

UNIVERSITY OF CALIFORNIA

Los Angeles

Forecasting the Time Series of Apple Inc.'s Stock Price

A thesis submitted in partial satisfaction  
of the requirements for the degree Master of Applied Statistics

by

Jordan Berninger

2018

©Copyright by

Jordan Berninger

2018

## ABSTRACT OF THE THESIS

Forecasting the Time Series of Apple Inc.'s Stock Price

by

Jordan Berninger

Master of Applied Statistics

University of California, Los Angeles, 2018

Professor Juana Sanchez, Co-Chair

Professor Qing Zhou, Co-Chair

The 21st Century has been defined by the exponential growth of the information technology (IT) industry, the smart phone and the personal computer. Apple Inc. has played a major role in that growth, with Apple products widely considered emblematic of the IT revolution. The health of Apple Inc. is a predictor of the health of the industry and industries that depend on it. Apple's stock price is an indicator of Apple's health. Therefore, it is of particular interest to have good models to predict the stock price of such a hallmark company of this IT revolution. Statistical time series analysis is of paramount importance to do that. This thesis compares the forecasting performance of

univariate and multivariate time series models of Apple stock’s opening price for the first day of each month. Among well-known univariate models, we consider Autoregressive Integrated Moving Average (ARIMA) models, ARIMA with Generalized Autoregressive Conditional Heteroscedascity (GARCH) and Exponential Smoothing, all models where the predicted opening price depends on past values of itself and nothing else. Among multivariate methods known in time series analysis, we consider Vector Autoregression (VAR), which endogenizes all the variables considered, and classical linear regression with ARIMA residuals. We fit these models to an “in-sample“ (training set) of historical opening price from January 1990 to September 2016, and use them to forecast 12 months “out of sample“ (test set) October 2016 to September 2017. In the multivariate models, the predictors of Apple’s opening price are: the stock price of the *S&P500*, Microsoft and Texas Instruments. We hypothesize a positive correlation of Apple’s stock with the *S&P500* stock, a negative correlation with its competitor Microsoft and a positive correlation with Apple’s supplier, Texas Instruments. We compare the forecasts from each model with the actual values of the “out of sample“ time series (the test set) to assess forecasting performance according to the Root Mean Square Error (RMSE). We find that an average forecast consisting of the average of all those models’ forecasts, what is known in the forecasting industry as “the consensus forecast,” has the lowest RMSE for the 12-month test set. This is an important result, not only because it explains the decades’ old practice of considering averages of forecasts from many models as “ the forecast“ of any relevant economic variable, but also because it highlights the benefits of integrating

multiple classical models to obtain a good predictive performance.

The thesis of Jordan Berninger is approved.

Ying Nian Wu

Frederic Paik Schoenberg

Qing Zhou, Committee Co-Chair

Juana Sanchez, Committee Co-Chair

University of California, Los Angeles

2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Recent Research . . . . .	5
1.2	Hypotheses . . . . .	7
<b>2</b>	<b>The Data</b>	<b>8</b>
<b>3</b>	<b>Descriptive Time Series Analysis of the Data</b>	<b>10</b>
3.1	Time Series Decomposition of AAPL . . . . .	13
3.2	The Sample Autocorrelation Function . . . . .	18
<b>4</b>	<b>ARIMA Model</b>	<b>20</b>
4.1	Identification of $ARIMA(p, d, q)$ . . . . .	22
4.2	Fitting our ARIMA Model . . . . .	25
4.3	ARIMA Model Diagnostics . . . . .	26
4.4	Forecasting with ARIMA Model . . . . .	27
4.5	ARIMA + GARCH Model . . . . .	28
4.6	Forecasting AAPL with ARIMA + GARCH Model . . . . .	31
<b>5</b>	<b>Exponential Smoothing</b>	<b>32</b>
5.1	Forecasting our Holt-Winters Model . . . . .	34
<b>6</b>	<b>Vector Autoregression</b>	<b>35</b>
6.1	Unit roots and co-integration tests . . . . .	36
6.2	Differencing Covariates . . . . .	37
6.3	Cross-correlations . . . . .	38

6.4	Fitting the VAR model . . . . .	38
6.5	Impulse Response Functions . . . . .	42
6.6	Forecasting with our VAR model . . . . .	45
<b>7</b>	<b>Time Series Regression</b>	<b>46</b>
7.1	Linear Regression Model - GLS and CORARMA . . . . .	49
7.2	Linear Regression Model on the Differenced Series . . . . .	49
7.3	Linear Regression Model - Forecasting with the Best Regression Model . . . . .	51
<b>8</b>	<b>Conclusions</b>	<b>53</b>
8.1	Best Short-Term Forecast . . . . .	54
8.2	Best Long-Term Forecast . . . . .	55
8.3	Final Conclusions . . . . .	57
8.4	Further Research . . . . .	58
<b>A</b>	<b>Appendix I: Figures and Tables</b>	<b>60</b>
<b>B</b>	<b>Appendix II: R Code</b>	<b>62</b>



## List of Figures

1	AAPL Opening Price . . . . .	11
2	AAPL Montly Opening Price 1990 - 2004 . . . . .	12
3	Opening Prices of the Covariates . . . . .	13
4	AAPL Multiplicative Decomposition Components . . . . .	16
5	AAPL Seasonal Boxplot . . . . .	17
6	ACF of Random component of AAPL obtained by multiplica- tive decomposition . . . . .	19
7	ACF $(1 - B)Y_t$ . . . . .	23
8	ACF of Residuals of ARIMA (0,1,0) on $Y_t$ . . . . .	26
9	ARIMA Forecast . . . . .	28
10	ACF of Squared Residuals $ARIMA(0, 1, 0)$ . . . . .	29
11	ARIMA(0,1,0) GARCH(1,1) . . . . .	31
12	ARIMA(0,1,0) GARCH(1,1) Forecast . . . . .	32
13	Holt-Winters Forecast . . . . .	34
14	Cross-correlograms . . . . .	39
15	ACF of Residuals . . . . .	40
16	Cross-correlogram Residuals . . . . .	41
17	IRF - Shock to AAPL . . . . .	43
18	IRF - Shock to MSFT . . . . .	44
19	IRF - Shock to GSPC . . . . .	45
20	IRF - Shock to TXN . . . . .	46
21	VAR(1) Forecast . . . . .	47
22	ACF of Linear Regression Model Residuals . . . . .	50

23	Residuals ACF - Regression on Differenced Series . . . . .	51
24	Linear Regression Forecast . . . . .	52
25	Model's Average Forecast . . . . .	56
26	All Models Forecast (Last 50 Observations) ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green .	57
27	All Models Forecast ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smooth- ing = yellow, Averages = green . . . . .	60
28	All Models Forecast (Last 100 Observations) ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green .	61
29	All Models Forecast (Last 50 Observations) ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green .	61

## List of Tables

1	Training and Test Periods. . . . .	10
2	Short Term (4 month) Forecast . . . . .	54
3	Long Term (12 month) Forecast . . . . .	55
4	Correlation Matrix of our 4 Time Series . . . . .	60

# 1 Introduction

Time series analysis differs significantly from classical statistics analysis. Time series models take into consideration that chronologically ordered observations of a random variable are correlated. This consideration is important because Donald Cochran and Guy Orcutt showed in 1949 that if residuals of an estimated regression equation are autocorrelated, the regression parameters' variances are underestimated, which leads to an overestimation of the  $F$  and  $t$  statistics<sup>1</sup>. James Durbin and Geoffrey Watson created a methodology in 1950 to detect first order auto-correlation<sup>2</sup>. Time series analysis models the autocorrelation in the data and, therefore, is a more comprehensive way of accounting for the autocorrelation, if it is an intrinsic feature of the data<sup>3</sup>.

Early time series analyses decomposed observed data into a small number of underlying, independent components. This old approach is very helpful in getting a good initial impression of what the signal is and what the noise is in the data. In 1919, Warren Persons formalized four distinct components - first, the trend (long term deterministic tendency), second, a cyclical deterministic component with period greater than one year, third, the seasonal deterministic component (cycle with period of one year), and fourth, a random residual component that catches all the fluctuations not explained by the first

---

<sup>1</sup>Kirchgassner, 1.

<sup>2</sup>Kirchgassner, 1.

<sup>3</sup>Chan, 2.

three systematic components<sup>4</sup>. The fourth residual component is the random signal to model. This classical concept of time series assumes that the first three systematic components are deterministic, and only the fourth, random component is subject to stochastic influence<sup>5</sup>. This classical approach adopts a descriptive approach to time series analysis; in Section 3, we take such an approach since it is helpful to determine the components of Apple stock at a descriptive level. In the last 50 years, a modern approach has developed which assumes all components of a time series could be stochastic<sup>6</sup>.

In 1970, Box and Jenkins introduced the Autoregressive Integrated Moving Average (ARIMA) methodology, which became one of the most popular methods for forecasting a time series<sup>7</sup>. Univariate ARIMA models are considered in Section 4 of this thesis. Univariate models predict future values using a linear combination of past values of itself and past errors. ARIMA models are very effective, but they are unable to model non-constant variance with respect to time, or volatility. A common remedy is to fit an autoregressive moving average (ARMA) model on the variance process; we call these models Autoregressive Conditional Heteroscedastic (ARCH) models<sup>8</sup>. In 1986, Tim Bollerslev and Stephen Taylor independently developed methods to generalize the ARCH model, which has been called the Generalized Autoregressive Con-

---

<sup>4</sup>Kirchgassner, 3.

<sup>5</sup>Kirchgassner, 3.

<sup>6</sup>Kirchgassner, 4.

<sup>7</sup>Box and Jenkins.

<sup>8</sup>Kirchgassner, 285-286.

ditional Heteroscedasticity (GARCH) model<sup>9</sup>. We refine our ARIMA model with the GARCH approach in the second half of Section 4.

In Section 5, we fit another univariate model, the Holt-Winters model, also known as an Exponential Smoothing model. The Holt-Winters model is similar to a moving average model, the difference being that exponentially decreasing weights are assigned to the values observed further in the past<sup>10</sup>.

We want to investigate whether using covariates results in a better forecast than models without covariates. For this reason, in Section 6, we implement a Vector Autoregressive Systems (VAR) model, which was developed by Christopher Sims in 1980, and for which he earned the 2011 Nobel Prize in Economics Sciences<sup>11</sup>. The VAR model is a multivariate model, which means it uses contemporaneous time series. VAR models make the assumption that all the time series are endogenous, that is, all past values of themselves and the other variables affects the time series in the system. Our VAR model will use the *S&P500* market index and the stock prices of Microsoft, a competitor of Apple, and Texas Instruments, a supplier of parts for Apple. One major advantage of the *VAR()* model in the R<sup>12</sup> package *vars* is that it allows us to easily plot impulse response functions, through which we can evaluate how a shock to one time series affects the others in the system across time and how long it takes the variables to return to equilibrium.

---

<sup>9</sup>Kirchgassner, 292.

<sup>10</sup>Cowpertwait and Metcalfe, 55.

<sup>11</sup>Kirchgassner, 128, 151.

<sup>12</sup>R: A language and environment for statistical computing.

In Section 7, we implement a multiple Linear Regression model, with corrections for autocorrelated errors. In classic regression, using ordinary least squares, the assumption is that the error terms are uncorrelated. In time series regression, this is often not the case, accordingly, we fit a stationary covariance structure for the residual series<sup>13</sup>. In Section 7, we adopt this approach and fit an ARMA structure to the residuals from a linear regression model. Again, we use the time series of the *S&P500*, Microsoft and Texas Instruments for this model, but, in this model, they are assumed to be independent variables, with Apple’s stock price being the only dependent variable. Personal choice played a major factor in this choice of three covariates. We expand on the reasons for choosing them, and the hypotheses about their relationships in Section 1.2.

Stock market data is significantly different from other economic time series. This introduction proceeds with an overview of recent research into time series forecasting of stock market data, and our hypotheses about the covariates. Then, we systematically identify, introduce, fit, diagnose, forecast and visualize the aforementioned models in the sections mentioned above.

Throughout this thesis, an overarching concept for model evaluation is the parsimony principle. This means, for each type of model, we choose the simplest model, with the fewest parameters, that adequately fits the time series, and produces a residual error series with no significant autocorrelations. This concept is similar to Occam’s razor, improves interpretability and prevents over-fitting. The Conclusions Section will state the best model for the short-

---

<sup>13</sup>Shumway and Stoffer, 145.

term (4 month) forecasting, long-term forecasting (12 month) and lists possible avenues for further research.

## 1.1 Recent Research

In the last several decades, Neural Networks and Deep Learning models have been extensively applied to forecasting stock prices. Kamijo and Tanigawa developed a method using pattern recognition to predict the stock prices of Japanese companies<sup>14</sup>. Yoon and Swalves developed a methodology using a neural-network with four layers to predict stock prices of American companies<sup>15</sup>. Baba and Kozaki also developed a neural network methodology, using back propagation and random optimization to predict Japanese stock prices<sup>16</sup>. Saad, Prokhorov, and Wunsch investigated time-delay, probabilistic, and recurrent neural networks to predict stock prices<sup>17</sup>. Hadavandi, Shavandi, and Ghanbari developed an approach that integrated genetic fuzzy systems and artificial neural networks to forecast stock prices and concluded that their method produced good forecasts<sup>18</sup>.

Many have also researched hybrid approaches that integrate ARIMA or ARMA models with machine learning algorithms to forecast stock prices. Pai and Lin developed a method to integrate support vector machines with

---

<sup>14</sup>Kamijo and Tanigawa.

<sup>15</sup>Yoon and Swalves.

<sup>16</sup>Baba and Kozaki.

<sup>17</sup>Saad, Prokhorov and Wunsch.

<sup>18</sup>Hadavandi, Esmail, et al.

ARIMA models to overcome the shortcomings in non-linear applications<sup>19</sup>. Wu and Harris adapted a fuzzy neural network to identify an operating point dependent ARMA model<sup>20</sup>.

As we see above, there has been extensive research into machine learning algorithms to predict stock prices, but there has been comparatively less research in averaging classical models to predict stock prices. *Blue Chip Economic Indicators* is one publication that is dedicated to averaging forecasts<sup>21</sup>. The main motivation for this average is that a model that overestimates a forecast can be averaged with a model that underestimates a forecast, resulting in a more accurate prediction<sup>22</sup>. Additionally, each contributing model provides a layer of interpretability. Machine learning models are accurate, but are notoriously un-interpretable. Furthermore, the average forecast approach in this paper is easily implemented in R, while it can be very difficult to implement machine-learning models and tune their parameters.

In this paper, we investigate whether the use of covariates improves a model performance, which is a fundamental question. Averaging the predicted values from a number of models has been called an average, committee or consensus forecast in the statistical forecasting literature<sup>23</sup>. We also compute the average forecast because this gives us some insight on the nature of average forecast, a

---

<sup>19</sup>Pai, Ping-Feng, and Chih-Sheng Lin.

<sup>20</sup>Wu and Harris.

<sup>21</sup>Gaynor, 578.

<sup>22</sup>Gaynor, 578.

<sup>23</sup>Gaynor, 578.



powerful method still today. In fact, model averaging, albeit at a higher level of sophistication than that performed in this thesis is a very important area of statistics.

## 1.2 Hypotheses

In selecting time series to incorporate into our multivariate models, we chose stock prices from companies that are indicators of the global economic activity affecting Apple, Inc.. We hypothesize that Apple's stock price will have a positive relationship with the *S&P500* market index. The *S&P500* is a leading market indicator, and a composite index of 500 of the largest companies from a diverse set of industries. Apple, Inc. is a significant factor of the *S&P500*. We believe that the *S&P500* effectively models how the stock market in general is doing, so it is a good proxy for the value of money in the stock market and is likely more robust than the US dollar, or any other currency.

We hypothesize that Apple's stock price will have a negative relationship with the stock price of Microsoft, one of its main competitors. Apple and Microsoft both produce desktop computers, laptops, phones, tablets, and watches. Microsoft hardware generally runs Microsoft operating systems, so Apple and Microsoft are competing in software as well as hardware. Every time that Microsoft launches an unpopular product, we believe that its stock will depreciate and Apple's will increase (and vice versa). Accordingly, we believe their stock prices will have a negative relationship.

We hypothesize that Apple's stock price will have a positive relationship with the stock price of Texas Instruments, one of its principal suppliers of

hardware components. Texas Instruments claims to have invented the silicon transistor in 1954 and was first traded on the New York Stock Exchange in 1953. Since it is connected to Apple's supply line, we hypothesize it will have a positive relationship with the stock price of Apple.

## 2 The Data

To model Apple's stock price, we collect data from Yahoo! Finance<sup>24</sup>. The raw data on Apple stock price can easily be found by searching for Apple on Yahoo! Finance<sup>25</sup>, one just has to set the appropriate time period, and frequency. Furthermore, by setting the "Show" variable to "Historical Prices", we ensure that the data is not historically adjusted. We pulled the price data on *S&P500*<sup>26</sup>, Microsoft<sup>27</sup>, and Texas Instruments<sup>28</sup>, through the same method. For all the stock indices, we set the Time Period to Jan 01, 1990 to Oct 01, 2017, Show to Historical Prices and Frequency to Monthly.

The variable AAPL is the opening stock price for the first of the month for Apple Inc.'s stock on the New York Stock Exchange. We use the name AAPL, because that is the ticker symbol of Apple Inc. We also refer to AAPL as  $Y_t$  since it is our response variable. The data<sup>29</sup> was exported into a csv,

---

<sup>24</sup><https://finance.yahoo.com/>

<sup>25</sup><https://finance.yahoo.com/quote/AAPL/history>

<sup>26</sup><https://finance.yahoo.com/quote/GSPC/history>

<sup>27</sup><https://finance.yahoo.com/quote/MSFT/history>

<sup>28</sup><https://finance.yahoo.com/quote/TXN/history>

<sup>29</sup><https://finance.yahoo.com/quote/AAPL/history?period1=631180800&period2=>

imported into R<sup>30</sup> and then converted into a *ts* (time series) object. We get a first glimpse of the time series plot of AAPL in Figure 1

As stated earlier, Vector Autoregression and Linear Regression with correction for autocorrelated errors models, use independent variables to help model our dependent variable (Apple's price); for these models, we use the *S&P500* Market index, Microsoft's stock price and Texas Instrument's stock price.

The variable GSPC is the opening stock price for the first of the month for the *S&P500* market index on the New York Stock Exchange. We use the name GSPC, because that is the ticker symbol for the *S&P500*. The data<sup>31</sup> was exported into a csv, imported into R and then converted into a *ts* object.

The variable MSFT is the opening stock price for the first of the month for the Microsoft Corporation on the New York Stock Exchange. We use the name MSFT, because that is the ticker symbol for Microsoft. The data<sup>32</sup> was exported into a csv, imported into R and then converted into a *ts* object.

