction errors during COVID-19 crash period, similarly including one month leading up to it and one month afterwards. To measure the predicting accuracy of each model during this time, we calculate the rMSE for only those 29 days in the aforementioned COVID-19 crash period marked by the red dotted lines. The COVID-19 crash rMSE exceeds 1000 for all six models, unsurprisingly higher than rMSEs for the overall testing period which are in the 200s.

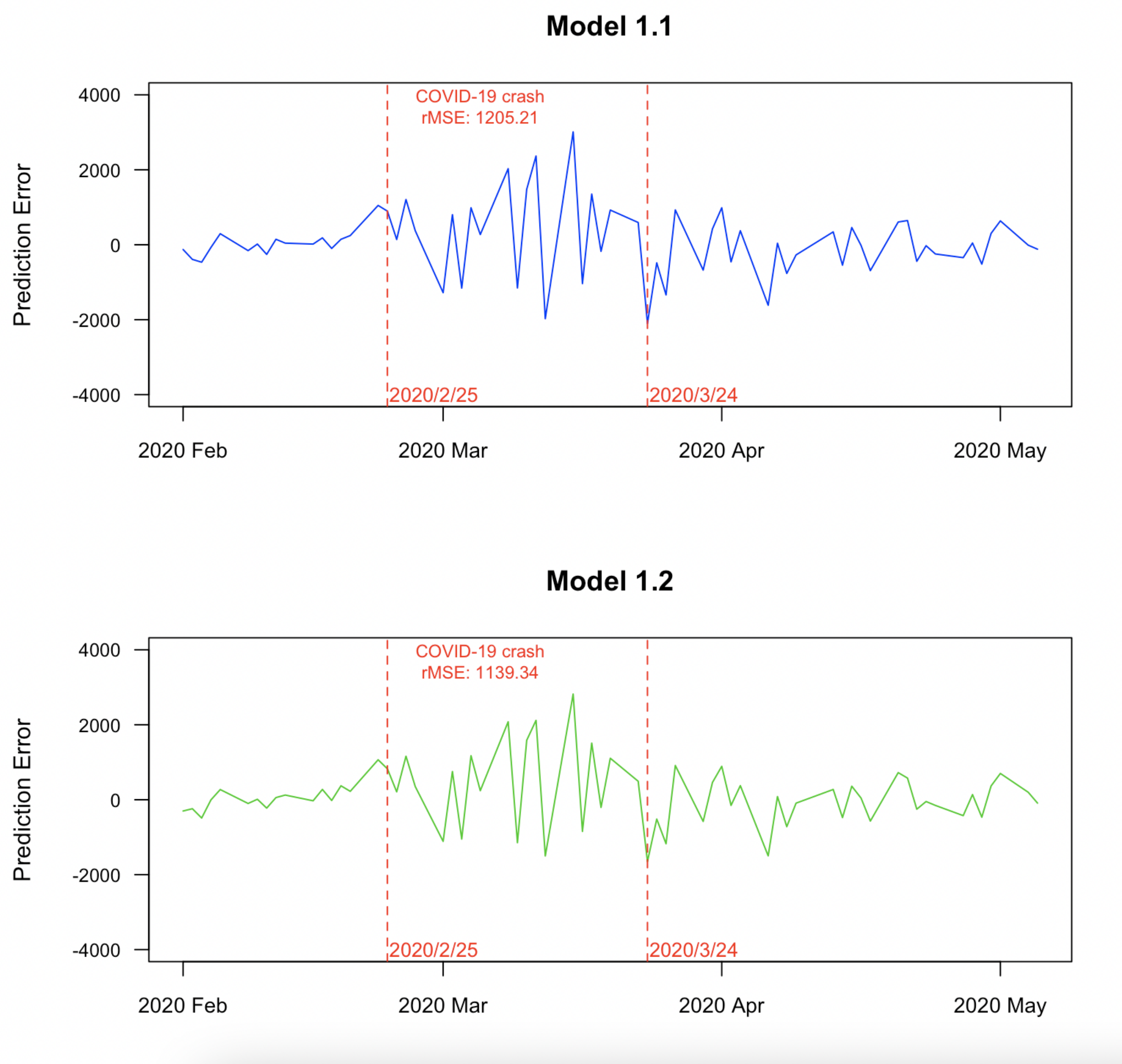


Figure 4.12: Models 1.1 and 1.2: prediction errors, during COVID-19 crash

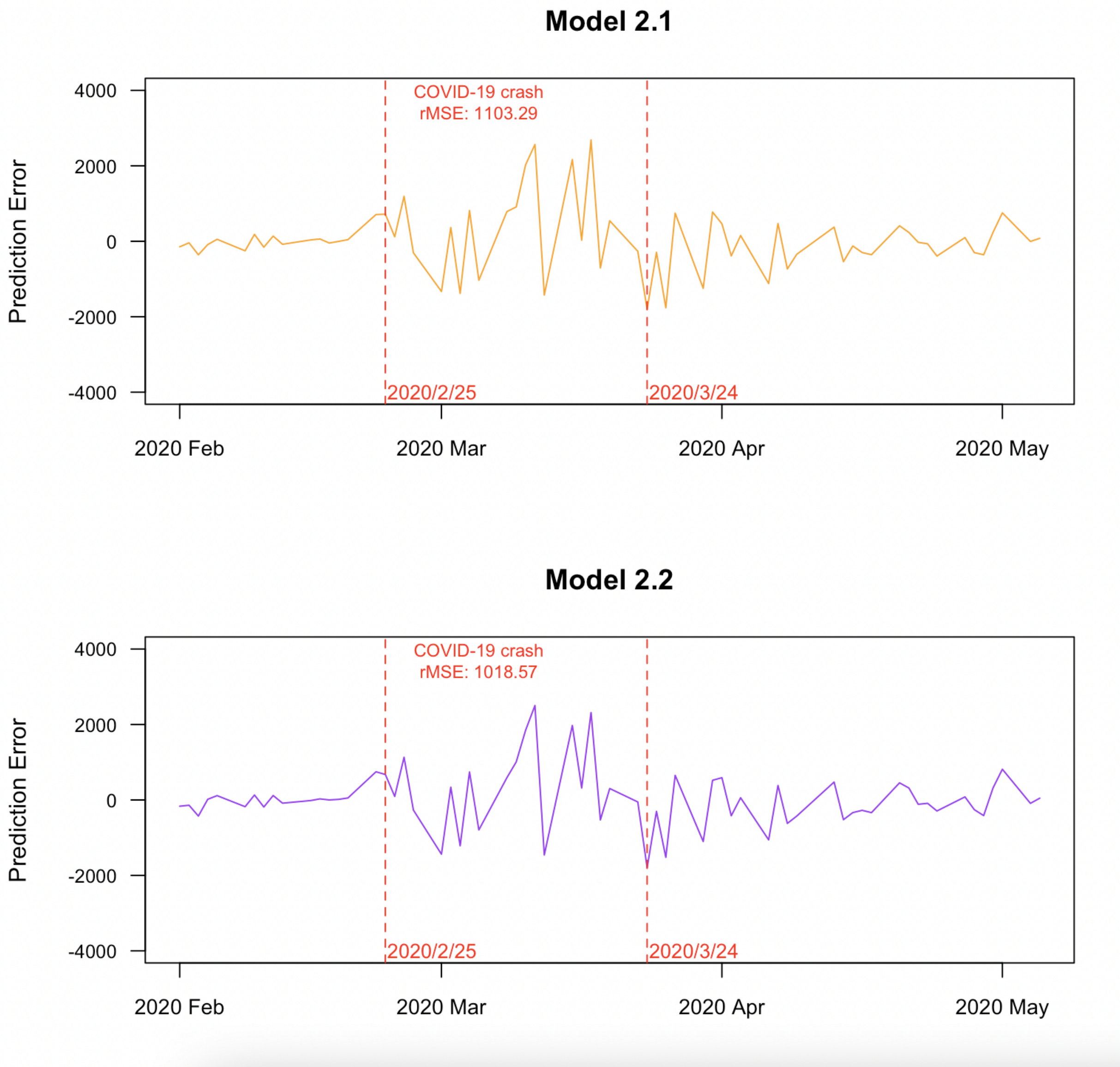


Figure 4.13: Models 2.1 and 2.2: prediction errors, during COVID-19 crash

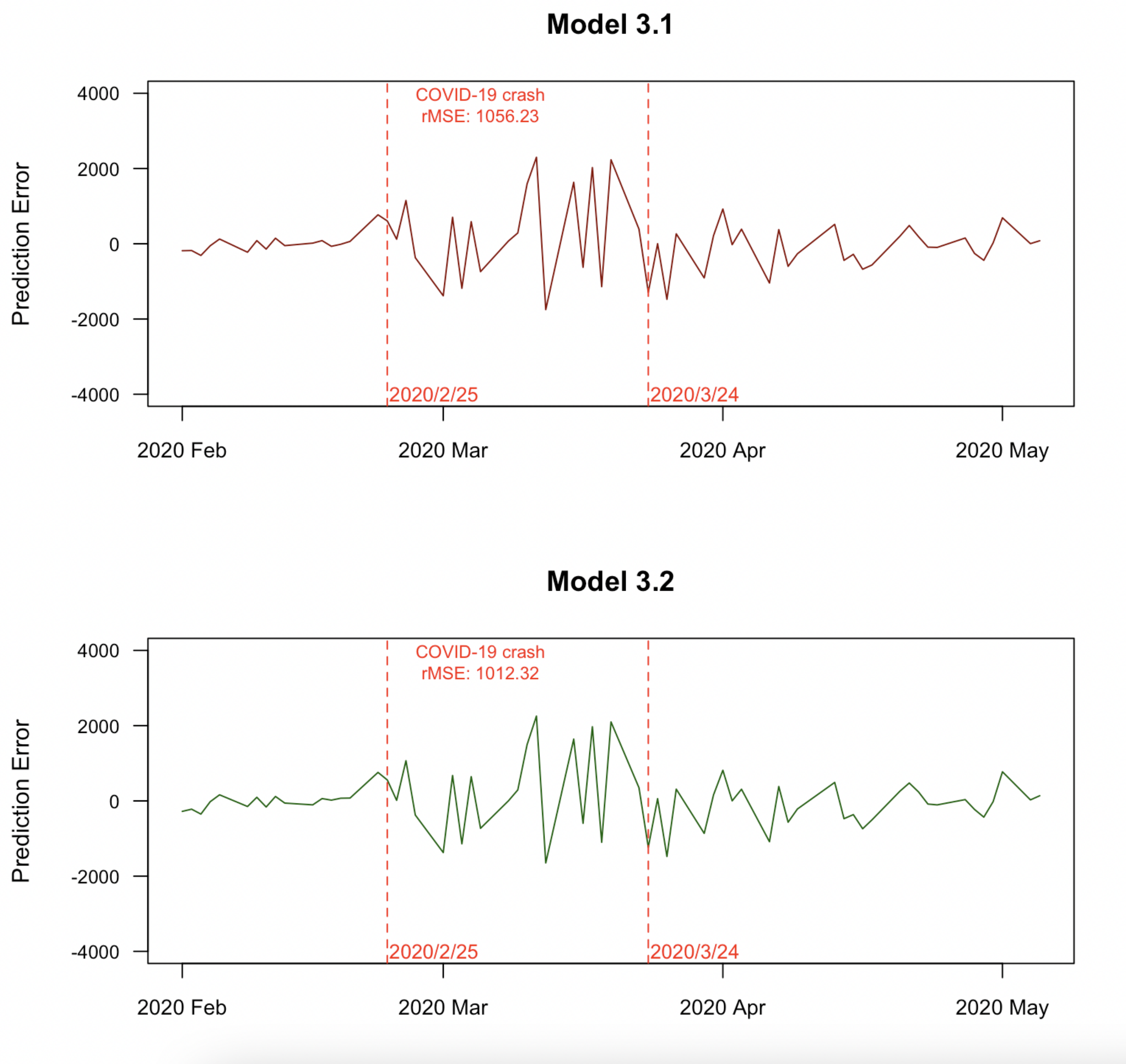


Figure 4.14: Models 3.1 and 3.2: prediction errors, during COVID-19 crash

4.3 Model Comparison and Recommendation

Table 4.1 shows the measurements of each model's fit and predicting accuracy, including model AIC, p-values from the Ljung-Box test of residuals, and rMSE for predictions on the overall testing period and the COVID-19 crash period.

	Model	Description	AIC	Ljung-Box p-value	Test rMSE	Test rMSE during COVID
1.1	ARMA(0,0)	DJIA Returns	-5101	0.177	265.75	1205.21
1.2	ARMA(2,3)	DJIA Returns, cycles removed	-5176	0.193	266.34	1139.34
2.1	ARMA(3,0)	DJIA Returns, U.S. Treasury 10-year yield percentage change removed	-5225	0.329	264.54	1103.29
2.2	ARMA(1,2)	DJIA Returns, U.S. Treasury 10-year yield percentage change, and cycles removed	-5270	0.558	256.43	1018.57
3.1	ARMA(0,0)	DJIA Returns, U.S. Treasury 10-year yield and inflation percentage change removed	-5220	0.044	243.16	1056.23
3.2	ARMA(0,0)	DJIA Returns, U.S. Treasury 10-year yield and inflation percentage change, and cycles removed	-5316	0.038	245.65	1012.32