The variable TXN is the opening stock price for the first of the month for Texas Instruments Incorporated on the New York Stock Exchange. Again, we use the name TXN, because that is the ticker symbol for Texas Instruments. The data<sup>33</sup> was exported into a csv, imported into R and then converted into

---

1506841200&interval=1mo&filter=history&frequency=1mo

<sup>30</sup>R: A language and environment for statistical computing.

<sup>31</sup><https://finance.yahoo.com/quote/GSPC/history?period1=631180800&period2=1506841200&interval=1mo&filter=history&frequency=1mo>

<sup>32</sup><https://finance.yahoo.com/quote/MSFT/history?period1=631180800&period2=1506841200&interval=1mo&filter=history&frequency=1mo>

<sup>33</sup><https://finance.yahoo.com/quote/TXN/history?period1=631180800&period2=1506841200&interval=1mo&filter=history&frequency=1mo>

Period	Start	End	Length
Training	January 1990	September 2016	321
Test	October 2016	September 2017	12

Table 1: Training and Test Periods.

a ts object.

The data from Yahoo! Finance includes the opening, high, low and closing price for each date. For each variable, we investigate only the opening price for simplicity and consistency, although the same models and results should hold if we chose closing price. The lengths of the time series AAPL, GSPC, MSFT and TXN are all 333; this is one data point from the first of every month from January 1, 1990 to September 1, 2017. We do not aggregate the data to get monthly readings, because it came directly in the desired frequency. To fit our models, we do so with a training set of the first 321 observations (January 1990 - September 2016); which leaves a with a 12 month (October 2016 - September 2017) test set. When we forecast our models, we do so for October 2016 - September 2017 and compare these predicted values to our test set. Table 1 shows the start and end dates for the training and test data.

### 3 Descriptive Time Series Analysis of the Data

Consider the time series plot of the Apple stock price in Figure 1. We see that starting in 2005, the stock price rises with a roughly exponential trend. There are some noticeable fluctuations in Figure 1, but these do not appear to be seasonal trends, as the period is too wide.

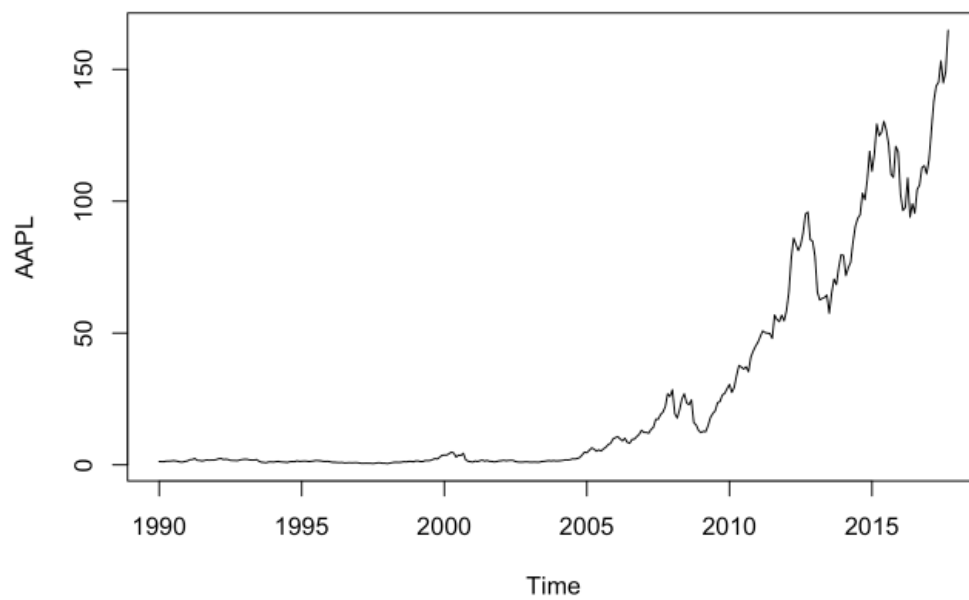


Figure 1: AAPL Opening Price

Due to the large range of data in this plot, it appears as if the stock price of Apple is roughly constant from 1990 - 2004. In Figure 2, we focus in on this time period, and see that there are major fluctuations.

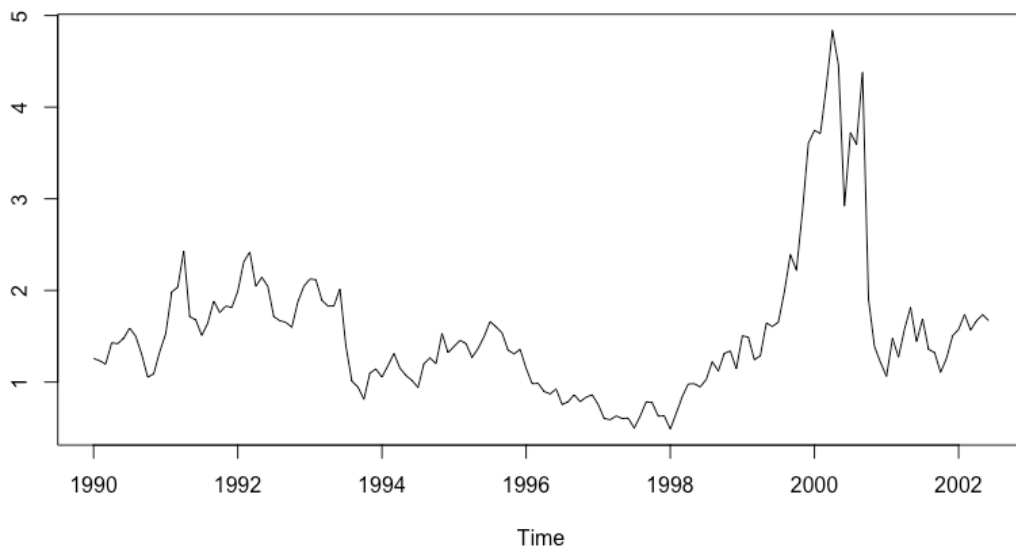


Figure 2: AAPL Montly Opening Price 1990 - 2004

Consider the time series plots of our covariates. In Figure 2(a), the time series of GSPC has, overall, trended positively from 1990 to 2017. The stock price has a lot of stochastic trends. There are 2 distinct valleys in Figure 2(a), the first around 2000-2002 and the second around 2008. These periods of negative trend coincide with the US economy's overall behavior, which makes sense, considering the *S&P500* is a market index comprised of 500 companies influential in the stock market.

Figure 2(b) shows the time series of MSFT, and is characterized by 2 periods of exponential growth, 1995-2000 and 2010-2016. In between these prosperous periods, there is a stretch where the MSFT price oscillates around a constant price, with nearly constant variability.

Figure 2(c), the time series of TXN follows a similar pattern to Figure 2(b), MSFT. There is exponential growth from 1995-2000 and 2010-2016. Figure

2(c) has a sharper correction from 2000-2001 than 2(b) and the interstitial period between the price increases has a little more variability than the same period in 2(b).

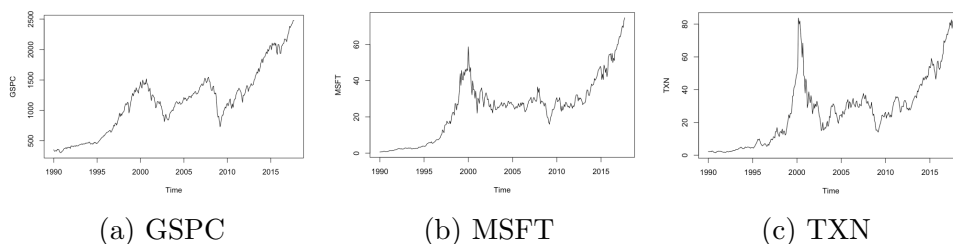


Figure 3: Opening Prices of the Covariates

Immediately apparent in Figures 1 and 3 is the irregularity in each of the time series. This is all very interesting, and there are lots of patterns here that merit further investigation. The next step is to decompose and analyze the time series of AAPL.

### 3.1 Time Series Decomposition of AAPL

Before we explore the time series of AAPL in depth, we need to define several terms that are central to time series analysis. First and foremost, a time series is generally regarded as one realization of an underlying stochastic process<sup>34</sup>. A stochastic, or random, process is a collection of indexed random variables. For time series, we do not distinguish between the stochastic process and its observed realization, for ease of notation. There is a key assumption of ergodicity embedded in this understanding of a stochastic process. Ergodicity has

---

<sup>34</sup>Kirchgassner, 12.

several conditions; first, we assume that the mean and variance of a realization of a stochastic process and the underlying process itself converge as time becomes large. Additionally, for a series to be ergodic, the stochastic process has to be stationary, or in statistical equilibrium<sup>35</sup>. A time series process at time  $t$ ,  $Y_t$  is considered weakly, or second-order, stationary if it meets the following conditions:

- (a)  $E(Y_t)$  the expected value is constant across time, and
- (b)  $Cov(y_t, y_{t+k})$  depends only on  $k$  and not on  $t$ <sup>36</sup>, where  $t$  is time.

A process is considered to be strictly stationary if the distribution of the time series is constant for all time periods; strict stationarity is difficult to verify, so we tend to look for weak stationarity<sup>37</sup>. A model is second-order stationary if it has stationary mean, stationary variance, and the correlation between variables is dependent on only the lag, or number of readings, between them<sup>38</sup>. If a time series is second-order stationary, we can define the auto-covariance function (acvf) as a function of the lag,  $k$ , where

$$acvf_k = \gamma_k = E[(x_t - \mu)(x_{t+k} - \mu)]^{39},$$

where  $\mu$  is the mean of the series.

Now, we decompose the Apple stock, using multiplicative decomposition,

---

<sup>35</sup>Kirchgassner, 13.

<sup>36</sup>Chan, 17.

<sup>37</sup>Chan, 16-17.

<sup>38</sup>Cowpertwait and Metcalfe, 33.

<sup>39</sup>Cowpertwait and Metcalfe, 33.



and inspect the different elements. Multiple decomposition of a time series,  $x_t$  means we use the model

$$x_t = m_t * s_t * z_t$$

where  $m_t$  is the trend of the data,  $s_t$  is the seasonal effect and  $z_t$  is an error term that is usually serially correlated, and it is the term that we model<sup>40</sup>. Additive decomposition, which uses the model:

$$x_t = m_t + s_t + z_t$$

was also considered, but rejected since it did not fit the data as well. Note that the following plots are on the raw values of AAPL time series, not the logged values. First, we look at the trend extracted from the multiplicative decomposition of AAPL using the *decompose()* function in R in Figure 4.

---

<sup>40</sup>Cowpertwait and Metcalfe, 13-14.

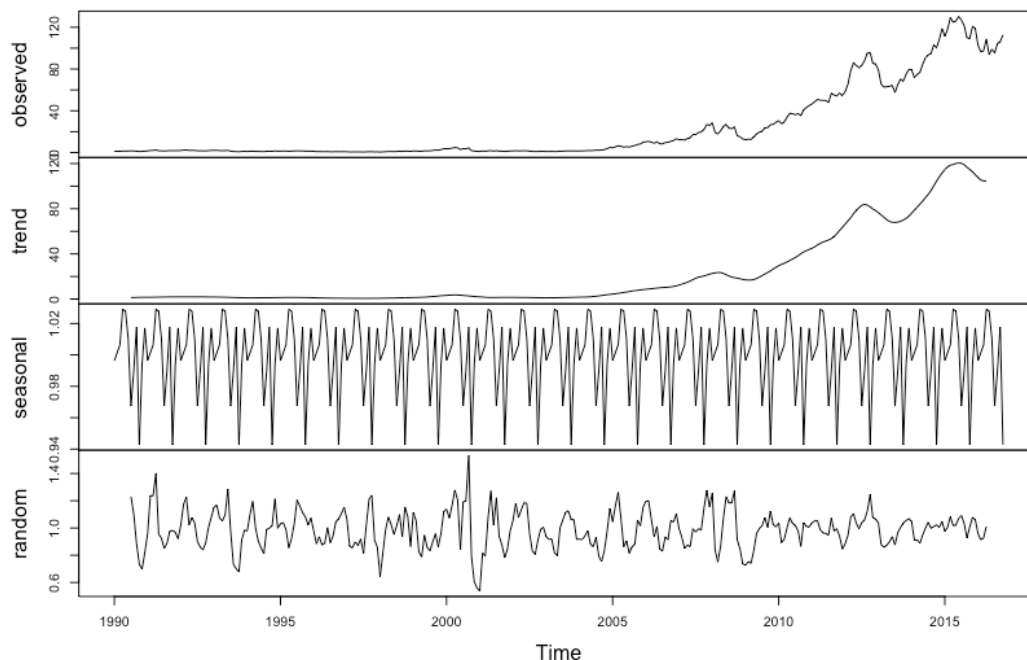


Figure 4: AAPL Multiplicative Decomposition Components

In Figure 4, we see that the trend is not linear, but appears to reflect exponential growth with 3 noticeable corrections in 2008, 2012, 2015. Apple's tumultuous history and near-failure has been well documented, so we are not surprised to see these significant corrections.

Now, consider the seasonal part of the multiplicative decomposition of AAPL in Figure 4. In this plot, the seasonal effect has the exponential trend removed. We see the seasonal variation is very small; it does not provide strong evidence for a significant seasonal effect. In Figure 5, we see a boxplot of price for each month, which confirms the lack of seasonal effect.

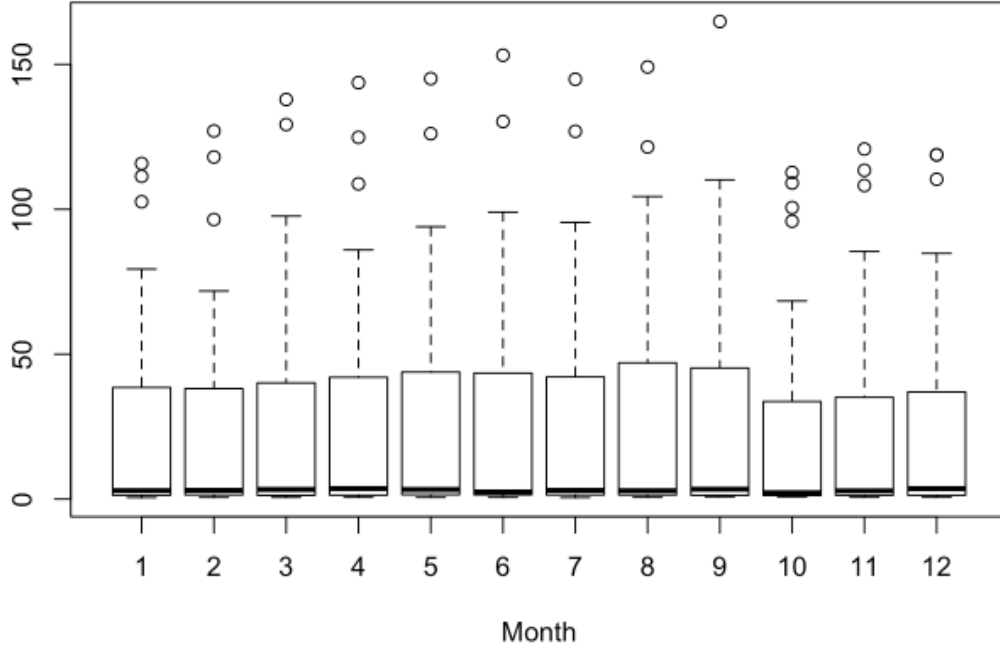


Figure 5: AAPL Seasonal Boxplot

Figure 5 shows insignificant seasonality, as each month has approximately the same median and variation. Figure 4 and 5 indicate that AAPL does not vary much by month, which is not surprising, seeing as it is a large company's stock price, and that their product launches do not follow the annual calendar with regular periods. We look at the random component obtained from multiplicative decomposition of AAPL in Figure 4. In this plot, the random component has the exponential trend and the seasonal effect removed. The random component in Figure 4 appears to have a constant mean of 1. For a random component to be second-order stationary, it has constant variance across different 'time windows'. In Figure 4, the mean is mostly constant, around 1. The random component in Figure 4 shows no clear departures from second-order stationary. Many transformations to AAPL were considered, in-

cluding the logarithm and square root, but these transformations made the data less stationary according to the diagnostics we used above.

### 3.2 The Sample Autocorrelation Function

Autocorrelation, or serial correlation, is the correlation of a variable with itself at different times and the sample autocorrelation is defined as

$$\rho(k) = \frac{Cov(x_t, x_{t+k})}{\hat{\sigma}^2} = \frac{\gamma_k}{\sigma^2},^{41}$$

where  $Cov(x_t, x_{t+k})$  is the covariance at lag  $k$  and  $\hat{\sigma}^2$  is the estimated variance of the time series. We estimate  $\rho(k)$  with the sample autocorrelation function, and the estimate of  $\sigma^2$  from the data. Autocorrelation is frequently visualized with a sample correlogram (or ACF), which plots the sample autocorrelation,  $r_k$ , which is unit-less, on the y-axis against discrete lags,  $k$ , on the x-axis<sup>42</sup>. The formula for the autocorrelogram is

$$r_k = \frac{\gamma_k}{\sigma^2}^{43}.$$

Statistical significance of the sample autocorrelation is determined at the threshold of

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}.^{44}$$

---

<sup>41</sup>Chan, 17.

<sup>42</sup>Cowpertwait and Metcalfe, 36.

<sup>43</sup>Cowpertwait and Metcalfe, 36.

<sup>44</sup>Cowpertwait and Metcalfe, 36.