Table 4.1: Summary of Performance Metrics of All Models

As previously mentioned, progressing from Model 1.1 to Model 3.2, we see an improvement in AIC. Even though the cycles discovered from spectral analysis do not appear statistically significant, removing them evidently help the models perform better. In the three times we remove cycles (going from Model 1.1 to 1.2, from 2.1 to 2.2, and from 3.1 to 3.2), there is consistent improvement in AIC. We can also observe that removing the effect of covariates notably improves both AIC and rMSE. The remaining series after we remove these elements fits best to an ARMA(0,0), suggesting that the remainder is close to a white noise series and that the effect of these covariates and cycles appear to help explain a lot of the patterns in the data.

On the far right of Table 4.1, rMSE during the time of the COVID-19 crash serves as a reference for us to understand how well these models predict compared to each other during a time of uncertainty. We can see that the rMSE during the COVID-19 crash is higher than the rMSE for the overall testing period, which is also observed in Figures 4.6, 4.7 and 4.8. It is worth noting that Models 2.2 and 3.2 appear to perform best during the COVID-19 crash period according to this metric. Both models deliver improvements in AIC and COVID-19 rMSE after the implementation of cycles, again suggesting that despite their lack of statistical significance, the cycles found in spectral analysis appear to contribute to model performance.

Based on the key metrics for model fit and predicting abilities shown in table 4.1, we recommend Models 3.1 and 3.2, because they incorporate the impact of two covariates that improve both AIC and rMSE by notable amounts. Residuals of these models do not pass the Ljung-Box test, but as stated in the Diagnosis section earlier, in incorporating the lag 2 element observed in the residuals, we discover that the resulting models do not perform better than Models 3.1 and 3.2. It would appear that despite the presence of patterns in the residuals, there is limited benefit for the models to include those observed patterns. As a result, we still recommend Model 3.1 and Model 3.2. Between these two models, the performance is relatively close with each of them having a little bit of advantage in different areas. While Model 3.1 delivers slightly more accurate predictions overall, Model 3.2 provides a better AIC and more accurate predictions during times of uncertainty.

CHAPTER 5

Conclusion

In sum, as we find the best fit to DJIA returns to be an ARMA(0,0), we confirm that the DJIA returns is similar to a white noise series, which suggests that the DJIA close prices appear to be close to a summation of white noise, therefore a random walk. In this study, we examine the models' fit by looking at AIC values and their predicting abilities by comparing each model's rMSE on testing data. The models remain unable to beat a white noise model in rMSE. This tells us that even after the incorporation of different elements found in the series, the resulting models do not predict the market returns better than a simple white noise model. This observation on the DJIA index is in agreement with the notion suggested by previous research that the stock market follows the random walk hypothesis [GM63][GGM64].