If  $r_k$  is above or below the dotted lines at these values, then we say that the sample autocorrelation is statistically significant at the respective lags. This indicates that there is some pattern or relationship between the random components. We look at the sample ACF of the random component of AAPL in Figure 6:

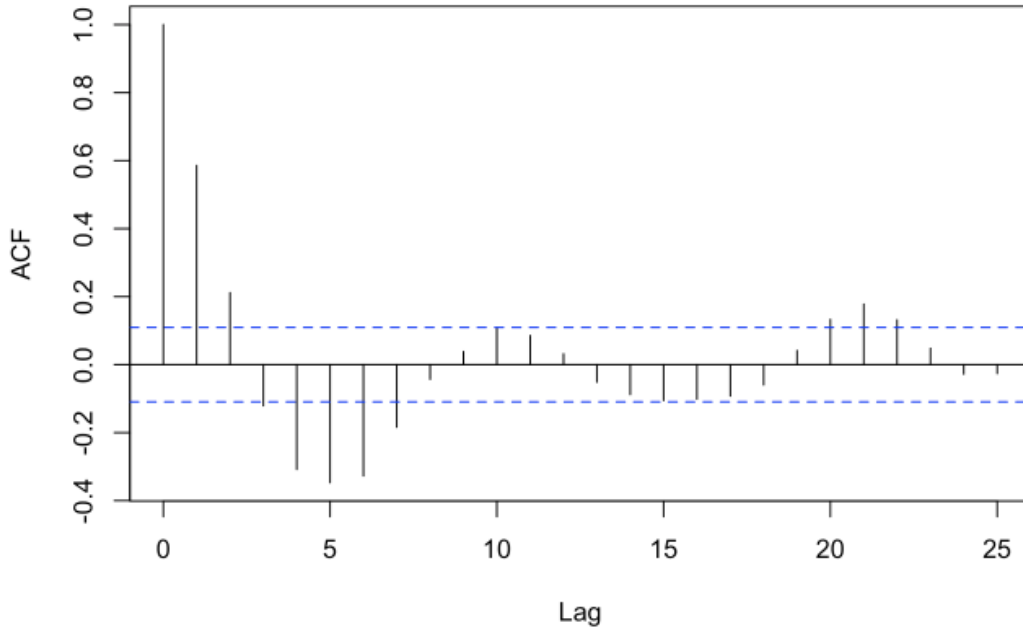


Figure 6: ACF of Random component of AAPL obtained by multiplicative decomposition

For stationary process, we expect the ACF to taper off very quickly to values below statistical significance<sup>45</sup>. Figure 6 has statistically significant autocorrelations for lags  $k = 1, 2, 3, 4, 5, 6, 7$ , but the autocorrelations follow a dampened sinusoidal pattern and die down very quickly. Figure 6 is the ACF of a stationary series. We want to develop a model for this relationship

---

<sup>45</sup>Chan, 17.

between random effects. We exclude the similar decompositions of the time series of GSPC, MSFT or TXN. We simply note that the same overarching patterns were observed - upward trend, insignificant seasonality, and a dampening sinusoidal random component's ACF. So we conclude that the random component of the multiplicative decomposition of the time series of AAPL is stationary, therefore pursuing modeling of this time series is justified.

## 4 ARIMA Model

$ARIMA(p, d, q)$  models are univariate models fit to stationary time series.  $ARIMA(p, d, q)$  models combine Autoregressive (AR(p)), Integrated (I(d)), and Moving Average (MA(q)) models, and the parameters,  $p, d, q$  represent the order of the AR, I, and MA models, respectively. The integrated component,  $d$  stands for the order of differencing needed to make the time series stationary. This means, for example,

$$(1 - B)^d x_t = s_t,$$

where  $s_t$  is stationary and  $B$  is the backshift operator, which means

$$Bx_t = x_{t-1}.$$

. A stationary time series,  $x_t$  follows an  $ARMA(p, q)$  process when it can be described as

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \dots + \alpha_p x_{t-p} + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \dots + \beta_q w_{t-q},$$

where  $w_t$  is a white noise process <sup>46</sup>. For example, for an  $ARMA(1, 1)$  model, we have

$$x_t = \alpha_1 x_{t-1} + w_t + \beta_1 w_{t-1} = w_t + (\alpha_1 + \beta_1) \sum_{i=1}^{\infty} \alpha^i w_{t-i}.$$

From here, we can see that  $E[x_t] = 0$ , since  $E[w_t] = 0$ . Furthermore, it follows that

$$Var(x_t) = Var[w_t + (\alpha_1 + \beta_1) \sum_{i=1}^{\infty} \alpha^i w_{t-i}] = \sigma_w^2 + \sigma_w^2 (\alpha_1 + \beta_1)^2 (1 - \alpha^2)^{-1},$$

And

$$Cov(x_t, x_{t+k}) = (\alpha_1 + \beta_1) \alpha_1^{k-1} \sigma_w^2 + (\alpha_1 + \beta_1)^2 \sigma_w^2 \alpha^k (1 - \alpha^2)^{-1} \text{<sup>47</sup>}. \text{<sup>48</sup>}$$

And the autocorrelation is given by,

$$\rho_k = \frac{\alpha_1^{k-1} (\alpha_1 + \beta_1) (1 + \alpha_1 \beta_1)}{1 + \alpha_1 \beta_1 + \beta_1^2} \text{<sup>48</sup>}. \text{<sup>49</sup>}$$

The derivations for higher order  $ARMA(p, q)$  models are more difficult to derive, but they are not required for this paper's application.

An ARIMA model strategy consists of

1. making the variance stationary,
2. identify the orders  $d, p, q$  of the model, in that order,
3. fitting the model,

---

<sup>46</sup>Cowpertwait and Metcalfe, 127.

<sup>47</sup>Cowpertwait, 128.

<sup>48</sup>Cowpertwait and Metcalfe, 129.

4. diagnostics - determining whether the residuals series has significant autocorrelations
5. start over if the diagnostic is not positive.

#### 4.1 Identification of $ARIMA(p, d, q)$

We used first order differencing to make our time series AAPL stationary. We also considered log transformation to the time series, but log-transformation made the variance of the random component from multiplicative decomposition more erratic. Therefore we fit the ARIMA model on the un-transformed data.

To identify the model, we first determined the order of differencing,  $d$ , that makes that time series stationary. We determined that  $d = 1$  makes the resultant time series stationary. Differencing is done before fitting an Autoregressive Moving Average (ARMA) model. We determine the order,  $p$ , of an  $AR(p)$  model by looking for characteristic behavior in the ACF and the PACF. Specifically, the ACF does not break off, while the PACF breaks off at lag  $p$ <sup>49</sup>. The order,  $q$ , of  $MA(q)$  models can be determined similarly; here, the ACF breaks off at  $q$ , while the PACF does not break off<sup>50</sup>.

In Section III, we conclude that our times series,  $Y_t$  has a strong upward trend and a weak seasonal component. Also, in Section III, we conclude it is most appropriate to develop a model on the untransformed time series,  $Y_t$ . In

---

<sup>49</sup>Kirchgassner, 67.

<sup>50</sup>Kirchgassner, 67.



this section, we develop an appropriate ARIMA model for  $Y_t$ . Remember that we hold out the last 12 observations in the time series as a test set, and unless otherwise noted, the following plots and data will be based on the January 1990 - September 2016 training set. The ACF of the differenced time series ( $d = 1$ ) is in Figure 7:

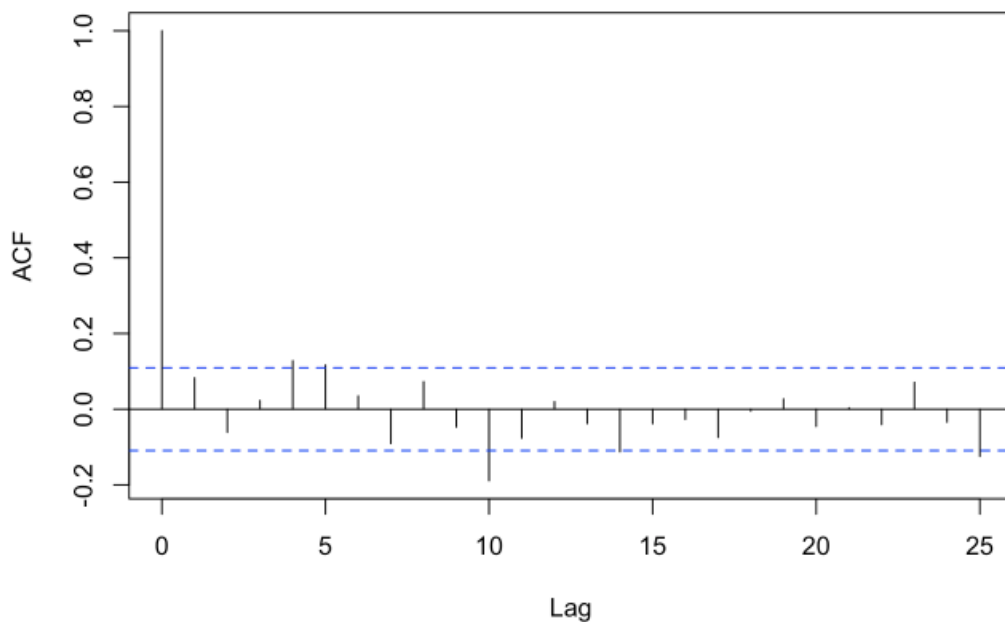


Figure 7: ACF  $(1 - B)Y_t$

In Figure 7, there are slightly statistically significant autocorrelations at lag values  $k = 4, 10, 25$ , but this does not motivate higher-order differencing, since this ACF does not have strongly significant autocorrelations. In most autocorrelation functions of sample data, as in any summary statistics, where significance is determined at  $\alpha = 5\%$ , we should expect about 5% of the spikes will be borderline significant, at random lags, even if there is no signal in the data.

We are satisfied with a model when its residual error series has no strongly significant autocorrelations, as this indicates that the model accounts for all autocorrelations in the time series. Since the ACF in Figure 7 does not begin with a string of statistically significant autocorrelations, there is no need to difference the time series by more than 1. We conclude that Figure 7 does not have any statistically significant autocorrelations.

A stochastic process,  $w_t$  is considered to be white noise, or purely random, if it has the following properties:

$$\begin{aligned} E[w_t] &= 0, \\ Var[w_t] &= \sigma^2 \end{aligned}$$

and

$$Cov[w_i, w_j] = 0 \text{ when } i \neq j \text{ }^{51}.$$

Connected to white noise is the important concept of a random walk, which is frequently encountered in financial data. A random walk (without drift),  $x_t$ , is defined as the summation of white noise. In other words, if we have a time series  $Y_t$  such that  $y_t - y_{t-1}$  is white noise, it follows that  $Y_t$  follows a random walk. A random walk is often used to assess the efficacy of more complicated models<sup>52</sup>. It is also useful to consider a random walk with drift process, which is modeled by the following

$$y_t = y_{t-1} + \delta + w_t.$$

---

<sup>51</sup>Kirchgassner, 14.

<sup>52</sup>Cowpertwait and Metcalfe, 67.

We consider higher-order and seasonal differencing, but these strategies produced ACF's with statistically significant auto-correlations, and thus imposed patterns on the data. Accordingly, we conclude that first-order differencing was the most appropriate. Accordingly, our chosen model for this time series is a random walk<sup>53</sup>. We have a satisfactory method for making the data stationary - the regular differenced time series. This meets all our requirements and since we do not see significant seasonal effect in exploratory data analysis we should not expect seasonal differencing to be beneficial.

## 4.2 Fitting our ARIMA Model

Now, since the ACF of the regularly differencing  $Y_t$  time series does not have statistically significant autocorrelations, we conclude that this new time series is stationary. Recall that our entire model fitting is done on the training set of 321 observations from January 1990 - September 2016. This stationary time series that we will work with moving forward is denoted

$$(1 - B)Y_t = w_t.$$

Now, we fit our model using the *arima()* function in R. There are no parameters for us to estimate the significance of in this model, which is typical of financial data. The forecasting or estimation model is a random walk, denoted by

$$\hat{y}_t = y_{t-1}.$$

---

<sup>53</sup>Cowpertime, 71.

Therefore, the only parameter we fit is the variance.

Since the ACF in Figure 7 does not begin with statistically significant values, this indicates the model does not have an autoregressive component. Furthermore, since the ACF in Figure 8 does not have statistically significant autocorrelations, we conclude that the model does not have a Moving Average (MA) component. Accordingly, the most appropriate ARIMA model for the logged time series is  $ARIMA(0, 1, 0)$ .

### 4.3 ARIMA Model Diagnostics

To perform model diagnostics on our  $ARIMA(0, 1, 0)$  model, we look at the ACF of the model's residuals in Figure 8.

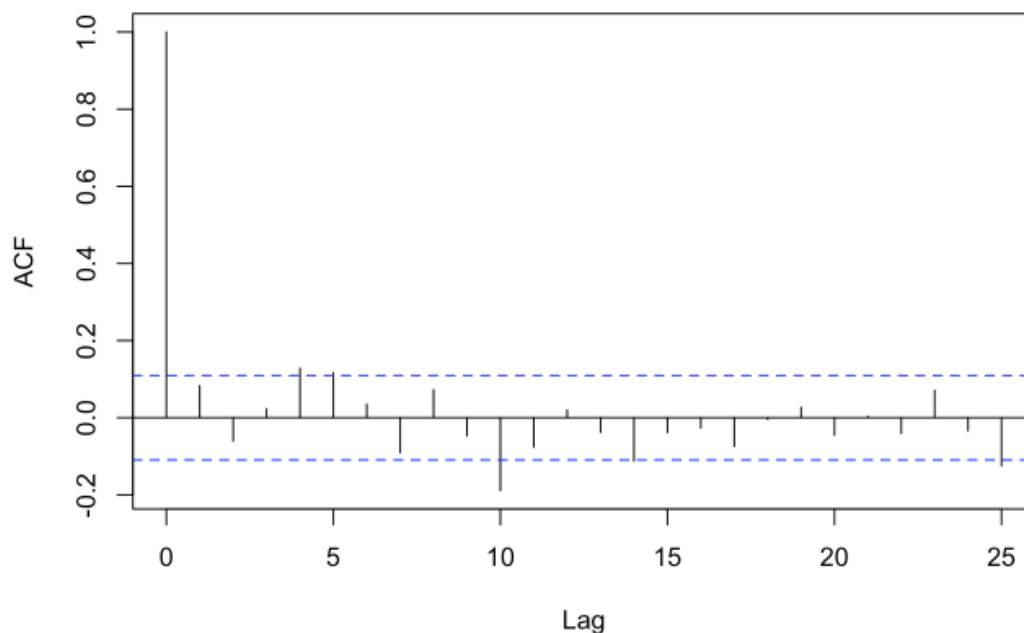


Figure 8: ACF of Residuals of ARIMA (0,1,0) on  $Y_t$

The ACF in Figure 8 does not have statistically significant autocorrelations,

which dissuades us from further complicating the ARIMA model. Since these residuals do not exhibit significant autocorrelation, we are satisfied with our  $ARIMA(0, 1, 0)$  model. As a further model diagnostic, we run the Ljung-Box test on the model's residuals.

$$H_o : \rho_1 = \rho_2 = \dots \rho_k = 0$$

$$H_a : \text{not all } \rho_i = 0$$

The Ljung-Box Test evaluates the null hypothesis that the autocorrelations are jointly zero, and the alternative hypothesis, that the autocorrelations are not jointly zero. We run the Ljung-Box test on the residuals from our  $ARIMA(0, 1, 0)$  model which returns a  $p - value = 0.135$ . Accordingly, we do not have evidence to reject the null hypothesis, and we conclude that the autocorrelations of the  $ARIMA(0, 1, 0)$ 's models' residuals are jointly zero. If the residual error series does not have statistically significant autocorrelations, then we cannot reject the null hypothesis.

## 4.4 Forecasting with ARIMA Model

Now that we determined that the  $ARIMA(0,1,0)$  model is appropriate, we forecast the model 12 months ahead and compare these predicted values to the observed values we removed from the training set. Recall that our held-out test set is the 12 observations from October 2016 - September 2017. We forecast with the model and provide the predicted values and compute the root mean square error in Table 3 which is shown in the Conclusions Section. We see that the forecasted value in Figure 9 simply the previous value, which

is what we expected from an  $\text{ARIMA}(0,1,0)$ . In Figure 9, the observed time series is in black and the forecast is in red.

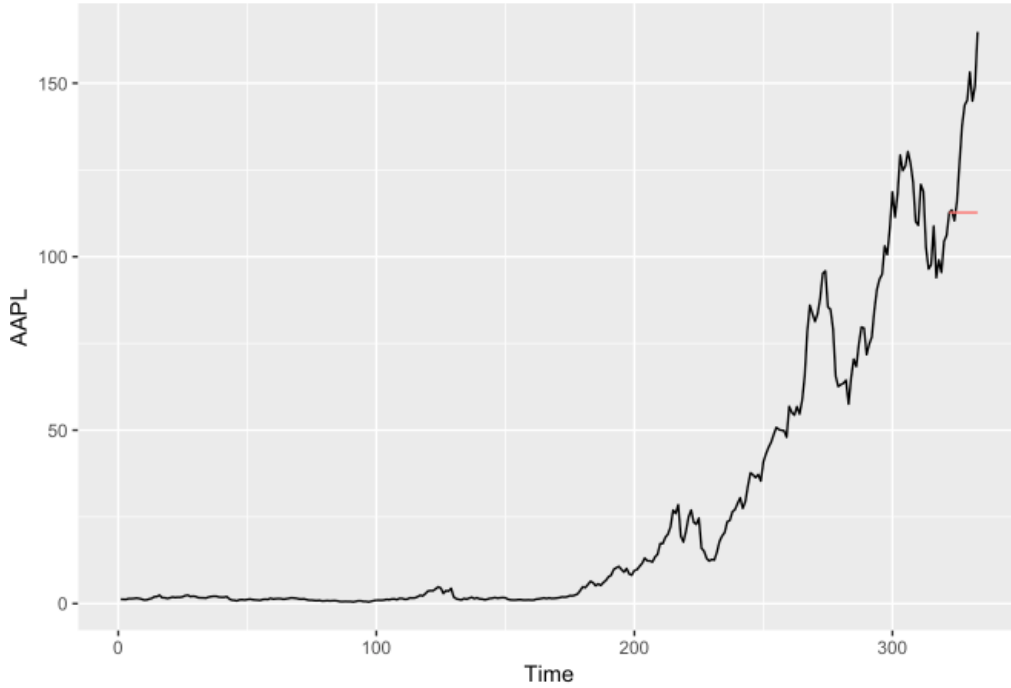


Figure 9: ARIMA Forecast

As we see in the forecast in Figure 9, this model does not attempt to model the fluctuations in the time series, it just predicts a flat value, with a standard error that increases slightly with time. We refine this model by using the GARCH methodology.

## 4.5 ARIMA + GARCH Model

Stock market data is known for its volatility, so we have good reasons to believe that AAPL has volatility. A time series is volatile if the ACF its residual errors series does not have statistically significant autocorrelations and the ACF of

its squared values has significant autocorrelations. An important first step is to check the residuals of our ARIMA model for volatility. We inspect two ACFs. First, we recall that the ACF of the ARIMA model's residuals do not have statistically significant autocorrelations; we see this with Figure 7. The second ACF we inspect is that of the ARIMA model's squared residuals - we see this in Figure 10.