With that said, some findings in this study remain meaningful. Specifically, changes in macroeconomic variables such as the U.S. Treasury 10-year yield and inflation rate are found to have statistically significant relationships with the DJIA returns. Furthermore, incorporating them notably improves model performance in terms of both model fit and prediction errors on the testing data. Through these results on the DJIA index, our study confirms that there is evidence of the correlation between macroeconomic variables and the stock market returns, which numerous previous research have hypothesized [RS07][Nai13][WS02][PK00].

There are certainly areas of opportunity in this research. One example of further analysis would be to attempt different types of models in fitting the data, since the ARMA models appear to have increased errors during a black swan event such as the COVID-19 crash in March 2020. Some alternative models that might be worth trying are the autoregressive conditionally heteroscedastic (ARCH) model and the generalized ARCH (GARCH) model, which are methodology also commonly used to fit financial data by incorporating changes in volatility [SS17]. This may specifically help improve the prediction accuracy during times of uncertainty. On the other hand, since the study is unable to find statistically significant cycles present in the data, another area for further research is expanding the length of time represented in the training data to dive deeper into longer term cycles, or analyzing aggregated data (e.g. weekly, monthly) to have cyclical patterns show up more distinctly through filtering of noise.

APPENDIX

```
library(ggplot2);library(forecast);library(astsa)
dt <- read.csv("DOW.csv", stringsAsFactors = FALSE)
dt$Date <- as.character(as.Date(dt$Date, format = "%m/%d/%y"))
dt <- dt[order(dt$Date),]; rownames(dt) <- 1:nrow(dt)
tsy10y <- read.csv("^TNX.csv", stringsAsFactors = FALSE)
tsy10y$Date <- as.character(as.Date(tsy10y$Date, format = "%Y-%m-%d"))
inflation <- read.csv("inflation.csv", stringsAsFactors = FALSE)
names(inflation) <- c("Date","infn")
inflation$infn <- as.numeric(inflation$infn)
inflation$Date <- as.character(as.Date(inflation$Date, format = "%Y-%m-%d"))
for (d in (dt$Date[which(!dt$Date %in% tsy10y$Date)])){tsy10y <- rbind(tsy10y,
    cbind(Date=d,tsy10y[tsy10y$Date==max(tsy10y$Date[tsy10y$Date < d]),!names(
    tsy10y)%in%("Date")]))}
tsy10y <- tsy10y[order(tsy10y$Date),]; rownames(tsy10y) <- 1:nrow(tsy10y)
for (d in (dt$Date[which(!dt$Date %in% inflation$Date)])){inflation <- rbind(
    inflation,cbind(Date=d,infn=as.numeric(inflation[inflation$Date==max(
    inflation$Date[inflation$Date < d]),!names(inflation)%in%("Date")])))}
inflation$infn <- as.numeric(inflation$infn)
inflation <- inflation[order(inflation$Date),]; rownames(inflation) <- 1:nrow(
    inflation)
dt$return <- sapply(1:length(dt$Close), function(i){ifelse(i==1, NA, round((dt
    $Close[i]-dt$Close[i-1])/dt$Close[i-1],5))})
dt <- dt[255:nrow(dt),]; rownames(dt) <- 1:nrow(dt)
tsy10y$perc_chg <- sapply(1:length(tsy10y$Close), function(i){ifelse(i==1, NA,
     round((tsy10y$Close[i]-tsy10y$Close[i-1])/tsy10y$Close[i-1],5))})
tsy10y <- tsy10y[2:nrow(tsy10y),]; rownames(tsy10y) <- 1:nrow(tsy10y)
inflation$perc_chg <- sapply(1:length(inflation$infn), function(i){ifelse(i
    ==1, NA, round((inflation$infn[i]-inflation$infn[i-1])/inflation$infn[i
    -1],5))})
inflation <- inflation[2:nrow(inflation),]; rownames(inflation) <- 1:nrow(
    inflation)
dt$t10y <- sapply(1:length(dt$Close), function(i){1/100.0*tsy10y$Close[tsy10y$
```

```
    Date==dt$Date[i]]})
dt$t10y_perc_chg <- sapply(1:length(dt$Close), function(i){tsy10y$perc_chg[
    tsy10y$Date==dt$Date[i]]})
dt$infn <- sapply(1:length(dt$Close), function(i){1/100.0*inflation$infn[
    inflation$Date == dt$Date[i]]})
dt$infn_perc_chg <- sapply(1:length(dt$Close), function(i){inflation$perc_chg[
    inflation$Date==dt$Date[i]]})
dt$idx <- order(dt$Date)
###### split into train and test
train <- dt[1:round(nrow(dt)*0.6,0),]; test <- dt[(nrow(train)+1):nrow(dt),]
n_dt <- nrow(dt);n_train <- nrow(train);n_test <- nrow(test)
###### models
m1 <- Arima(train$return, order = c(0,0,0));checkresiduals(m1)
### predictions at lag 1, with data continuing being updated
rolling_prediction <- function(object, newdata, olddata){
  res <- as.numeric(object$residuals)
  pq <- object$arma
  p <- pq[1]
  q <- pq[2]
  ar <- object$coef[seq(length.out=p)]
  ma <- object$coef[p+seq(length.out=q)]
  int <- object$coef["intercept"]
  newdata <- as.numeric(newdata)
  olddata <- as.numeric(olddata)
  n_new <- length(newdata)
  n_old <- length(olddata)
  dat <- c(olddata, newdata)
  yhat <- c()
  if (p+q==0) return(rep(0, nnew))
  for (i in 1:n_new){
    ind <- n_old + i