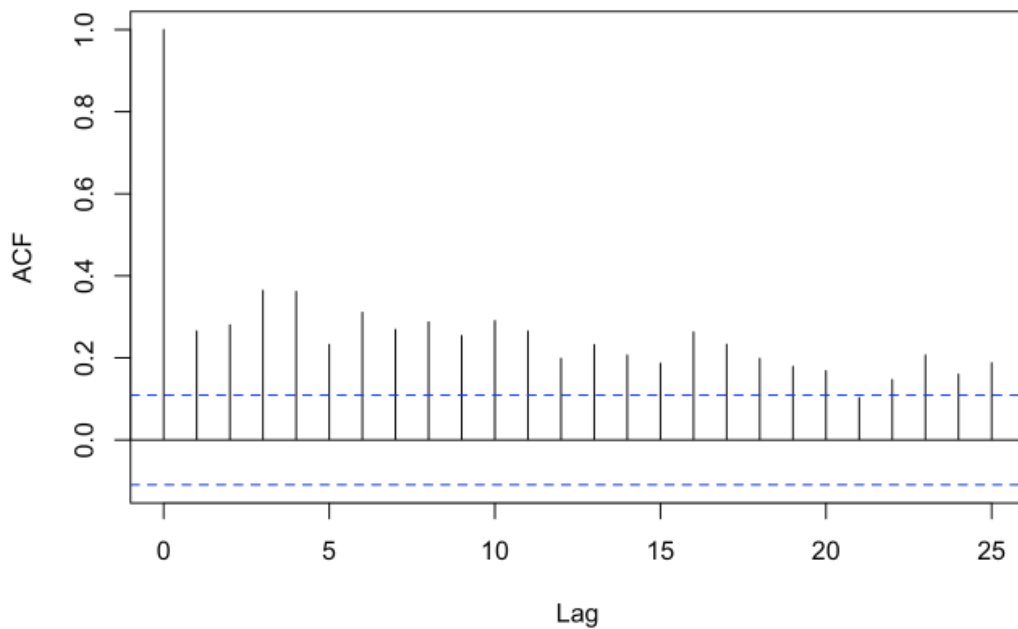


Figure 10: ACF of Squared Residuals  $ARIMA(0, 1, 0)$

Figure 10 has statistically significant autocorrelations, which indicates volatility, or conditional heteroscedasticity, in our ARIMA model's residuals. This means that the observed variance follows a stochastic trend and merits further modeling, which we do with a GARCH (Generalized Conditionally Heteroscedastic) model. The  $GARCH(p, q)$  model is denoted

$$h_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_q \epsilon_{t-q}^2 + \beta_1 h_{t-1}^2 + \dots + \beta_p h_{t-p}^2,$$

where

$$h_t = E[\epsilon_t^2 | y_{t-1}, y_{t-2}, \dots, y_1]^{54}.$$

In the above notation,  $\epsilon_t$  is our residual term,  $h_t$  is our estimate of the squared residual term, and  $y_t$  is our observed time series, all at time  $t$ .  $GARCH(p, q)$  model allow us to parsimoniously parameterize conditional variances, and, for financial data, a  $GARCH(1, 1)$  is often sufficient, as it is in this case<sup>55</sup>. After we determine the orders  $p, q$  of our GARCH model, we use R to estimate our parameters  $\alpha, \beta$ .

To fit the  $GARCH(p, q)$  model, we use the *garch()* function in the *library(tseries)*. We try several different  $GARCH(p, q)$  values and use AIC as a validation metric to determine the optimal values of  $p$  and  $q$  that work best on these residuals. We try many different GARCH models and the two models  $GARCH(1, 0)$  and  $GARCH(0, 2)$  perform significant better than the others in terms of AIC (recall that the lower the AIC, the better performing the model). Here  $GARCH(1, 1)$  is the best performing model with  $AIC = 794.9447$ . For the sake of model parsimony, we do not fit GARCH models with parameter values greater than 3. Our best model  $ARIMA(0, 1, 0) + GARCH(1, 1)$  is denoted by:

$$Y_t = Y_{t-1} + w_t \sqrt{0.00325 + 0.7244\epsilon_{t-1}^2 + 0.5444h_{t-1}}.$$

An important model diagnostic is checking the ACF of our  $GARCH(1, 1)$  model's residuals and squared residuals. We look at these in Figure 11

---

<sup>54</sup>Kirchgassner, 292.

<sup>55</sup>Kirchgassner, 293-294.



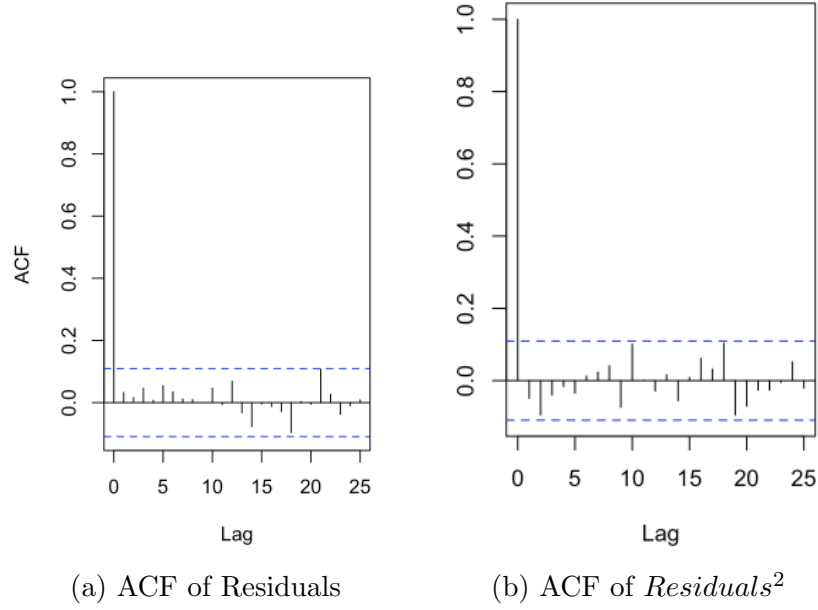


Figure 11: ARIMA(0,1,0) GARCH(1,1)

Neither ACFs in Figure 11 have statistically significant autocorrelations at non-zero lags. Therefore, our  $GARCH(1,1)$  model fits our  $ARIMA(0,1,0)$  model's residual satisfactorily.

## 4.6 Forecasting AAPL with ARIMA + GARCH Model

Now that we determined that we cannot and should not develop a more complicated model for the residuals, we forecast our  $ARIMA(0,1,0)+GARCH(1,1)$  model. Again, refer to Table 3 in the Conclusions Section to see the forecasted values and the computed RMSE. Figure 12 visualizes the forecasted values in red and the AAPL time series is in black.

We see from Table 3 that the adding the GARCH model to our ARIMA model dramatically reduces the model's forecasting performance on the test

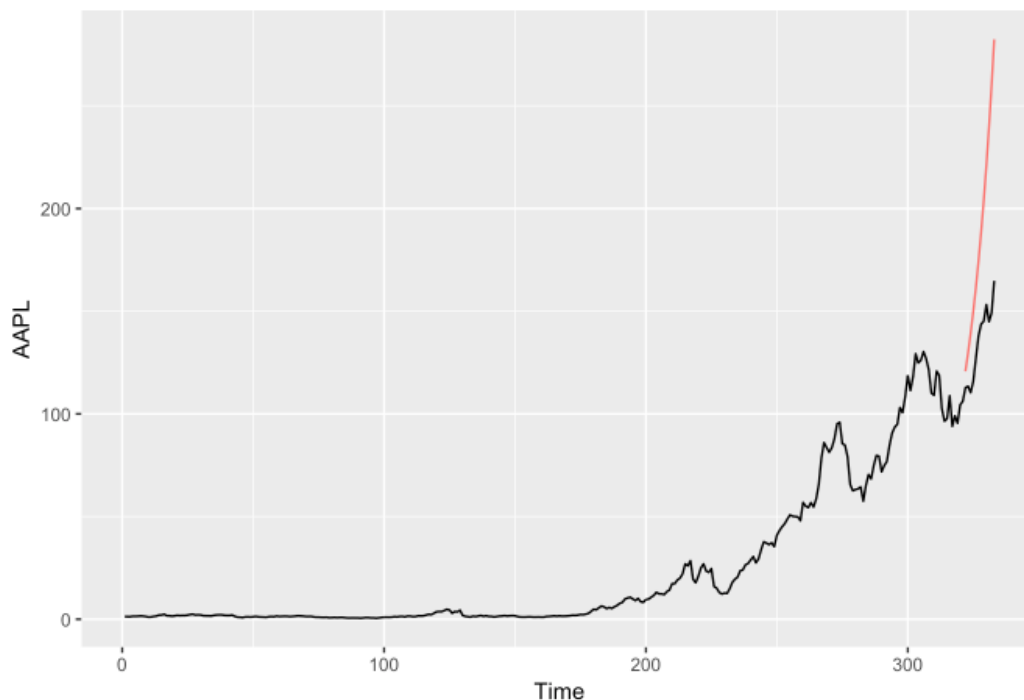


Figure 12: ARIMA(0,1,0) GARCH(1,1) Forecast

set. The GARCH model tries to model the volatility and as the time gets further away from our training set, the predicted values skyrocket away from the observed values.

## 5 Exponential Smoothing

Exponential smoothing is a concept related to moving average, where earlier observations are given exponentially diminishing weights. For data with a strong trend and changing mean like AAPL, the Holt-Winters model that is the most appropriate is the Holt-Winters trend corrected exponential smoother; in this model, we essentially fit a series of lines, with changing slope and intercept

over time, denoted

$$\begin{aligned}\hat{Y}_t &= a_t + b_t t + \epsilon_t, \text{ where} \\ a_{t+1} &= a_t + \alpha(y_t - a_t) + b_t \\ b_{t+1} &= \beta(a_{t+1} - a_t) + (1 - \beta)b_t.\end{aligned}$$

It is easy to think of  $a_t$  as the estimated intercept, and  $b_t$  the estimated slope, of the fitted model at time  $t$ . In the late 1950's Holt and Winters generalized this model to update the estimates of the non-stationary mean (also called level), slope (or trend) and seasonals<sup>56</sup>. Our data does not display significant seasonality, so we do not pursue that here. The parameters,  $\alpha, \beta$  are easily estimated in R with the *HoltWinters()* function. The forecasting equation for some future value of  $y_t$  is given by adding the trend component,  $b_t$  and the level component,  $a_t$ .

We saw already that our data has a strong trend. Additionally, we have already shown that our target variable does not exhibit significant seasonality; this was explicitly seen in Figure 6. Since we do not have seasonality in this time series, there is no justification to perform Holt-Winters seasonal exponential smoothing. Accordingly, we use Holt-Winter's trend corrected non-seasonal exponential smoothing model to this time series, meaning we estimate two parameters,  $\alpha$  and  $\beta$ . We fit this model to our training set using the *HoltWinters()* function in R. This returns  $\alpha = 1$  and  $\beta = 0.01017029$ . When we condense our notation, our forecasting equation is

$$Y_t = 112.7099 + 0.6296 * t + e_t.$$

---

<sup>56</sup>Cowpertwait and Metcalfe, 59.

## 5.1 Forecasting our Holt-Winters Model

We use this model to forecast 12 months ahead for the test period October 2016 - September 2017 and compare the values predicted by Holt-Winters to our test data. We record the predicted values and test RMSE in Table3 and visualize the same in Figure 13 (again, forecasted in red, observed in black).

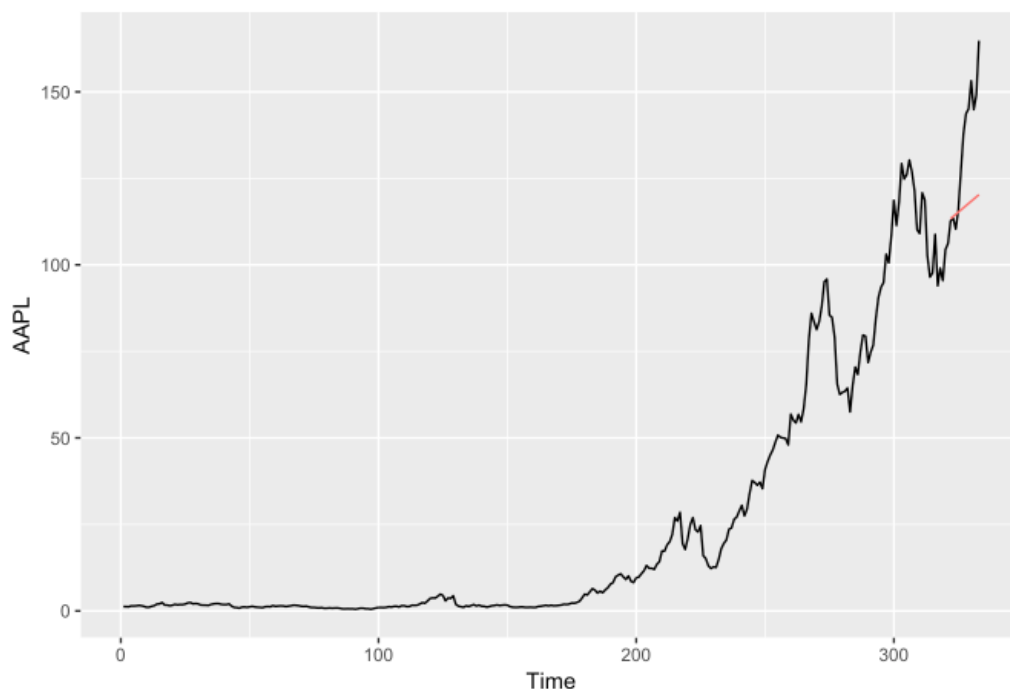


Figure 13: Holt-Winters Forecast

Figure 13 shows this model performs fairly well in the short-term, but is significantly less accurate in long-term forecasts. The exponential smoothing model captures some of the trend, which makes sense, because we tell the *HoltWinters()* function to compute a parameter for the trend. This model performs very well in comparison to the other models as seen in by its forecasting performance listed in Table 3 in the Conclusions Section. Comparing

Figure 13 to Figure 21 shows that the Exponential Smoothing model is slightly more conservative in modeling the trend than other models, which accounts for sub-optimal test set RMSE performance.

## 6 Vector Autoregression

In this section, we fit a Vector Autoregression (VAR) model for the AAPL time series. Vector Autoregression is a method that assumes that all variables are jointly endogenous, which means everything depends on everything<sup>57</sup>. A  $VAR(p)$  model has order,  $p$ , where each variable is modeled linearly on all variables in to the system, including itself, up to lag  $p$ <sup>58</sup>. In this thesis the  $VAR$  model we assume gives us a system of four simultaneous equations models of AAPL, and the 3 covariates. These systems of simultaneous equations allow us to evaluate the dynamics of shocks to the system composed of these 4 time series. The VAR model starts with the  $k$ -dimensional stochastic process  $X$ , and the  $VAR(p)$  process is conceptualized as

$$X_t = \delta + A_1X_{t-1} + A_2X_{t-2} + \dots + A_pX_{t-p} + U_t,$$

where  $A_i$  are  $k$ -dimensional matrices and  $U$  is a residual vector<sup>59</sup>.

This notation is conensed to

$$A(L)X_t = \delta + U_t,$$

---

<sup>57</sup>Kirchgassner, 127.

<sup>58</sup>Kirchgassner, 128.

<sup>59</sup>Kirchgassner, 129.

where,

$$A(L) = I_k - A_1L - A_2L^2 - \dots - A_pL^p,$$

$$E[U_t] = 0,$$

$$E[U_tU_t'] = \Sigma_{\hat{u}\hat{u}}, \text{ and}$$

$$E[U_tU_s'] = 0 \text{ for } t \neq s^{60}.$$

We fit the VAR model in Section 6.1 - 6.6, we inspect the system's Impulse Response Functions in Section 5.7, and we forecast AAPL with this VAR model in Section 6.8.

## 6.1 Unit roots and co-integration tests

We begin by testing our dependent and independent variables for unit root using the augmented Dickey-Fuller test. For the Augmented Dickey-Fuller test, the null hypothesis states that a unit root is present in the time series. We run the test in R using `tseries::adf.test()`. The smallest p-value returned by the test was 0.386 for TXN, the other 3 time series had p-values greater than 0.80. The Augmented Dickey-Fuller test fails to reject the null hypothesis for each of these time series and concludes that all 4 of our time series are non-stationary. This means the ADF test concludes that all of these series follow a random walk, which is typical for stock market data and confirms what we learned earlier.

To avoid against spurious relationships between variables in a dynamic

---

<sup>60</sup>Kirchgassner, 129

system, such as the VAR system, we test for co-integration<sup>61</sup>. Co-integrated time series have a common long-run trend<sup>62</sup>. Co-integration has been researched heavily, and there are many different methods for evaluating it, but we use the ADF test with some modifications in R to test for co-integration in each pair of our un-differenced time series. For these tests, the null hypothesis is that the time series are co-integrated, and the alternative hypothesis is that the time series are not co-integrated. The test concludes that AAPL is co-integrated with all 3 of the covariates, with the lowest p-value = 0.587, from the test with the *S&P500*.

## 6.2 Differencing Covariates

It is a good strategy to model a Vector Autoregression model using stationary covariates<sup>63</sup>. We explored how differencing can be used to make a time series stationary with AAPL. It suffices to say that the same strategy worked for our 3 covariates - that single order differencing is the most effective strategy. This was expected because all of the time series are of financial data which is known to exhibit behavior indicative of a random-walk. In summation, all of the time series needed to be differenced once to become stationary. We will refer to the first order differenced time series as follows:

$$AAPL*_t = (1 - B)AAPL_t$$

---

<sup>61</sup>Kirchgassner, 207.

<sup>62</sup>Kirchgassner, 209.

<sup>63</sup>Kirchgassner, 207.

$$MSFT*_{*t} = (1 - B)MSFT_t$$

$$GSPC*_{*t} = (1 - B)GSPC_t$$

$$TXN*_{*t} = (1 - B)TXN_t.$$

### 6.3 Cross-correlations

First, we inspect the decompositions of each time series vs. the decompositions of each time series log-transformed. We conclude that, like the AAPL time series, none of the time series benefit from the log-transformation. In each case, the original data's decomposed random component has greater first and second order stationarity than the logged counterpart, so we keep the unlogged data. Now that we isolated stationary time series, we inspect the cross-correlations between the series. Figure 14 has the cross-correlation of AAPL and each of our three independent, stationary time series.

Across Figures 14, we do not see a single statistically significant cross-correlation. This tells us that AAPL is not influenced by any of the other series, at any lag values; furthermore, AAPL does not influence any of these series at any lags. This indicates that when we fit a VAR model it will not likely be of high-order.