    x_ar <- dat[ind - seq(length.out=p)]
    x_ma <- res[ind - seq(length.out=q)]
    yhat_ar <- sum(x_ar*ar)
    yhat_ma <- sum(x_ma*ma)
```

```
    if (length(yhat_ar)==0) yhat_ar <- 0
    if (length(yhat_ma)==0) yhat_ma <- 0
    yhat[i] <- yhat_ar + yhat_ma# + int
    res[ind] <- dat[ind] - yhat[i]
  }
  return (list(predictions=yhat,
               res=res[n_old + 1:n_new]))
}
# predictions 1
n_pred <- n_test
return.preds1 <- predict(m1, n.ahead = n_pred)$pred
predictions1 <- c(); returns1 <- c()
for (i in c(1:n_pred)){
  returns1 <- c(returns1, return.preds1[i])
  predictions1 <- c(predictions1, dt$Close[n_train+i-1]*(1+return.preds1[i]))
}
close.error1 <- mean(abs(predictions1-dt$Close[(n_train+1):(n_train+n_pred)]))
###### spectral analysis
spec <- spectrum(na.omit(ma(train$return, order = 3)), main = "Spectral
    Density Estimation by Periodogram", method = "pgram", las=1,ci.col="
    dodgerblue2")
lines(spec$freq, rep(mean(spec$spec),length(spec$freq)), col = "red")
1/spec$freq[order(spec$spec, decreasing = T)][1:20]
###### remove cycles from spectrum analysis
train$c1 <- NULL; train$c2 <- NULL; train$c3 <- NULL; train$c4 <- NULL; train$
    c5 <- NULL
train$mean1 <- NULL; train$mean2 <- NULL; train$mean3 <- NULL; train$mean4 <-
    NULL; train$mean5 <- NULL; train$res <- NULL
train$c1 <- as.numeric(rownames(train))%%25; train$c1[which(dt$train==0)] <-
    25
train$c2 <- as.numeric(rownames(train))%%51; train$c2[which(train$c2==0)] <-
    51
c1.mean <- data.frame(c1=names(tapply(train$return, train$c1, mean)), mean1=
    tapply(train$return, train$c1, mean))
cycle1 <- merge(train, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
```

```
c2.mean <- data.frame(c2=names(tapply(train$return, train$c2, mean)), mean2=
    tapply(train$return, train$c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
cycle4 <- cycle2[order(cycle2$Date),]
cycle4$res <- cycle4$return-cycle4$mean1-cycle4$mean2
train$mean1 <- sapply(1:nrow(train), function(i){cycle4$mean1[cycle4$Date ==
    train$Date[i]]})
train$mean2 <- sapply(1:nrow(train), function(i){cycle4$mean2[cycle4$Date ==
    train$Date[i]]})
train$res <- sapply(1:nrow(train), function(i){cycle4$res[cycle4$Date == dt$
    Date[i]]})
###### model on residuals after removing cycles
m2 <- Arima(train$res, order = c(2,0,3)); checkresiduals(m2)
AIC(m1); AIC(m2)
# predictions model 2
dt$c1 <- NULL; dt$c2 <- NULL; dt$c3 <- NULL; dt$c4 <- NULL; dt$c5 <- NULL
dt$mean1 <- NULL; dt$mean2 <- NULL; dt$mean3 <- NULL; dt$mean4 <- NULL; dt$
    mean5 <- NULL; dt$res <- NULL
dt$c1 <- as.numeric(rownames(dt))%%25; dt$c1[which(dt$c1==0)] <- 25
dt$c2 <- as.numeric(rownames(dt))%%51; dt$c2[which(dt$c2==0)] <- 51
c1.mean <- data.frame(c1=names(tapply(dt$return, dt$c1, mean)), mean1=tapply(
    dt$return, dt$c1, mean))
cycle1 <- merge(dt, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
c2.mean <- data.frame(c2=names(tapply(dt$return, dt$c2, mean)), mean2=tapply(
    dt$return, dt$c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
cycle5 <- cycle2[order(cycle2$Date),]
cycle5$res <- cycle5$return-cycle5$mean1-cycle5$mean2
dt$mean1 <- sapply(1:nrow(dt), function(i){cycle5$mean1[cycle5$Date == dt$Date
    [i]]})
dt$mean2 <- sapply(1:nrow(dt), function(i){cycle5$mean2[cycle5$Date == dt$Date
    [i]]})
dt$res <- sapply(1:nrow(dt), function(i){cycle5$res[cycle5$Date == dt$Date[i
    ]]})
test <- dt[(nrow(train)+1):nrow(dt),]
```

```
res.preds2 <- predict(m2, n.ahead = n_pred)$pred
res.preds2 <- rolling_prediction(m2, test$res[1:n_pred], train$res)$
   predictions
predictions2 <- c(); returns2 <- c()
for (i in c(1:n_pred)){
  returns2 <- c(returns2, res.preds2[i]+test$mean1[i]+test$mean2[i])
  predictions2 <- c(predictions2, dt$Close[n_train+i-1]*(1+res.preds2[i]+test$
     mean1[i]+test$mean2[i]))
}
close.error2 <- mean(abs(predictions2-dt$Close[(n_train+1):(n_train+n_pred)]))
###### regression with treasury 10 year yield
fit <- lm(data=train, return ~ t10y_perc_chg); summary(fit)
dt$return_t10y_perc_chg_res <- dt$return-predict(fit, newdata=dt)
train$return_t10y_perc_chg_res <- train$return-predict(fit, newdata=train)
test$return_t10y_perc_chg_res <- test$return-predict(fit, newdata=test)
m3 <- Arima(train$return_t10y_perc_chg_res, order = c(3,0,0)); checkresiduals(
   m3)
AIC(m1); AIC(m2); AIC(m3)
# predictions
return_t10y_perc_chg_res.preds3 <- predict(m3, n.ahead = n_pred)$pred
return_t10y_perc_chg_res.preds3 <- rolling_prediction(m3, test$return_t10y_
   perc_chg_res[1:n_pred], train$return_t10y_perc_chg_res)$predictions
predictions3 <- c(); returns3 <- c()
for (i in c(1:n_pred)){
  returns3 <- c(returns3, return_t10y_perc_chg_res.preds3[i]+predict(fit,
     newdata=test)[i])
  predictions3 <- c(predictions3, dt$Close[n_train+i-1]*(1+return_t10y_perc_
     chg_res.preds3[i]+predict(fit, newdata=test)[i]))
}
close.error3 <- mean(abs(predictions3-dt$Close[(n_train+1):(n_train+n_pred)]))
###### remove cycles from spectrum analysis on return less t10y perc change
train$c1 <- NULL; train$c2 <- NULL; train$c3 <- NULL; train$c4 <- NULL; train$
   c5 <- NULL
train$mean1 <- NULL; train$mean2 <- NULL; train$mean3 <- NULL; train$mean4 <-
   NULL; train$mean5 <- NULL; train$res <- NULL
```

```
train$c1 <- as.numeric(rownames(train))%%19; train$c1[which(train$c1==0)] <-
   19
train$c2 <- as.numeric(rownames(train))%%10; train$c2[which(train$c2==0)] <-
   10
train$c3 <- as.numeric(rownames(train))%%25; train$c3[which(train$c3==0)] <-
   25
c1.mean <- data.frame(c1=names(tapply(train$return_t10y_perc_chg_res, train$c1
   , mean)), mean1=tapply(train$return_t10y_perc_chg_res, train$c1, mean))
cycle1 <- merge(train, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
c2.mean <- data.frame(c2=names(tapply(train$return_t10y_perc_chg_res, train$c2
   , mean)), mean2=tapply(train$return_t10y_perc_chg_res, train$c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
c3.mean <- data.frame(c3=names(tapply(train$return_t10y_perc_chg_res, train$c3
   , mean)), mean3=tapply(train$return_t10y_perc_chg_res, train$c3, mean))
cycle3 <- merge(cycle2, c3.mean, by = "c3", sort = FALSE, all.x = TRUE)
cycle5 <- cycle3[order(cycle3$Date),]
cycle5$res <- cycle5$return_t10y_perc_chg_res-cycle5$mean1-cycle5$mean2-cycle5
   $mean3
train$mean1 <- sapply(1:nrow(train), function(i){cycle5$mean1[cycle5$Date ==
   train$Date[i]]})
train$mean2 <- sapply(1:nrow(train), function(i){cycle5$mean2[cycle5$Date ==
   train$Date[i]]})
train$mean3 <- sapply(1:nrow(train), function(i){cycle5$mean3[cycle5$Date ==
   train$Date[i]]})
train$res <- sapply(1:nrow(train), function(i){cycle5$res[cycle5$Date == train
   $Date[i]]})
###### model 4
m4 <- Arima(train$res, order = c(1,0,2)); checkresiduals(m4)
AIC(m1); AIC(m2); AIC(m3); AIC(m4)
# predictions model 4
dt$c1 <- NULL; dt$c2 <- NULL; dt$c3 <- NULL; dt$c4 <- NULL; dt$c5 <- NULL
dt$mean1 <- NULL; dt$mean2 <- NULL; dt$mean3 <- NULL; dt$mean4 <- NULL; dt$
   mean5 <- NULL; dt$res <- NULL
dt$c1 <- as.numeric(rownames(dt))%%19; dt$c1[which(dt$c1==0)] <- 19
dt$c2 <- as.numeric(rownames(dt))%%10; dt$c2[which(dt$c2==0)] <- 10
```

```
dt$c3 <- as.numeric(rownames(dt))%%25; dt$c3[which(dt$c3==0)] <- 25
c1.mean <- data.frame(c1=names(tapply(dt$return_t10y_perc_chg_res, dt$c1, mean
    )), mean1=tapply(dt$return_t10y_perc_chg_res, dt$c1, mean))
cycle1 <- merge(dt, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
c2.mean <- data.frame(c2=names(tapply(dt$return_t10y_perc_chg_res, dt$c2, mean
    )), mean2=tapply(dt$return_t10y_perc_chg_res, dt$c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
c3.mean <- data.frame(c3=names(tapply(dt$return_t10y_perc_chg_res, dt$c3, mean
    )), mean3=tapply(dt$return_t10y_perc_chg_res, dt$c3, mean))
cycle3 <- merge(cycle2, c3.mean, by = "c3", sort = FALSE, all.x = TRUE)
cycle5 <- cycle3[order(cycle3$Date),]
cycle5$res <- cycle5$return_t10y_perc_chg_res-cycle5$mean1-cycle5$mean2-cycle5
    $mean3
dt$mean1 <- sapply(1:nrow(dt), function(i){cycle5$mean1[cycle5$Date == dt$Date
    [i]]})
dt$mean2 <- sapply(1:nrow(dt), function(i){cycle5$mean2[cycle5$Date == dt$Date
    [i]]})
dt$mean3 <- sapply(1:nrow(dt), function(i){cycle5$mean3[cycle5$Date == dt$Date
    [i]]})
dt$res <- sapply(1:nrow(dt), function(i){cycle5$res[cycle5$Date == dt$Date[i
    ]]})
test <- dt[(nrow(train)+1):nrow(dt),]
res.preds4 <- predict(m4, n.ahead = n_pred)$pred
predictions4 <- c(); returns4 <- c()
for (i in c(1:n_pred)){
  returns4 <- c(returns4, (res.preds4[i]+test$mean1[i]+test$mean2[i]+test$
      mean3[i])+predict(fit, newdata=test)[i])
  predictions4 <- c(predictions4, dt$Close[n_train+i-1]*(1+(res.preds4[i]+test
      $mean1[i]+test$mean2[i]+test$mean3[i])+predict(fit, newdata=test)[i]))
}
close.error4 <- mean(abs(predictions4-dt$Close[(n_train+1):(n_train+n_pred)]))
###### checking correlation with treasury 10 year yield and inflation perc chg
fit2 <- lm(data=train, return ~ t10y_perc_chg + infn_perc_chg); summary(fit2)
dt$return_t10y_infn_perc_chg_res <- dt$return-predict(fit2, newdata=dt)
train$return_t10y_infn_perc_chg_res <- train$return-predict(fit2, newdata=
```

```
    train)
test$return_t10y_infn_perc_chg_res <- test$return-predict(fit2, newdata=test)
m5 <- Arima(train$return_t10y_perc_chg_res, order = c(0,0,0)); checkresiduals(
    m5)
AIC(m1); AIC(m2); AIC(m3); AIC(m4); AIC(m5)
# predictions
return_t10y_infn_perc_chg_res.preds5 <- predict(m5, n.ahead = n_pred)$pred
predictions5 <- c(); returns5 <- c()
for (i in c(1:n_pred)){
  returns5 <- c(returns5, return_t10y_infn_perc_chg_res.preds5[i]+predict(fit2
      , newdata=test)[i])