### 6.4 Fitting the VAR model

We use Akaike Information Criterion (AIC) as a metric to compare the efficacies of the different order VAR(p) models we evaluate, with

$$AIC(p) = \log|\sum_{\hat{u}\hat{u}}(p)| + (k + pk^2)\frac{2}{T},$$



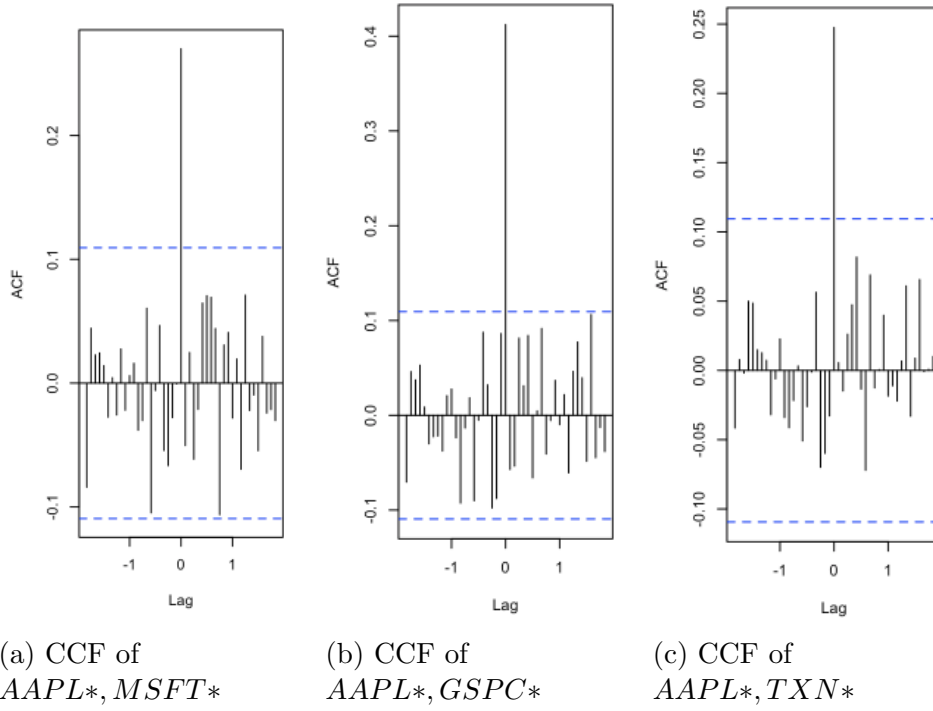


Figure 14: Cross-correlograms

where  $|\sum_{\hat{u}\hat{u}}(p)|$  is the determinant of the variance-covariance matrix of the estimated residuals,  $k$  is the number of variables, with  $T$  observations, and  $p$  is the order of the lag<sup>64</sup>.

Now, we use the `ar()` function in `library(vars)` to fit the appropriate  $VAR(p)$  model to this data with AIC as a comparison metric. We run the code which tells us that the most appropriate model is  $VAR(1)$ ; this agrees with the cross correlation plots seen above, which so not indicate any statistically significant cross correlations.

Now, we use the `VAR()` function in R to fit the  $VAR(1)$  model on the

---

<sup>64</sup>Kirchgassner, 135.

data. This gives us the estimates of the coefficients and their standard errors. We do not print out the output, but we generate the ACFs of the VAR(1) model's residuals in Figure 15.

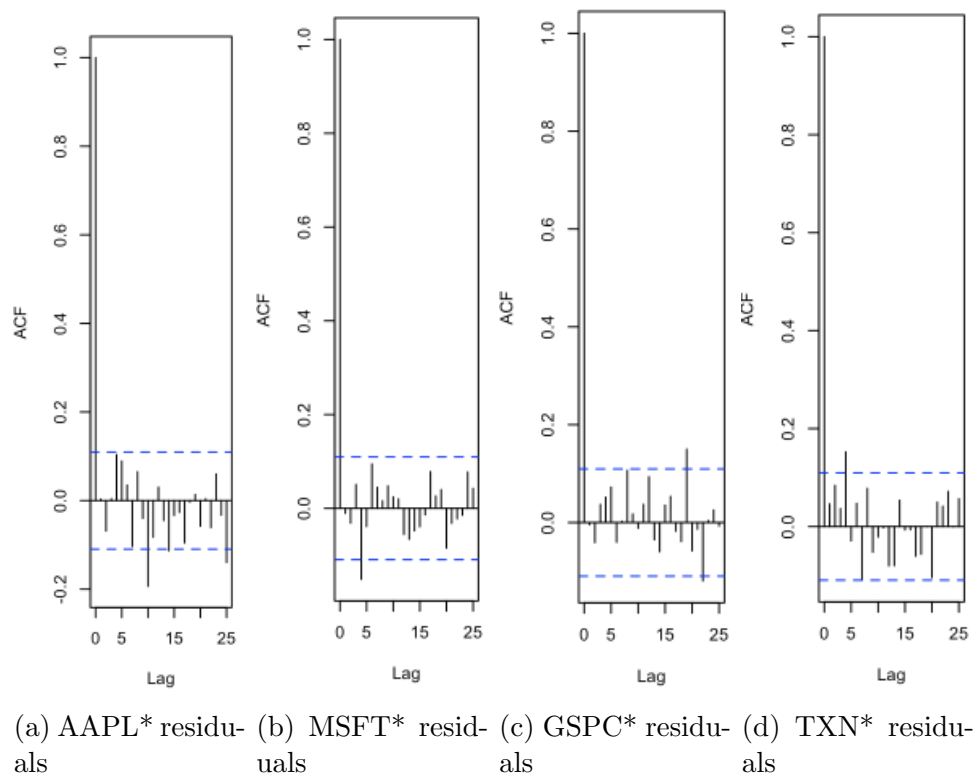


Figure 15: ACF of Residuals

In these ACFs in Figure 15, we see a few statistically significant autocorrelations, but we expect roughly 5% of autocorrelations to be randomly significant in a white noise model. The ACF in Figure 15 does not have statistically significant autocorrelations and we are satisfied with the model. If we saw strongly statistically significant autocorrelations with the independent variables and model residuals at the immediate lags i.e. ( $k = 1, 2$ ) then that would indicate that we might do better by fitting a VAR(p) model with a

larger value for p. We look at the CCFs of the model's residuals and each of the time series in Figure 16.

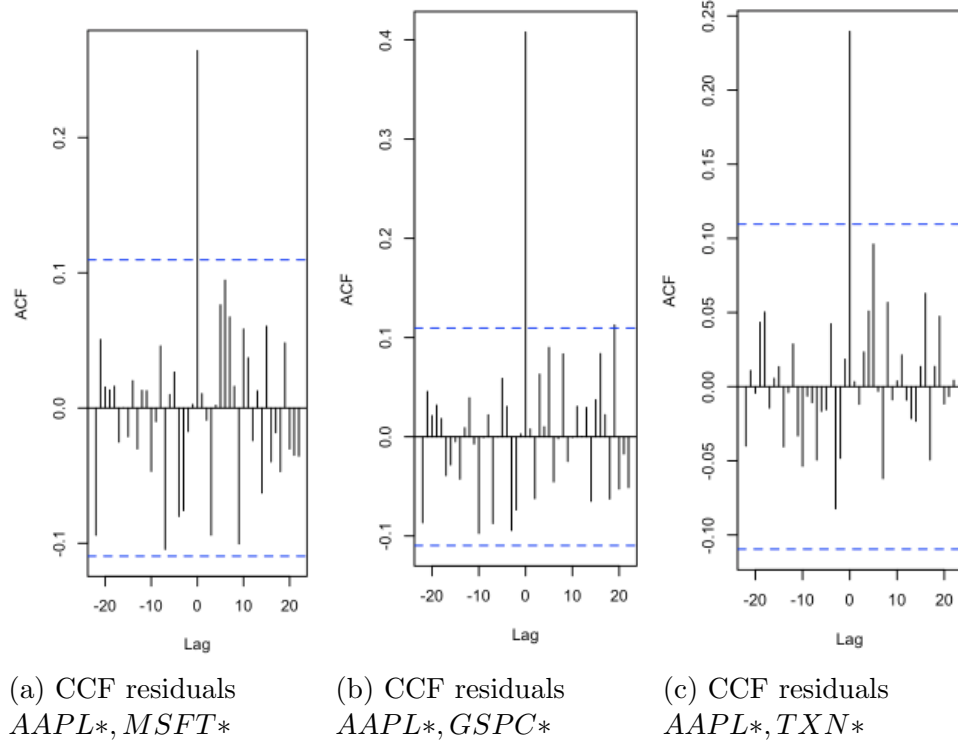


Figure 16: Cross-correlogram Residuals

In Figure 16, we do not see any statistically significant cross-correlations, which makes us confident in the model. We conclude there is no need to refine the model further. Since we are happy with the model, we will state it in the appropriate form below:

$$\begin{aligned}
 \hat{AAPL}_t &= 0.00252 * * + 0.12053 * y_{t-1} - 0.03532 MSFT_{t-1} - \\
 &\quad 0.00733 GSPC_{t-1} + 0.02798 TXN_{t-1} \\
 \hat{MSFT}_t &= 0.00104 + 0.02937 y_{t-1} - 0.27146 * * MSFT_{t-1} + \\
 &\quad 0.00246 GSPC_{t-1} + 0.04490 TXN_{t-1}
 \end{aligned}$$

$$\begin{aligned}
\hat{GSPC}_t &= 0.02769 * +1.20513y_{t-1} - 2.08957MSFT_{t-1} - 0.01275GSPC_{t-1} + \\
&\quad 2.22779 * *TXN_{t-1} \\
\hat{TXN}_t &= 0.00153 + 0.01959y_{t-1} - 0.19735 * *MSFT_{t-1} - 0.00144GSPC_{t-1} + \\
&\quad -0.02499TXN_{t-1}
\end{aligned}$$

Here, a double asterisk represents statistical significance at the  $\alpha < 0.001$  threshold, and a single asterisk represents statistical significance at the  $\alpha < 0.05$  level. The VAR model also gives us the following equations for our other variables: The relatively low number of statistically significant coefficients in the system of four equations above confirms with our earlier findings - that these time series do not profoundly impact the others.

## 6.5 Impulse Response Functions

From a VAR model, we can plot the Impulse Response Functions (IRF) which show the dynamics of a system when we simulate a shock to one variable, in this case increasing its value by one standard deviation. The plot shows the dynamic effect this has on itself and the other variables; the line shows the value and when it returns to 0 this means the time series has returned to equilibrium. First, we look at the IRF from shocking  $Y_t$ , the time series we want to forecast. We inspect the shock for 1 year ahead in Figure 17. Note that for the impulse response function labels, we are inspecting the differenced series, and use the notation  $a.d1 = AAPL$  differenced once,  $g.d1 = GSPC$  differenced once,  $m.d1 = MSFT$  differenced once, and  $t.d1 = TXN$  differenced once.

We see in Figure 17 that all of time series return to equilibrium within 2 months of the shock. This is not surprising because the above analysis

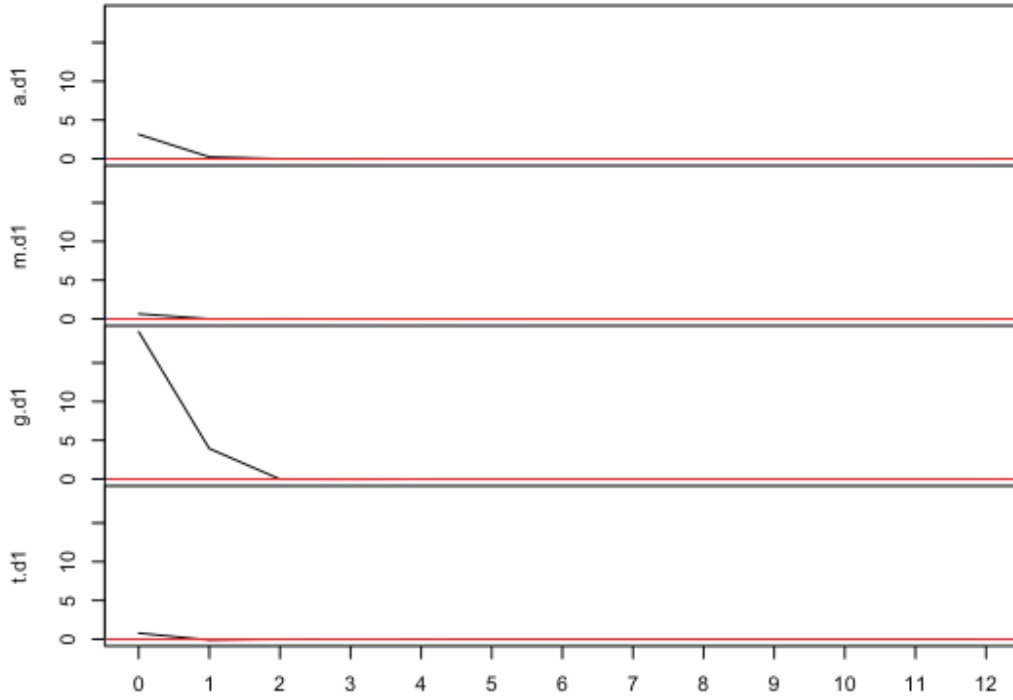


Figure 17: IRF - Shock to AAPL

indicated that the variables are not highly cross-correlation. Now, we consider the impact of a shock to MSFT on the other time series in Figure 18

Figure 18 shows us that shocking MSFT does not have profound impact on AAPL, MSFT, or TXN. GSPC, on the other hand, experiences an immediate increase, but then returns to equilibrium by month 3. Figure 19 shows the impact of a shock to GSPC.

Figure 19 shows that shocking GSPC has negligible impact on AAPL, MSFT, and TXN. Shocking GSPC does increase GSPC itself, but this bump quickly returns to equilibrium. The impacts of shocking TXN are seen in Figure 20.

Figure 20 shows that shocking TXN has negligible impact on AAPL and

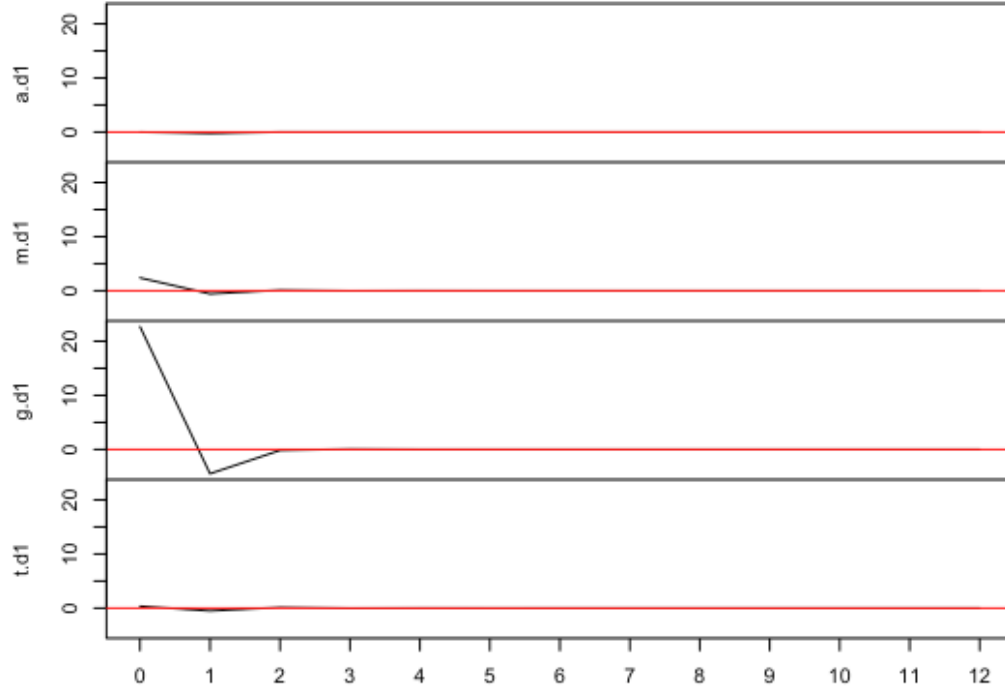


Figure 18: IRF - Shock to MSFT

MSFT. GSPC starts at its same level, but increases in the first month post-shock, and then decreases to equilibrium in the second month post-shock. TXN itself sees a slight bump in the month of the shock, and returns to equilibrium in the first month post-shock.

All of these impulse response functions show all the time series returning to equilibrium within 3 months. We see a few cases where a series oscillates above and below its equilibrium value, but none of the deviations are very dramatic. It is noteworthy that when we shock the 3 independent time series,  $Y_t$  barely changes at all. It would be interesting to see if this characteristic of these impulse response functions is unique to different frequencies of data - it is possible there is more intricate response behavior experience in daily, or

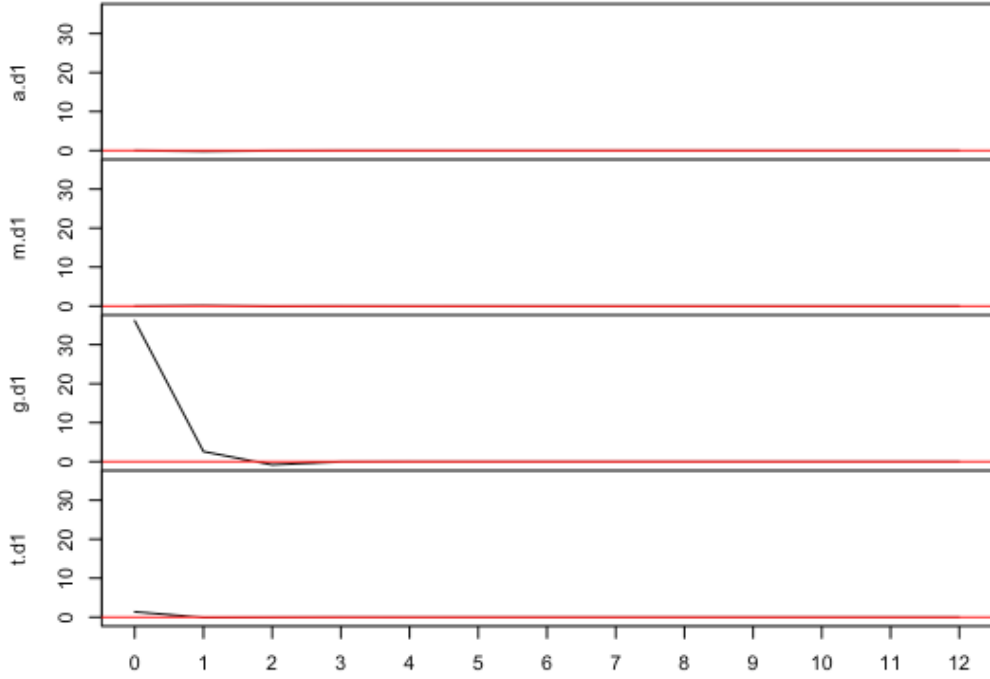


Figure 19: IRF - Shock to GSPC

hourly data, but that will be the focus of future research.

## 6.6 Forecasting with our VAR model

Now, we forecast our fitted VAR(1) model 12 months ahead and undo the differencing transformation. We record the predicted values in the original units, compute the RMSE for this 12 month forecast and record the data in Table 3. We visualize the forecast (red) in context of the original time series (black) in Figure 32.