  predictions5 <- c(predictions5, dt$Close[n_train+i-1]*(1+return_t10y_infn_
      perc_chg_res.preds5[i]+predict(fit2, newdata=test)[i]))
}
close.error5 <- mean(abs(predictions5-dt$Close[(n_train+1):(n_train+n_pred)]))
###### remove cycles from spectrum analysis on return less t10y perc change
    and infn perc chg
train$c1 <- NULL; train$c2 <- NULL; train$c3 <- NULL; train$c4 <- NULL; train$
    c5 <- NULL
train$mean1 <- NULL; train$mean2 <- NULL; train$mean3 <- NULL; train$mean4 <-
    NULL; train$mean5 <- NULL; train$res <- NULL
train$c1 <- as.numeric(rownames(train))%%17; train$c1[which(train$c1==0)] <-
    17
train$c2 <- as.numeric(rownames(train))%%8; train$c2[which(train$c2==0)] <- 8
train$c3 <- as.numeric(rownames(train))%%25; train$c3[which(train$c3==0)] <-
    25
c1.mean <- data.frame(c1=names(tapply(train$return_t10y_infn_perc_chg_res,
    train$c1, mean)), mean1=tapply(train$return_t10y_infn_perc_chg_res, train$
    c1, mean))
cycle1 <- merge(train, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
c2.mean <- data.frame(c2=names(tapply(train$return_t10y_infn_perc_chg_res,
    train$c2, mean)), mean2=tapply(train$return_t10y_infn_perc_chg_res, train$
    c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
c3.mean <- data.frame(c3=names(tapply(train$return_t10y_infn_perc_chg_res,
```

```
    train$c3, mean)), mean3=tapply(train$return_t10y_infn_perc_chg_res, train$
    c3, mean))
cycle3 <- merge(cycle2, c3.mean, by = "c3", sort = FALSE, all.x = TRUE)
cycle5 <- cycle3[order(cycle3$Date),]
cycle5$res <- cycle5$return_t10y_infn_perc_chg_res-cycle5$mean1-cycle5$mean2-
    cycle5$mean3
train$mean1 <- sapply(1:nrow(train), function(i){cycle5$mean1[cycle5$Date ==
    train$Date[i]]})
train$mean2 <- sapply(1:nrow(train), function(i){cycle5$mean2[cycle5$Date ==
    train$Date[i]]})
train$mean3 <- sapply(1:nrow(train), function(i){cycle5$mean3[cycle5$Date ==
    train$Date[i]]})
train$res <- sapply(1:nrow(train), function(i){cycle5$res[cycle5$Date == train
    $Date[i]]})
###### model 6
m6 <- Arima(train$res, order = c(0,0,0)); checkresiduals(m6)
AIC(m1); AIC(m2); AIC(m3); AIC(m4); AIC(m5); AIC(m6)
# predictions model 6
dt$c1 <- NULL; dt$c2 <- NULL; dt$c3 <- NULL; dt$c4 <- NULL; dt$c5 <- NULL
dt$mean1 <- NULL; dt$mean2 <- NULL; dt$mean3 <- NULL; dt$mean4 <- NULL; dt$
    mean5 <- NULL; dt$res <- NULL
dt$c1 <- as.numeric(rownames(dt))%%17; dt$c1[which(dt$c1==0)] <- 17
dt$c2 <- as.numeric(rownames(dt))%%8; dt$c2[which(dt$c2==0)] <- 8
dt$c3 <- as.numeric(rownames(dt))%%25; dt$c3[which(dt$c3==0)] <- 25
c1.mean <- data.frame(c1=names(tapply(dt$return_t10y_infn_perc_chg_res, dt$c1,
     mean)), mean1=tapply(dt$return_t10y_infn_perc_chg_res, dt$c1, mean))
cycle1 <- merge(dt, c1.mean, by = "c1", sort = FALSE, all.x = TRUE)
c2.mean <- data.frame(c2=names(tapply(dt$return_t10y_infn_perc_chg_res, dt$c2,
     mean)), mean2=tapply(dt$return_t10y_infn_perc_chg_res, dt$c2, mean))
cycle2 <- merge(cycle1, c2.mean, by = "c2", sort = FALSE, all.x = TRUE)
c3.mean <- data.frame(c3=names(tapply(dt$return_t10y_infn_perc_chg_res, dt$c3,
     mean)), mean3=tapply(dt$return_t10y_infn_perc_chg_res, dt$c3, mean))
cycle3 <- merge(cycle2, c3.mean, by = "c3", sort = FALSE, all.x = TRUE)
cycle5 <- cycle3[order(cycle3$Date),]
cycle5$res <- cycle5$return_t10y_infn_perc_chg_res-cycle5$mean1-cycle5$mean2-
```

```
    cycle5$mean3
dt$mean1 <- sapply(1:nrow(dt), function(i){cycle5$mean1[cycle5$Date == dt$Date
    [i]]})
dt$mean2 <- sapply(1:nrow(dt), function(i){cycle5$mean2[cycle5$Date == dt$Date
    [i]]})
dt$mean3 <- sapply(1:nrow(dt), function(i){cycle5$mean3[cycle5$Date == dt$Date
    [i]]})
dt$res <- sapply(1:nrow(dt), function(i){cycle5$res[cycle5$Date == dt$Date[i
    ]]})
test <- dt[(nrow(train)+1):nrow(dt),]
res.preds6 <- predict(m6, n.ahead = n_pred)$pred
predictions6 <- c(); returns6 <- c()
for (i in c(1:n_pred)){
  returns6 <- c(returns6, (res.preds6[i]+test$mean1[i]+test$mean2[i]+test$
      mean3[i])+predict(fit2, newdata=test)[i])
  predictions6 <- c(predictions6, dt$Close[n_train+i-1]*(1+(res.preds6[i]+test
      $mean1[i]+test$mean2[i]+test$mean3[i])+predict(fit2, newdata=test)[i]))
}
close.error6 <- mean(abs(predictions6-dt$Close[(n_train+1):(n_train+n_pred)]))
#################################### comparison graphs
m1.res <- checkresiduals(m1);m2.res <- checkresiduals(m2);m3.res <-
    checkresiduals(m3);m4.res <- checkresiduals(m4);m5.res <- checkresiduals(
    m5);m6.res <- checkresiduals(m6)
AIC(m1); AIC(m2); AIC(m3); AIC(m4); AIC(m5); AIC(m6)
m1.res$p.value;m2.res$p.value;m3.res$p.value;m4.res$p.value;m5.res$p.value;m6.
    res$p.value
close.error1; close.error2; close.error3; close.error4; close.error5; close.
    error6
sqrt(mean(test$Close-dt$Close[n_train:(n_train+n_test-1)])^2) # 0 percent
    return assumption
```