We see from the RMSE in Table 3 at the end of the paper that our VAR model out performs our earlier ARIMA model. We see in Figure 21 that the VAR model captures some of the trend in the test set, and that is what

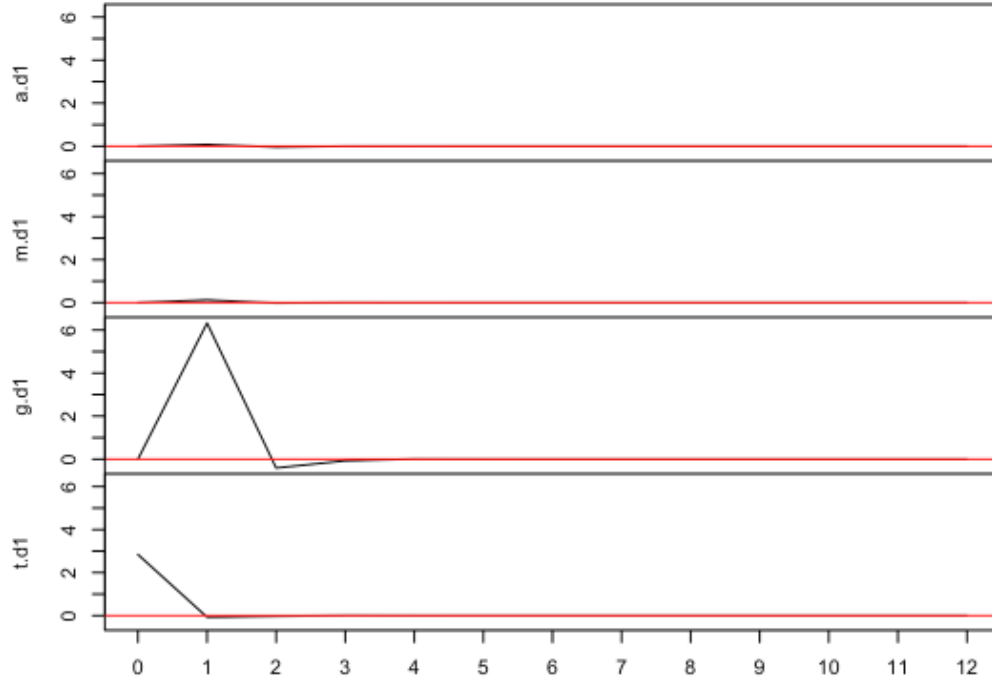


Figure 20: IRF - Shock to TXN

improves the overall performance. Additionally, the VAR model performs fairly well in the short term, but we see that the predicted values get less accurate past 6 months.

## 7 Time Series Regression

We fit a linear regression model with to time series,  $Y_t$  as the dependent variable and the other three time series as the independent variables. We fit this model using generalized least squares fitting the correlation in the residuals with an autoregressive moving average model, a well-documented approach for time series regression.



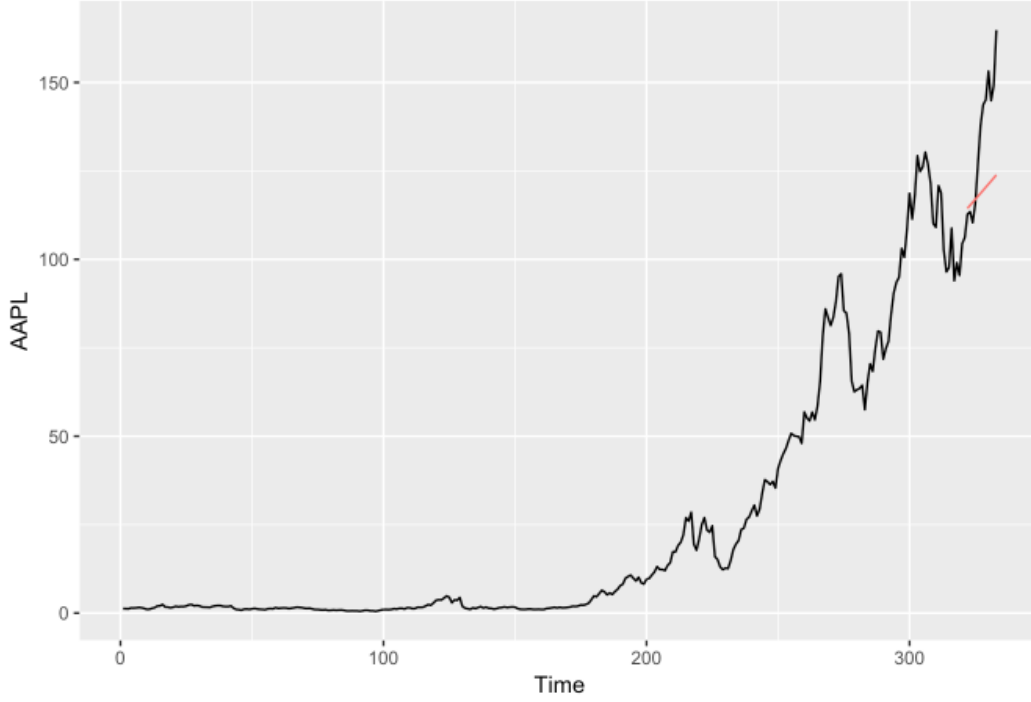


Figure 21: VAR(1) Forecast

Consider the regression model,

$$y_t = \sum_{j=1}^r \beta_j z_{tj} + e_t,$$

where  $e_t$  is a time series with a covariance function  $\gamma_e(s, t)$ . Ordinary least squares is predicated on  $e_t$  being a realization of a Gaussian white noise process, but that is often not the case in time series regression <sup>65</sup>. We use weighted least squares, which is most easily understood in vector notation; let

$$\vec{y} = \vec{Z}\vec{\beta} + \vec{e},$$

where  $\vec{y}$  and  $\vec{e}$  are  $n \times 1$  vectors,  $\vec{\beta}$  is a  $r \times 1$  vector and  $\vec{Z}$  is a  $n \times r$  matrix of input

---

<sup>65</sup>Shumway and Stoffer, 145.

vectors, in this case our contemporaneous time series<sup>66</sup>. Now, we let  $\Gamma$  be our autocovariance matrix of the residuals between two time series, such that,

$$\Gamma = \gamma_e(s, t)$$

and it follows that

$$\Gamma^{-1/2}\vec{y} = \Gamma^{-1/2}\vec{Z}\vec{\beta} + \Gamma^{-1/2}\vec{e};$$

it also follows that  $\Gamma^{-1/2}\vec{e}$  is the a set of uncorrelated random terms<sup>67</sup>. If we define

$$\vec{Z}_* = \Gamma^{-1/2}\vec{Z},$$

then we can conceptualize weighted estimate of  $\vec{\beta}$  as

$$\vec{\beta}_w = (\vec{Z}_*'\vec{Z}_*)^{-1}\vec{Z}_*'\vec{y}_* = (\vec{Z}'\vec{\Gamma}\vec{Z})^{-1}\vec{Z}'\vec{\Gamma}\vec{y} \text{ }^{68}.$$

This is essentially a method for refining a least squares regression estimate by accounting for autocorrelated residual series.

Now that we have established the basics of weighted least squares regression, we state the process for fitting a Linear Regression with correction for correlated error series model in practice. There are four main steps in this process, first, we run a standard regression on  $y_t$  using our predictor variables and keep the error series,  $\hat{e}_t$ . Second, we use the ACF and the PACF of the residuals to identify an ARMA model to this error series. Third, we use the R

---

<sup>66</sup>Shumway and Stoffer, 145.

<sup>67</sup>Shumway and Stoffer, 145

<sup>68</sup>Shumway and Stoffer, 145.

function *gls()* with the assumption that ARMA residuals fit in the regression model. Fourth, and finally, we inspect the residuals of this new model - if the ACF of the residual error series does not have statistically significant autocorrelations, we are satisfied with the resultant model, if not, we model further <sup>69</sup>. We use the *gls()* and *corARMA()* functions in R to implement this methodology, which correspond to generalized least squares, and correlated errors ARMA().

## 7.1 Linear Regression Model - GLS and CORARMA

Recall that we had to difference  $Y_t$  to make it stationary, but we did not have to pre-transform  $Y_t$  to make its variance stationary. It is also worth explicitly stating that none of the series involved in the regression fitting have been differenced. We fit a linear regression model on AAPL using our four independent time series (and nothing else) as the independent variables. We let R fit the model and we inspect the ACF of the residuals in Figure 22.

In Figure 22, the string of strongly statistically significant autocorrelations past lag 25 indicates a strong trend in the residuals.

## 7.2 Linear Regression Model on the Differenced Series

All of the regression models we fit above exhibit the same shortcomings - the residuals are highly trended and resistant to reduction with a *corARMA* model. When we try to alleviate the residuals trend by adding high order

---

<sup>69</sup>Shumway and Stoffer, 146.

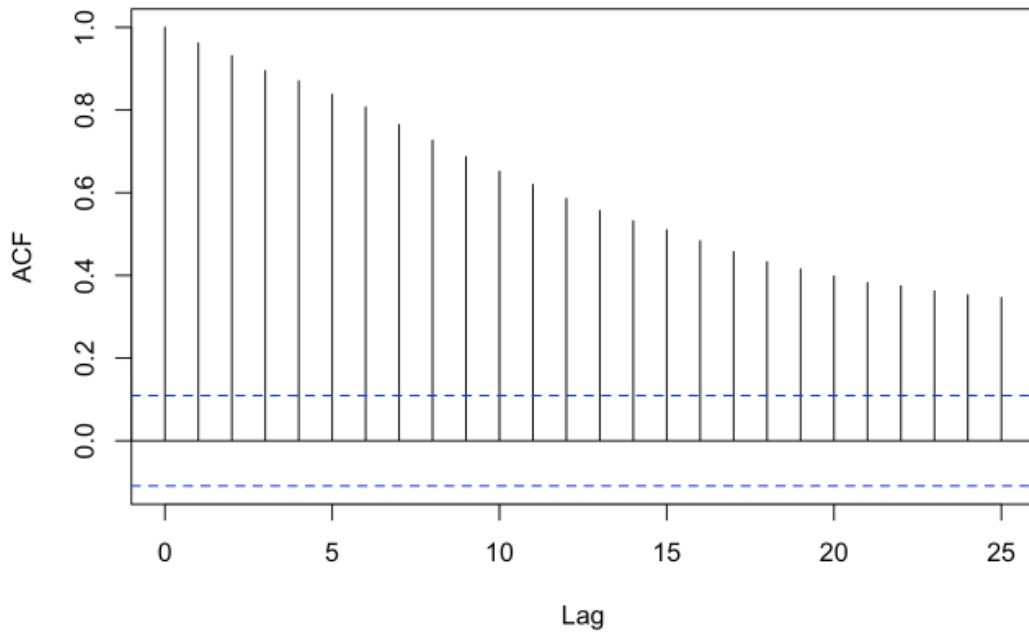


Figure 22: ACF of Linear Regression Model Residuals

dummy variables for time, we see that this was unsuccessful at de-trending the residuals. Another tactic is to de-trend the data before fitting a model, to see if that produces a model with stationary residuals. To do this, we fit a linear regression model on the regularly differenced time series of AAPL, using our other regularly-differenced time series as the independent variables. The independent variables will be our 3 regularly differenced time series. We look at the ACF of the residuals from our fitted regression model in Figure 23.

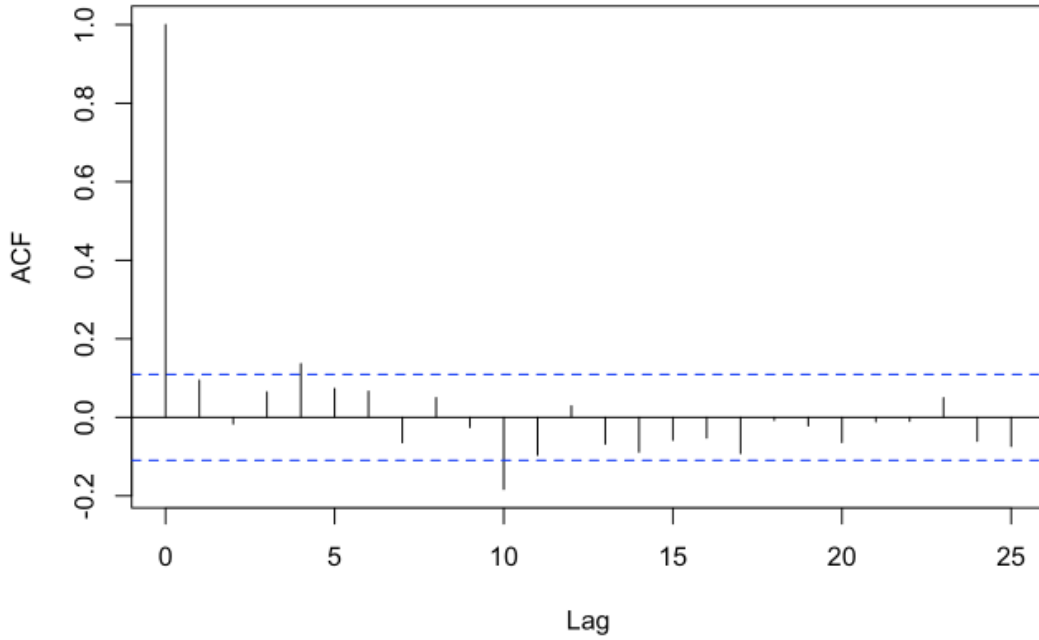


Figure 23: Residuals ACF - Regression on Differenced Series

The residuals ACF in Figure 23 has significant autocorrelations at lags 4 and 10, but these autocorrelations are small and negligible. Accordingly, this is the best regression model and we forecast it. The model that we fit is:

$$\hat{Y}_t = Y_{t-1} + 0.18905 + 0.02315(GSPC_t - GSPC_{t-1}) + 0.06954(MSFT_t - MSFT_{t-1}) + 0.07219(TXN_t - TXN_{t-1})$$

### 7.3 Linear Regression Model - Forecasting with the Best Regression Model

To forecast *AAPL* with the regularly differenced regression model, we construct a data frame of test set of the differenced independent time series, then we plug these into our model, and then un-difference the output. The results

are listed in Table 3 in the Conclusions Section and a plot is in Figure 24 (forecasts in red, original time series in black)

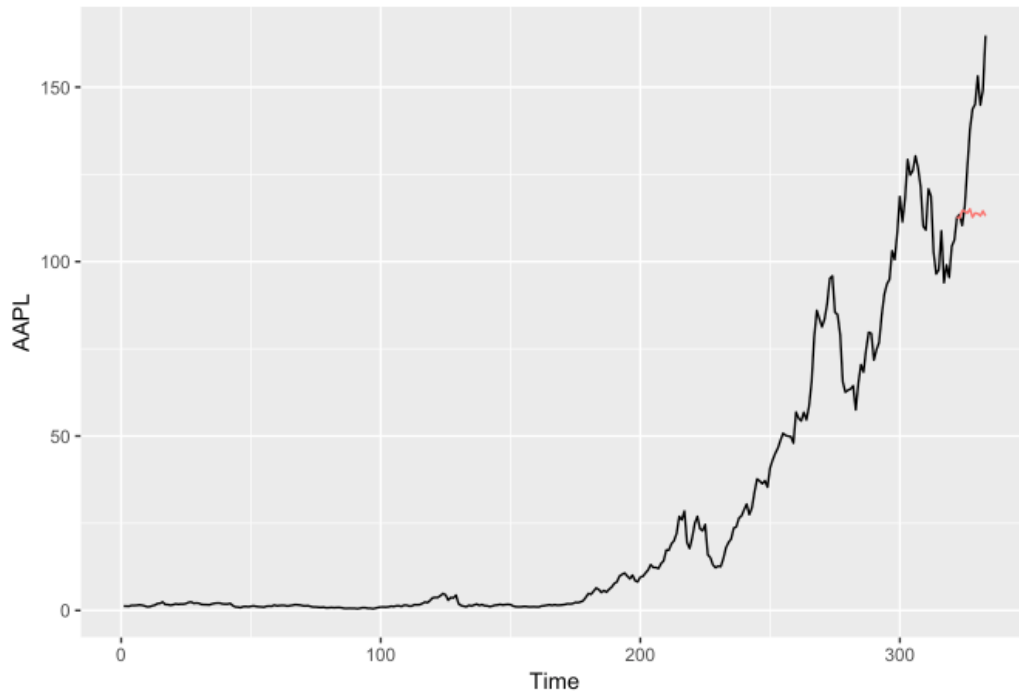


Figure 24: Linear Regression Forecast

Refer to Table 3, use RMSE as a comparison metric, and note that this regression model on the regularly differenced series slightly outperforms our ARIMA model and greatly underperforms our VAR model. We see in Figure 24 that this is because the linear regression model fails to identify the continued upward trend in the test set. We also see in Figure 24 that the linear regression model models some of the fluctuations in the time series, rather than predicting a constant value.

## 8 Conclusions

In Section 1, we stated our objectives for this thesis and our hypotheses on the relationships between the time series we analyze. In Section 2, we formally introduced the data and how to access it. In Section 3, we used multiplicative decomposition on the time series of Apple's stock price to inspect its trend, seasonal and random components on a descriptive level.

After introducing and getting familiar with the data, we moved on to univariate modeling. We also introduced the sample autocorrelation function in Section 3 - an important visualization and diagnostic throughout this thesis. In Section 4, we fit an ARIMA model on AAPL and forecasted AAPL with the resultant ARIMA(0,1,0) model. We also fit, and forecasted with an ARIMA+GARCH model in Section 4. In Section 5, we introduced, fit, and forecasted with a Holt-Winters exponential smoothing model.

After univariate models, we moved on to fitting and forecasting with multivariate models, using the contemporaneous time series of Microsoft's stock price, Texas Instruments stock price, and the value of the *S&P500* market index. In Section 6, we fit and forecasted with a VAR model. We also inspected the dynamics of this 4 variable system with impulse response functions and plots in Section 6. In Section 7, we performed linear regression with correction for autocorrelated errors. We also compute an average forecast from an unweighted average of all the aforementioned model's forecasts. We visualize the Averages Model Forecast in Figure 25 in red provide its forecasts and RMSE in Table 3.

We see in Figure 3 that the ARIMA model forecasts the last observed data

Period	Month	Observed	ARIMA	GARCH	Exp. Smooth	VAR	Regression	Average
M1	Oct 2016	112.71	112.71	120.74	113.34	114.39	112.79	114.79
M2	Nov 2016	113.46	112.71	129.56	113.97	115.27	112.44	116.79
M3	Dec 2016	110.37	112.71	139.24	114.60	116.11	114.57	119.45
M4	Jan 2017	115.80	112.71	149.86	115.23	116.69	114.47	121.85
MSE	-	-	1.3	193.1	1.57	3.37	1.7	11.21
RMSE	-	-	1.14	13.89	1.25	1.83	1.3	3.35

Table 2: Short Term (4 month) Forecast

point, the ARIMA+GARCH model dramatically over estimates its forecasts, the VAR, Time Series Regression, and Exponential Smoothing models capture some of the upward trend in the data, but still underestimate the test data. The average of all the forecasts is the best forecaster over the 12-month test period October 2016 - September 2017.

## 8.1 Best Short-Term Forecast

We are interested in the model that performs best for a short-term forecast. We determine that a 4-month period is appropriate for a short-term forecast, because this is financial data. Table 2 has the short-term forecasts and RMSE for our models.

We see in Table 2 that the model with the lowest RMSE is our ARIMA model. This is interesting because the ARIMA model's predictions are the last observed value. This means that when forecasting AAPL, even for the next 4 months, the last observed value is a safe and accurate forecast.



Period	Month	Observed	ARIMA	GARCH	Exp. Smooth	VAR	Regression	Average
M1	Oct 2016	112.71	112.71	120.74	113.34	114.39	112.79	114.79
M2	Nov 2016	113.46	112.71	129.56	113.97	115.27	112.44	116.79
M3	Dec 2016	110.37	112.71	139.24	114.60	116.11	114.57	119.45
M4	Jan 2017	115.80	112.71	149.86	115.23	116.69	114.47	121.85
M5	Feb 2017	127.03	112.71	161.53	115.86	117.81	113.91	124.37
M6	Mar 2017	137.89	112.71	174.35	116.49	118.67	115.06	127.45
M7	Apr 2017	143.71	112.71	188.42	117.12	119.53	112.73	130.10
M8	May 2017	145.10	112.71	203.87	117.75	120.39	113.91	133.73
M9	Jun 2017	153.17	112.71	220.83	118.38	121.25	113.75	137.38
M10	July 2017	144.88	112.71	239.46	119.01	122.12	113.13	141.29
M11	Aug 2017	149.1	112.71	259.92	119.64	122.99	114.49	145.95
M12	Sep 2017	164.80	112.71	282.37	120.26	123.86	113.07	150.45
MSE	-	-	797.86	4159.86	565.74	465.46	756.91	86.92
RMSE	-	-	28.35	64.50	23.79	21.57	27.51	11.96

Table 3: Long Term (12 month) Forecast

## 8.2 Best Long-Term Forecast

Table 3 provides the test set and the 12-month forecast for each of our models. We see in Table 3 that the Average Forecast has the best performance on the entire 12-month test set, using RMSE as a model comparison metric. The ranked order of model performance is Averages, Vector Autoregression, Exponential Smoothing, Linear Regression, ARIMA, and ARIMA+GARCH.

It is noteworthy that ARIMA+GARCH dramatically underperformed all other models, but helps influence the Average forecast's top performance. We see that The ARIMA+GARCH model vastly over estimates its forecast on the test set, but the other models we fit underestimate the test set data. Accordingly, the Averages Model finds a solid middle ground estimate and happens to accurately model the trend in the data.

The forecasts of all of our models, including the averages model, are plotted on the same graph in Appendix 1 Figures 27, 28, and 29(each plot has a

different time window). We show the last 50 observations and all the models' forecasts in Figure 26. Note that in Figure 26 ARIMA is red, ARIMA + GARCH is pink, VAR is orange, Linear Regression is blue, Exponential Smoothing is yellow, the Averages Model is green, and the observed values are in black.

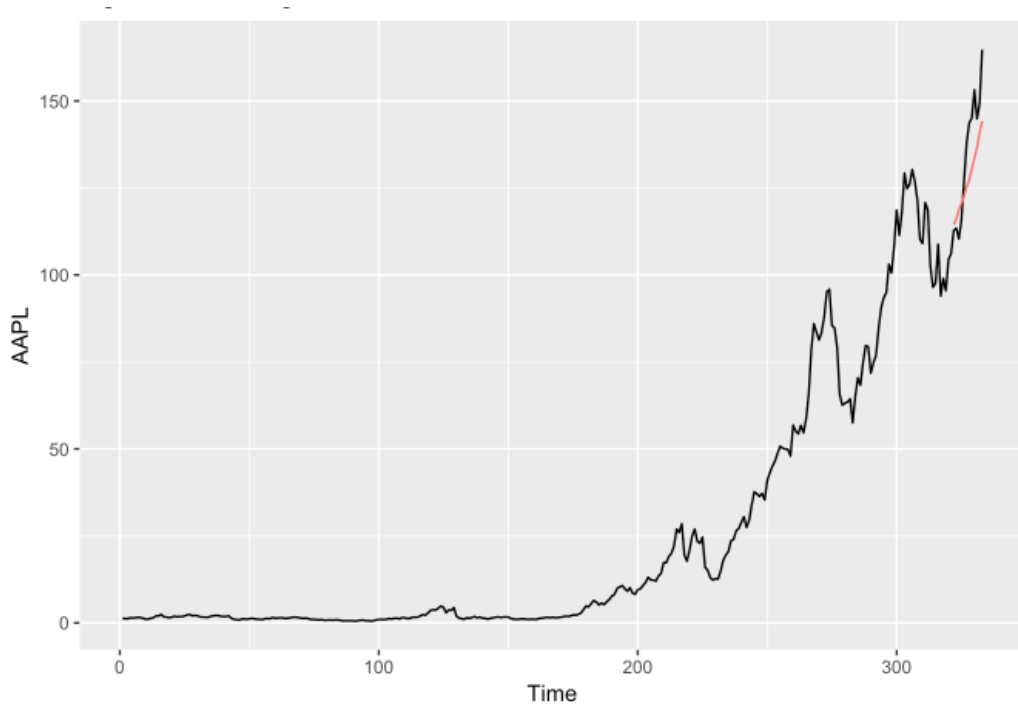


Figure 25: Model's Average Forecast

Figure 26 shows that the vast overestimations of the ARIMA+GARCH pull up the Averages Model and the other models are fairly conservative. If we did not have this over-estimating model, then the Averages model would not model the trend in the data as well and would underperform our second best long-term model, the  $VAR(1)$  model.

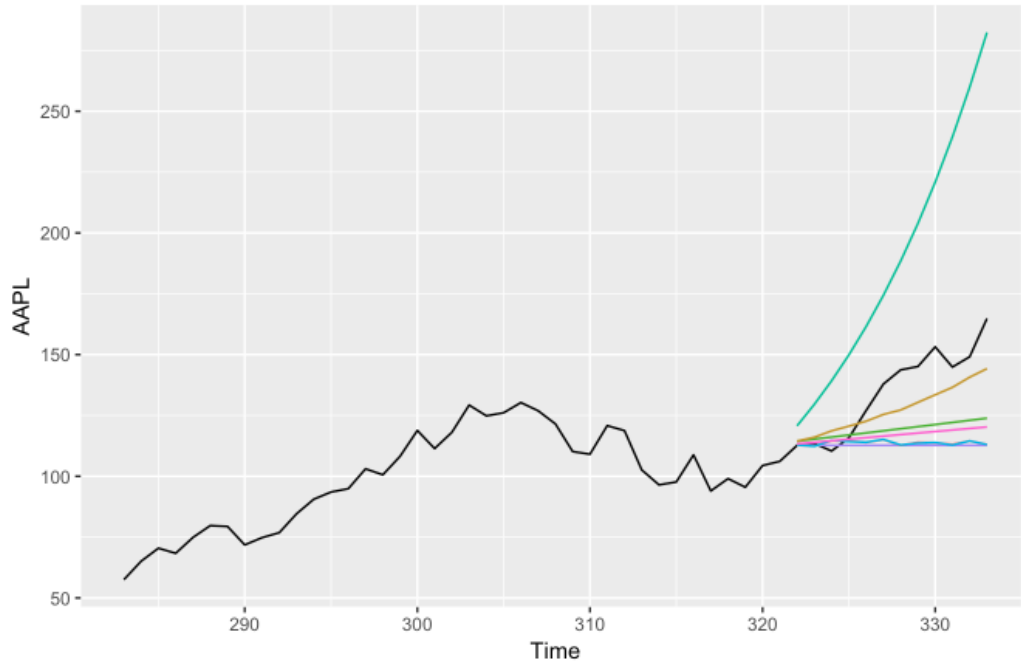


Figure 26: All Models Forecast (Last 50 Observations)  
 ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green

### 8.3 Final Conclusions

The first and second best models for a short term forecast were the ARIMA and Exponential Smoothing Models. Both these models only use past values of Apple stock to predict the future, this indicates that for a short term forecast, including covariates or other time series reduces performance.

One shortcoming of the ARIMA and Exponential smoothing models is that they fail to accurately forecast the trend in the data, which reduces long term forecast performance. The ARIMA+GARCH model on the other hand, mistakes volatility for trend and dramatically overestimates the trend in the data, making it a poor predictor in both the short and long term.

This thesis indicates that univariate models are better for short-term forecasting, but multivariate models are better for long-term forecasting. For short term forecasting, our best and worst models were both univariate, ARIMA and ARIMA+GARCH, respectively. Furthermore, for long-term forecasting, the multivariate models outperform all univariate models, except for the average forecast.

These results indicate that averaging conservative and aggressive estimates is a good strategy for 12-month forecasts. This indicates that for long-term forecasts, incorporating more data can also improve performance. It is noteworthy that while the Averages model was the best in the long-term, it is not as good as the more parsimonious models in the short term, because it is heavily influenced by the GARCH model's overestimation of volatility.

## 8.4 Further Research

One avenue for further research is to perform the same analysis with stock data of a different frequency, such as daily or hourly data. It would also be interesting to integrate a larger set of contemporaneous time series into our multivariate models - the VAR and Time Series Regression models, and see how this impacts performance.

Recalling Figure 1, we can see that our training period spanned two distinguishable periods - a fairly constant stock price from 1990 - 2005, and a period of exponential growth from 2005 - 2016. It would be interesting to see how model performance changes if the training period had been reduced to the period of growth.

Additionally, it would be beneficial to include a modern machine learning model into this analysis and compare the difference in performance. We saw in the Recent Research Section 1.1 that there is a plethora of neural network models that we could have been implemented.

One final avenue for further research is to dive deeper into ways of averaging the models we fit. We computed an un-weighted average of the models, but there are many ways we could have generated weighted average forecasts<sup>70</sup>.

---

<sup>70</sup>Gaynor, 578.

## Appendix A    Appendix I: Figures and Tables

	aapl.train	gspc.train	msft.train	txn.train
aapl.train	1.0000000	0.7579023	0.6120881	0.6168998
gspc.train	0.7579023	1.0000000	0.9395705	0.9107980
msft.train	0.6120881	0.9395705	1.0000000	0.9048573
txn.train	0.6168998	0.9107980	0.9048573	1.0000000

Table 4: Correlation Matrix of our 4 Time Series

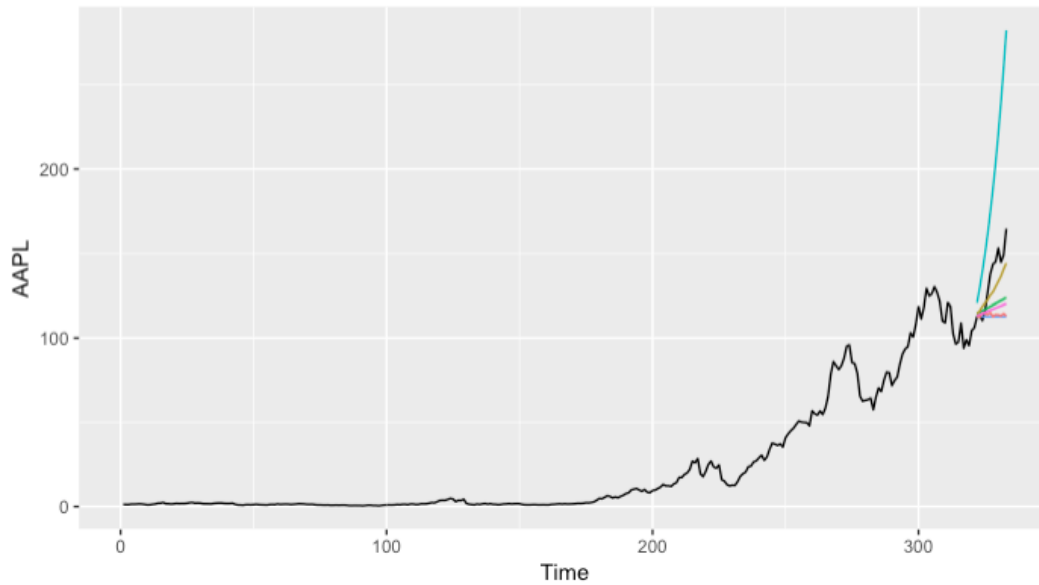


Figure 27: All Models Forecast

ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green

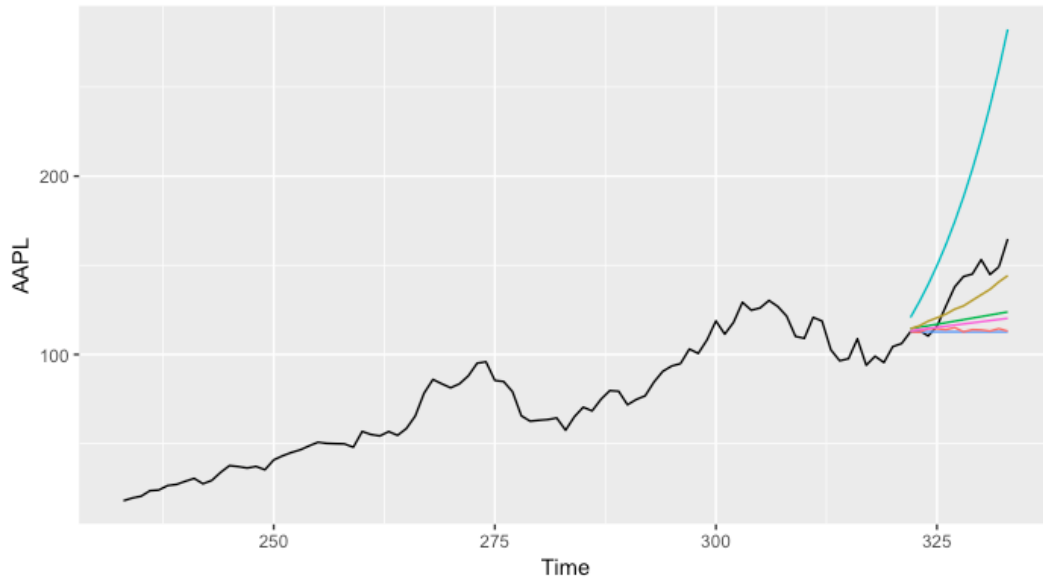


Figure 28: All Models Forecast (Last 100 Observations)  
 ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green

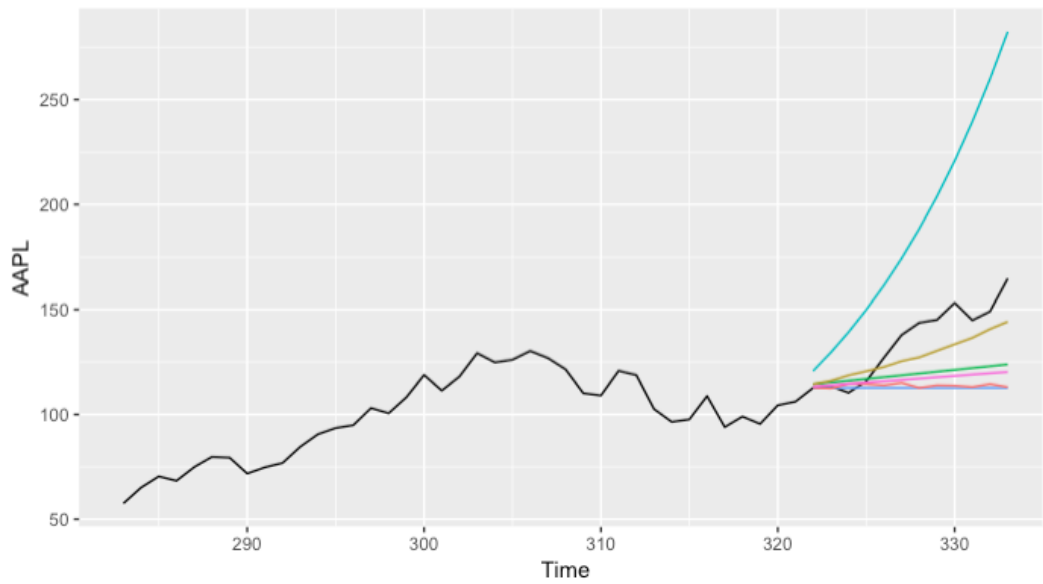


Figure 29: All Models Forecast (Last 50 Observations)  
 ARIMA = red, ARIMA + GARCH = pink, VAR = orange, Linear Regression = blue, Exponential Smoothing = yellow, Averages = green

## Appendix B    Appendix II: R Code

```
library(DT)
library(readr)
library(reshape2)
library(Hmisc)
library(knitr)
library(car)
library(ggplot2)
library(forecast)
library(dplyr)
library(tseries)
library(xts)
setwd("/Users/Jordan/Desktop/UCLA/MAS/thesis/")
data <- read_csv("stats415project.csv")

aapl2.ts <- ts(data[,2], frequency = 12, start = c(1990,1))
gspe2.ts <- ts(data[,3], frequency = 12, start = c(1990,1))
msft2.ts <- ts(data[,4], frequency = 12, start = c(1990,1))
txn2.ts <- ts(data[,5], frequency = 12, start = c(1990,1))

aapl.train <- aapl2.ts[1:322] %>%
  ts(frequency = 12, start = c(1990,1))
msft.train <- msft2.ts[1:322] %>%
```



```

    ts(frequency = 12, start = c(1990,1))
gspc.train <- gspc2.ts[1:322] %>%
    ts(frequency = 12, start = c(1990,1))
txn.train <- txn2.ts[1:322] %>%
    ts(frequency = 12, start = c(1990,1))

# Describing the Data
plot.ts(aapl.train, type = "l",
        main = "Figure_1: AAPL_opening_price")
aapl.150 <- aapl2.ts[1:150] %>% ts()
plot.ts(aapl.150, type = "l",
        main = "AAPL_First_150_Obs")

plot.ts(gspc2.ts, type = "l",
        main = "Figure_2a: GSPC_opening_value")
plot.ts(msft2.ts, type = "l",
        main = "Figure_2b: MSFT_opening_price")
plot.ts(txn2.ts, type = "l",
        main = "Figure_2c: TXN_opening_price")

plot(decompose(aapl.train, type = "multi"))

boxplot(aapl.train~cycle(aapl.train), xlab = "Month",
        main = "AAPL_Seasonal_Boxplot")

```

```

aapl.random <- decompose(aapl.train ,
  type = "multi")$random
acf(aapl.random, na.action = na.omit,
main = "ACF_random_component")

# ARIMA Section
d1 <- diff(aapl.train, differences = 1)
acf(d1, na.action = na.exclude,
  main = "ACF_of_(1-B)Y_t")

model2 <- arima(aapl.train, order = c(0,1,0))

acf(model2$residuals[-1], main =
"ACF_of_the_Residuals_of_ARIMA(0,1,0)_on_Y_t")

pred <- predict(model2, n.ahead=12)
model2_predicted <- pred$pred
model2_pred_se <- pred$se
aapl.test.real <- aapl2.ts[322:333,1]

ggplot(data, aes(x=1:333)) +
  geom_line(aes(y=AAPL)) +
  geom_line(data=data.frame(model2_predicted),