REFERENCES

[GGM64] Michael D Godfrey, Clive WJ Granger, and Oskar Morgenstern. "THE RANDOM-WALK HYPOTHESIS OF STOCK MARKET BEHAVIOR." https://doi.org/10.1111/j.1467-6435.1964.tb02458.x, 1964.

[GM63] Clive WJ Granger and Oskar Morgenstern. "Spectral analysis of New York stock market prices 1." https://doi.org/10.1111/j.1467-6435.1963.tb00270.x, 1963.

[Nai13] Pramod Kumar Naik. "Does stock market respond to economic fundamentals? Time-series analysis from Indian data." *Journal of Applied Economics and Business Research*, **3**(1):34–50, 2013.

[PK00] Abhay Pethe and Ajit Karnik. "Do Indian stock markets matter? Stock market indices and macro-economic variables." *Economic and political weekly*, pp. 349–356, 2000. https://www.jstor.org/stable/4408881.

[RS07] Orawan Ratanapakorn and Subhash C Sharma. "Dynamic analysis between the US stock returns and the macroeconomic variables." *Applied Financial Economics*, **17**(5):369–377, 2007. https://doi.org/10.1080/09603100600638944.

[SS17] Robert H Shumway and David S Stoffer. *Time series analysis and its applications*, volume 4, pp. 21,75–84,98–99,253–255. Springer, 2017.

[Tal07] Nassim Nicholas Taleb. *The black swan: The impact of the highly improbable*, volume 2. Random house, 2007.

[TTK21] Manish Talwar, Shalini Talwar, Puneet Kaur, Naliniprava Tripathy, and Amandeep Dhir. "Has financial attitude impacted the trading activity of retail investors during the COVID-19 pandemic?" *Journal of Retailing and Consumer Services*, **58**:102341, 2021.

[WS02] Praphan Wongbangpo and Subhash C Sharma. "Stock market and macroeconomic fundamental dynamic interactions: ASEAN-5 countries." *Journal of asian Economics*, **13**(1):27–51, 2002.