```

```

aes(x=c(322:333), y=model2_predicted,
     color = "red")) +
  xlab("Time") + ggtitle("ARIMA_Forecast") +
  theme(legend.position="none")

Box.test(model2$residuals,
         type = "Ljung-Box")

# ARIMA + GARCH Section

acf((model2$residuals[-1])^2,
     main = "ARIMA(0,1,0) Residuals ^2")

library(fGarch)

garch.3 <- garch(model2$residuals, order = c(1,1),
                 trace = F)
AIC(garch.3)

garch.4 <- garch(model2$residuals, order = c(0,2),
                 trace = F)
AIC(garch.4)

d1 <- diff(aapl.train, differences = 1)
garch.fit <- garchFit(formula=~arma(0,0)+garch(1,1),

```

```

data = d1, trace = F)
garch.pred <- predict(garch.fit , n.ahead = 12,
trace = FALSE, mse = c("cond"))
garch.forecast <- ts(garch.pred[,1]+garch.pred[,2])

acf(garch.fit@residuals , main = "ACF_of_Residuals")
acf(garch.fit@residuals^2, main = "ACF_of_Residuals^2")

garch.fc.og <- c()
for(i in 1:12){
  garch.fc.og[i] <- aapl.train[322] +
    sum(garch.forecast[1:i])
}

ggplot(data, aes(x=1:333)) + geom_line(aes(y=AAPL)) +
  geom_line(data=data.frame(garch.fc.og),
    aes(x=c(322:333),
y=garch.fc.og, color = "red")) +
  xlab("Time") + ggtitle("ARIMA+GARCH_Forecast") +
  theme(legend.position="none")

# VAR Section
library(tseries)
adf.test(aapl.train)

```

```

adf.test(gspc.train)
adf.test(msft.train)
adf.test(txn.train)

a.d1 <- diff(aapl.train, differences = 1)
m.d1 <- diff(msft.train, differences = 1)
g.d1 <- diff(gspc.train, differences = 1)
t.d1 <- diff(txn.train, differences = 1)

adf.test(m.d1)
adf.test(a.d1)
adf.test(g.d1)
adf.test(t.d1)

library(vars)
var1 <- VAR(cbind(a.d1, m.d1, g.d1, t.d1), p = 1,
            type = "trend")

ccf(a.d1, m.d1, main = "AAPL*—MSFT*")
ccf(a.d1, g.d1, main = "AAPL*—GSPC*")
ccf(a.d1, t.d1, main = "AAPL*—TXN*")

j1 <- ar(ts.union(a.d1, m.d1, g.d1, t.d1),

```

```

aic= TRUE)

aic_p_vector <- j1$aic

acf(resid(var1)[,1], main = "ACF_res_AAPL")
acf(resid(var1)[,2], main = "ACF_res_MSFT")
acf(resid(var1)[,3], main = "ACF_res_GSPC")
acf(resid(var1)[,4], main = "ACF_res_TXN")

ccf(resid(var1)[,1], resid(var1)[,2],
    main = "CCF_res_AAPL-MSFT")
ccf(resid(var1)[,1], resid(var1)[,3],
    main = "CCF_res_AAPL-GSPC")
ccf(resid(var1)[,1], resid(var1)[,4],
    main = "CCF_res_AAPL-TXN")

irf1 <- irf(var1, impulse = "a.d1", response =
            c("a.d1", "m.d1", "g.d1", "t.d1"),
    boot = FALSE, n.ahead = 12)
plot(irf1, main = "IRF_shock_to_AAPL")

irf2 <- irf(var1, impulse = "m.d1", response =
            c("a.d1", "m.d1", "g.d1", "t.d1"),
    boot = FALSE, n.ahead = 12)

```

```

plot(irf2 , main = "IRF_shock_to_MSFT")

irf3 <- irf(var1 , impulse = "g.d1" , response =
           c("a.d1" , "m.d1" , "g.d1" , "t.d1") ,
           boot = FALSE, n.ahead = 12)
plot(irf3 , main = "IRF_shock_to_GSPC")

irf4 <- irf(var1 , impulse = "t.d1" , response =
           c("a.d1" , "m.d1" , "g.d1" , "t.d1") ,
           boot = FALSE, n.ahead = 12)
plot(irf4 , main = "IRF_shock_to_TXN")

library(tseries)

cointegration<-function(x,y)
{
  vals<-data.frame(x,y)
  beta<-coef(lm( vals[,2] ~ vals[,1]+0 ,
    data=vals))[1]
  (adf.test( vals[,2] - beta*vals[,1] ,
    alternative="stationary" ,
    k=0))$p.value
}

cointegration(a.d1, m.d1)

```

```

cointegration(a.d1, g.d1)
cointegration(a.d1, t.d1)
cointegration(g.d1, m.d1)
cointegration(t.d1, m.d1)
cointegration(g.d1, g.d1)

```

```

cointegration(aapl.train, msft.train)
cointegration(aapl.train, gspc.train)
cointegration(aapl.train, txn.train)
cointegration(gspc.train, msft.train)
cointegration(txn.train, msft.train)
cointegration(txn.train, gspc.train)

```

```

p1 <- predict(var1, n.ahead = 12)
var1.fc <- p1$fcst$a.d1[,1]
var1.fc.og <- c()
for(i in 1:12){
  var1.fc.og[i] <- aapl.train[322] + sum(var1.fc[1:i])
}

```

```

ggplot(data, aes(x=1:333)) +
  geom_line(aes(y=AAPL)) +
  geom_line(data=data.frame(var1.fc.og),
    aes(x=c(322:333),

```



```

y=var1.fc.og, color = "red")) +
  xlab("Time") + ggtitle("VAR(1) Forecast") +
  theme(legend.position="none")

```

*# Regression Section*

```

mod.reg1 <- lm(aapl.train ~ gspc.train +
  msft.train + txn.train)

```

```

acf(resid(mod.reg1),
  main = "ACF of Linear Regression
  Model Residuals")

```

```

pacf(mod.reg1$residuals,
  main = "PACF of Linear Regression
  Model Residuals")

```

```

library(nlme)

```

```

mod.reg1.arma <- arma(resid(mod.reg1),
  order = c(1,0), include.intercept = FALSE)
j1 <- gls(aapl.train ~ gspc.train + msft.train +
  txn.train, correlation =
  corARMA(c(coef(mod.reg1.arma)),
  p=1))

```

```
acf(resid(j1 , type = "normalized"),
    main = "ACF of Residuals of GLS
    \\\( All Vars , corARMA(1,0))")
```

```
mod.reg.g <- lm(aapl.train ~ gspc.train )
mod.reg.m <- lm(aapl.train ~ msft.train )
mod.reg.t <- lm(aapl.train ~ txn.train )
```

```
mod.reg.m.arma <- arma(resid(mod.reg.m),
    order = c(1,0),
    include.intercept = FALSE)
gls.m <- gls(aapl.train ~ msft.train ,
    correlation = corARMA(c(coef
    (mod.reg.m.arma)),p=1))
```

```
mod.reg.g.arma <- arma(resid(mod.reg.g),
    order = c(1,0), include.intercept = FALSE)
gls.g <- gls(aapl.train ~ gspc.train ,
    correlation = corARMA(c(coef
    (mod.reg.g.arma)),p=1))
```

```
mod.reg.t.arma <- arma(resid(mod.reg.t),
    order = c(1,0), include.intercept = FALSE)
```

```

gls.t <- gls(aapl.train ~ txn.train,
             correlation = corARMA(c(coef
                                     (mod.reg.t.arma)), p=1))

```

```

par(mfrow = c(1,3))
acf(resid(gls.m, type = "normalized"),
    main = "ACF_of_Residuals_of_GLS
          (~MSFT, _corARMA(1,0))")
acf(resid(gls.g, type = "normalized"),
    main = "ACF_of_Residuals_of_GLS
          (~GSPC, _corARMA(1,0))")
acf(resid(gls.t, type = "normalized"),
    main = "ACF_of_Residuals_of_GLS
          (~TXN, _corARMA(1,0))")

```

```

a.train.d1 <- diff(aapl.train, differences = 1)
g.train.d1 <- diff(gspc.train, differences = 1)
m.train.d1 <- diff(msft.train, differences = 1)
t.train.d1 <- diff(txn.train, differences = 1)

```

```

reg.diff <- lm(a.train.d1 ~ g.train.d1 +
              m.train.d1 + t.train.d1)
acf(resid(reg.diff),
    main = "Residuals_ACF_-_Regression")

```

```

onDifferencedSeries")

aapl.test.real <- aapl2.ts[322:333,1]
gspc.test.real <- gspc2.ts[322:333,1]
msft.test.real <- msft2.ts[322:333,1]
txn.test.real <- txn2.ts[322:333,1]

gspc.test.diff <- diff(gspc2.ts,
  differences = 1)[321:332,1]
msft.test.diff <- diff(msft2.ts,
  differences = 1)[321:332,1]
txn.test.diff <- diff(txn2.ts,
  differences = 1)[321:332,1]

reg.diff.pred.frame <- data.frame(gspc.test.diff,
  msft.test.diff, txn.test.diff)

reg.diff.pred.aapl <- c()
reg.diff.pred.aapl.og <- c()
for(i in 1:nrow(reg.diff.pred.frame)){
  reg.diff.pred.aapl[i] = 0.18905 +
    0.02315*reg.diff.pred.frame[i,1] +
    0.06954*reg.diff.pred.frame[i,2] +
    0.07219*reg.diff.pred.frame[i,2]
}

```

```

reg.diff.pred.aapl.og[i] = 112.709999 +
  reg.diff.pred.aapl[i]
}

ggplot(data, aes(x=1:333)) +
  geom_line(aes(y=AAPL)) +
  geom_line(data=data.frame(reg.diff.pred.aapl.og),
aes(x=c(322:333), y=reg.diff.pred.aapl.og,
  color = "red")) +
  xlab("Time") +
  ggtitle("Linear Regression Forecast") +
  theme(legend.position="none")

```

*#Exponential Smoothing Section*

```
hw1 <- HoltWinters(aapl.train, gamma = FALSE)
```

```
hw1$alpha
```

```
hw1$beta
```

```
hw1.pred <- predict(hw1, n.ahead = 12)
```

```
hw.table1 <- cbind( as.data.frame(aapl.test.real),
as.data.frame(hw1.pred)) %>%
```

```
as.data.frame() %>%
  dplyr::select("observed" = 1,
    "predicted" = 2)
```

```
ggplot(data, aes(x=1:333)) + geom_line(aes(y=AAPL)) +
  geom_line(data=data.frame(hw1.pred),
    + aes(x=c(322:333), y=hw1.pred,
    color = "red")) +
  xlab("Time") + ggtitle("Holt-Winters_Forecast") +
  +theme(legend.position="none")
```

*# Conclusions Section and Appendix I*

```
arima.pred <- c(rep(112.709, 12))
preds <- data.frame(arima.pred, garch.fc.og,
  var1.fc.og, reg.diff.pred.aapl.og, hw1.pred) %>%
  mutate(averages = rowMeans(.)) %>%
  mutate_if(is.numeric, funs(round(.,2)))
conc.tab <- data.frame(round(aapl.test.real,2),
  preds)
st <- data.frame(conc.tab[1:4,])
st.arima.mse <- sum((st$arima.pred -
  +st$round.aapl.test.real..2.)^2)/12
```

```

st.garch.mse <- sum((st$garch.fc.og -
                    +st$round.aapl.test.real..2.)^2)/12
st.var.mse <- sum((st$var1.fc.og -
                    +st$round.aapl.test.real..2.)^2)/12
st.reg.diff.mse <- sum((st$reg.diff.pred.aapl.og -
                        +st$round.aapl.test.real..2.)^2)/12
st.fit.mse <- sum((st$fit -
                  st$round.aapl.test.real..2.)^2)/12
st.averages.mse <- sum((st$averages -
                        +st$round.aapl.test.real..2.)^2)/12
st.mse <- c("-", round(st.arima.mse,2),
            round(st.garch.mse,2),
            round(st.var.mse,2), round(st.reg.diff.mse,2),
            round(st.fit.mse,2),
            round(st.averages.mse,2))
st.tab1 <- rbind(st, st.mse)
period.st <- c("M1", "M2", "M3", "M4", "MSE")
st.tab2 <- data.frame(period.st, st.tab1) %>%
  dplyr::select("period" = 1,
               "Observed" = 2, "ARIMA_Predicted" = 3,
               "ARIMA_GARCH_Predicted" = 4,
               "VAR(1)_Predicted" = 5, "Regression_Predicted" = 6,
               "Exponential_Smoothing_Predicted" = 7,
               "Averages" = 8)

```

```

arima.pred <- c(rep(112.709, 12))
preds <- data.frame(arima.pred, garch.fc.og,
  var1.fc.og, reg.diff.pred.aapl.og, hw1.pred) %>%
  mutate(averages = rowMeans(.)) %>%
  mutate_if(is.numeric, funs(round(.,2)))
conc.tab <- data.frame(round(aapl.test.real,2), preds)
averages.mse <- sum((((conc.tab$averages -
  conc.tab$round.aapl.test.real..2)^2)/12)
conc.mse.vec <- c("-", 797.86, 4159.86,
  465.46, 756.91, 565.74, round(averages.mse,2))
conc.rmse.vec <- c("-", round(797.86^.5,2),
  round(4159.86^.5,2), round(465.46^.5,2),
  round(756.91^.5,2),
  round(565.74^.5,2), round(143.07^.5,2))
conc.tab1 <- rbind(conc.tab, conc.mse.vec,
  conc.rmse.vec)
period <- c("M1", "M2", "M3", "M4", "M5",
  "M6", "M7", "M8", "M9", "M10", "M11", "M12",
  "MSE", "RMSE")
conc.tab2 <- data.frame(period, conc.tab1) %>%
  dplyr::select("period" = 1, "Observed" = 2,
    "ARIMA_Predicted" = 3,
    "ARIMA_GARCH_Predicted" = 4, "VAR(1)_Predicted" = 5,

```



"Regression\_Predicted" = 6,

"Exponential\_Smoothing\_Predicted" = 7, "Averages" = 8)

```
ggplot(data[283:333,], aes(x=283:333)) +  
  +geom_line(aes(y=AAPL)) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=arima.pred, color = "red")) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=garch.fc.og, color = "pink")) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=var1.fc.og, color = "orange")) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=reg.diff.pred.aapl.og, color = "blue")) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=fit, color = "yellow")) +  
  geom_line(data=conc.tab, aes(x=c(322:333),  
    y=averages, color = "green")) +  
  xlab("Time") + ggtitle("All_Models_Forecast  
  _Last_50_Observations") +  
  +theme(legend.position="none")  
  
cor(cbind(aapl.train, gsfc.train, msft.train, txn.train))
```

```

ggplot(data, aes(x=1:333)) + geom_line(aes(y=AAPL)) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=arima.pred, color = "red")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=garch.fc.og, color = "pink")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=var1.fc.og, color = "orange")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=reg.diff.pred.aapl.og, color = "blue")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=fit, color = "yellow")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=averages, color = "green")) +
  xlab("Time") + ggtitle("All_Models_Forecast") +
  +theme(legend.position="none")

```

```

ggplot(data[233:333,], aes(x=233:333)) +
  geom_line(aes(y=AAPL)) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=arima.pred, color = "red")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=garch.fc.og, color = "pink")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
    y=var1.fc.og, color = "orange")) +

```

```

geom_line(data=conc.tab, aes(x=c(322:333),
                             y=reg.diff.pred.aapl.og, color = "blue")) +
geom_line(data=conc.tab, aes(x=c(322:333),
                             y=fit, color = "yellow")) +
geom_line(data=conc.tab, aes(x=c(322:333),
                             y=averages, color = "green")) +
  xlab("Time") + ggtitle("All Models Forecast
  ... (Last 100 Observations)") +
  +theme(legend.position="none")

ggplot(data[283:333,], aes(x=283:333)) +
  geom_line(aes(y=AAPL)) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=arima.pred, color = "red")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=garch.fc.og, color = "pink")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=var1.fc.og, color = "orange")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=reg.diff.pred.aapl.og, color = "blue")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=fit, color = "yellow")) +
  geom_line(data=conc.tab, aes(x=c(322:333),
                              y=averages, color = "green")) +

```

```
  xlab("Time") + ggtitle("All Models Forecast  
~~~~(Last 50 Observations)") +  
  +theme(legend.position="none")
```

## References

- [1] N. Baba, M. Kozaki. "An intelligent forecasting system of stock price using neural networks". *Proceedings of the international joint conference on neural networks* (1992): pp. 371-377.
- [2] G.E.P. Box, G. Jenkins. *Time series analysis, forecasting and control*. Holden-Day, San Francisco, CA (1970)
- [3] Chan, Ngai H. *Time Series: Applications to Finance with R and S-Plus, Second Edition*. Hoboken, N.J: Wiley, 2010. Internet resource.
- [4] Cowpertwait, Paul S. P, and Andrew V. Metcalfe. *Introductory Time Series with R*. New York: Springer-Verlag, 2009. Internet resource.
- [5] Gaynor, Patricia E, and Rickey C. Kirkpatrick. *Introduction to Time-Series Modeling and Forecasting in Business and Economics*. New York: McGraw-Hill, 1993. Print.
- [6] Hadavandi, Esmail, et al. "Integration of Genetic Fuzzy Systems and Artificial Neural Networks for Stock Price Forecasting." *Knowledge-Based Systems* vol. 23. No. 8 (2010): pp. 800–808.
- [7] K. Kamijo, T. Tanigawa. "Stock price pattern recognition—a recurrent neural network approach". *Proceedings of the international joint conference on neural networks* (1990): pp. 211-215. Print.
- [8] Kirchgaässner, Gebhard, Jürgen Wolters, and Uwe Hassler. *Introduction to Modern Time Series Analysis*. Berlin: Springer, 2013. Internet resource.

- [9] Pai, Ping-Feng, and Chih-Sheng Lin. "A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting." *Omega* vol. 33. No. 6 (2005): pp. 497–505.
- [10] R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- [11] E.W. Saad, D.V. Prokhorov, D.C. Wunsch. "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks". *IEEE Transactions on Neural Networks* 9 (1998): pp. 1456-1470.
- [12] Shumway, Robert H, and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Cham: Switzerland, 2017. Internet resource.
- [13] Wu, Z.Q. and Harris, C.J. "Indirect Adaptive Neurofuzzy Estimation of Nonlinear Time Series" *Neural Network World* vol. 6. No 3 (1996): pp. 407-416.
- [14] Y. Yoon, G. Swalves. "Predicting stock price performance: a neural network approach". *Proceedings of the twenty-fourth annual Hawaii international conference on system sciences* (1991): pp. 156-162. Print.