Forecasting APPL Stock Returns Using Time Series in R

Team members:

Kanika Puri
Kao-Ying Chen
Harshita Shyam

Abstract

Forecasting stock returns is a popular and widely used practice both in the academic and financial worlds. Time series analysis focuses on analyzing data set to study the characteristics of the data and extract meaningful statistics in order to predict future values of the series. There are four methods in time series analysis namely, time series decomposition, forecasting, clustering and classification. In this paper, we forecast Apple stock returns using historical stock prices data from Yahoo Finance. We began by comparing the original non-stationary stock returns and the differenced (stationary) stock returns, which is a common practice in financial time series. Then we built the autoregressive moving average (ARMA) and the autoregressive integrated moving average (ARIMA) model to forecast the stock returns. We then compared the ARIMA model with the original stock returns that was obtained from Yahoo Finance to show the difference between the non-stationary and stationary series. In the last part of our analysis, we compared the ARIMA model and the ARMA model to forecast returns.

Our analysis indicated that the original data was non-stationary and the Augmented Dickey-Fuller Test confirmed this. As a result, we converted the data to stationary series by first differencing the time series data and then using log transformations on the original series and the differenced series, which is the common technique for analysis of financial data. The output graphs indicated that the differenced log price of AAPL had constant variance and it did not change as the level of the original series changed indicating that the series was stationary. Then we built the ARMA and ARIMA model. The ARMA model was not suitable for long term predictions due to the assumption of mean reversion of the series and it had a poor performance/accuracy. The ARIMA model was used to forecast predictions providing an estimate of the expected next time series point(s) and showed the prediction graphically. The 'forecast' function provided an 80% and 95% confidence intervals as well as a point estimate and it had a better performance than the ARMA model with a lower RMSE value.

In building the ARIMA model, we first calculated the autocorrelation and partial autocorrelation to obtain the p (autoregressive parameter) and q (moving average parameters). This was done by using the 'acf' and 'pacf' functions as well as

using d (number of differencing) as the middle parameter. Once this was obtained, we built the ARIMA model by calling the 'arima' function. The output graph clearly indicated that the series was stationary and the predicted returns was accurate given the linear nature of the graph outputs as well as the lack of randomness. When the graph from the ARIMA model was compared to the original pre-processed data, it clearly showed that making the series stationary ensures that the predicted model output provided a much better and reliable forecast than the non-stationary pre-processed data. Please note that we don't go into depth about building the ARMA model in this paper since it requires just building a testing and training data set and then analyzing the data to forecast it, which is discussed in brief later on.

Background

Forecasting stock returns is a popular and widely used practice. We decided to analyze and forecast Apple Inc. (AAPL) because this stock is popular and there is a large amount of information available online that will aid us in the process of analyzing the data and ensuring that our model predictions are accurate.

As sated earlier, there are four methods in time series analysis namely, time series decomposition, forecasting, clustering and classification. In *Time Series Decomposition*, the time series is decomposed into trend, seasonal, cyclical and irregular components. In *Time Series Clustering*, time series data is partitioned into groups based on similarity or distance, so that time series in the same cluster are similar to each other. In *Time Series Classification,* classification models are built based on labeled time series and then the model is used to predict the label of unlabeled time series. In *Time Series Forecasting*, historical data is used to predict/ forecast future events. We used this technique to forecast Apple Stock returns, which will be discussed in further detail below. We used the concept of differencing and logging the series to make the data stationary and then compared and contrasted the results with the original pre-processed data followed by autocorrelation and partial autocorrelation of the post-processed (differenced and logged) series which was then used to fit the autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models to predict the future returns of the AAPL Stock Price and draw comparisons between the two models.

We chose the ARIMA model over the ARMA model since it was an appropriate choice for long term forecasting, as it doesn't assume mean reversion. It also had better performance as we observed in the model outputs. However, in R, this model when used directly with the data tends to ignore the mean and differencing therefore, we first performed the differencing and logging and then fit the post-processed data to the ARMA and ARIMA models which ensured that the end output was accurate. Time domain method is established and implemented by observing the autocorrelation of the time series. Therefore, autocorrelation and partial autocorrelation are the core of ARIMA model. Akaike Information Criterion

(AIC) provides yet another way to identify ARIMA model according to autocorrelation and partial autocorrelation graph of the series. The parameters of ARIMA consist of three components: p (autoregressive parameter), d (number of differencing), and q (moving average parameters). Using these parameters, we were able to accurately predict the future returns of AAPL stock price.

Data Mining Model and Results

About the Data: Daily stock prices of AAPL from January 2nd, 2004 to December 31st, 2013 are extracted from the Yahoo Finance website. This data set contains the open, high, low, close and adjusted close prices of AAPL stock however to achieve consistency, we used only the adjusted close prices as they reflected the general measure of stock prices of AAPL over the past ten years.

Our main model is the ARIMA model so we wont go into depth about the ARMA model. In order to build the ARIMA model, we first imported the daily historical stock price AAPL data directly from the Yahoo Finance website. We used only two fields from the data namely, the date (index as referred to in the dataset) and the close prices as a general measure of stock price of AAPL over the past ten years to achieve consistency. We then plotted a graph of the original daily close price, which helped us to visualize the data in a much better manner than just displaying the summary. We observed that the series was non-stationary and verified this by performing Augmented Dickey-Fuller Test. Given the stationary series, the ARIMA model fits perfectly as its built for non-stationary data. However, in order to make the data stationary to ensure accurate predictions, we performed differencing and logging which also converted the data to time series. This methodology's forecasting was accurate which was evident when compared to the original data. The next step in building the model was autocorrelation and partial autocorrelation of time series. These provide the input parameters: p and q that are required for the 'arima' function along with the differencing parameter, d that is obtained from the differencing and logging step in the conversion of non-stationary series to stationary series. In the final step, we used the logged time series and the input parameters for the 'arima' function to fit the ARIMA model in order to forecast AAPL's stock returns. Detailed explanation for each step along with the R code is provided below.

We began by installing the required packages and libraries for our analysis and forecasting of AAPL stock as follows:

```
# Installing required packages for time series forecasting #
install.packages("tseries")
install.packages("timeDate")
install.packages("Rcmdr")
install.packages("RcmdrPlugin.epack")
install.packages("tseries")
```

```
install.packages("zoo")
install.packages("quantmod")
install.packages("forecast")
library(FitARMA)
library(forecast)
library(tseries, quietly = T)
library(forecast, quietly = T)
```

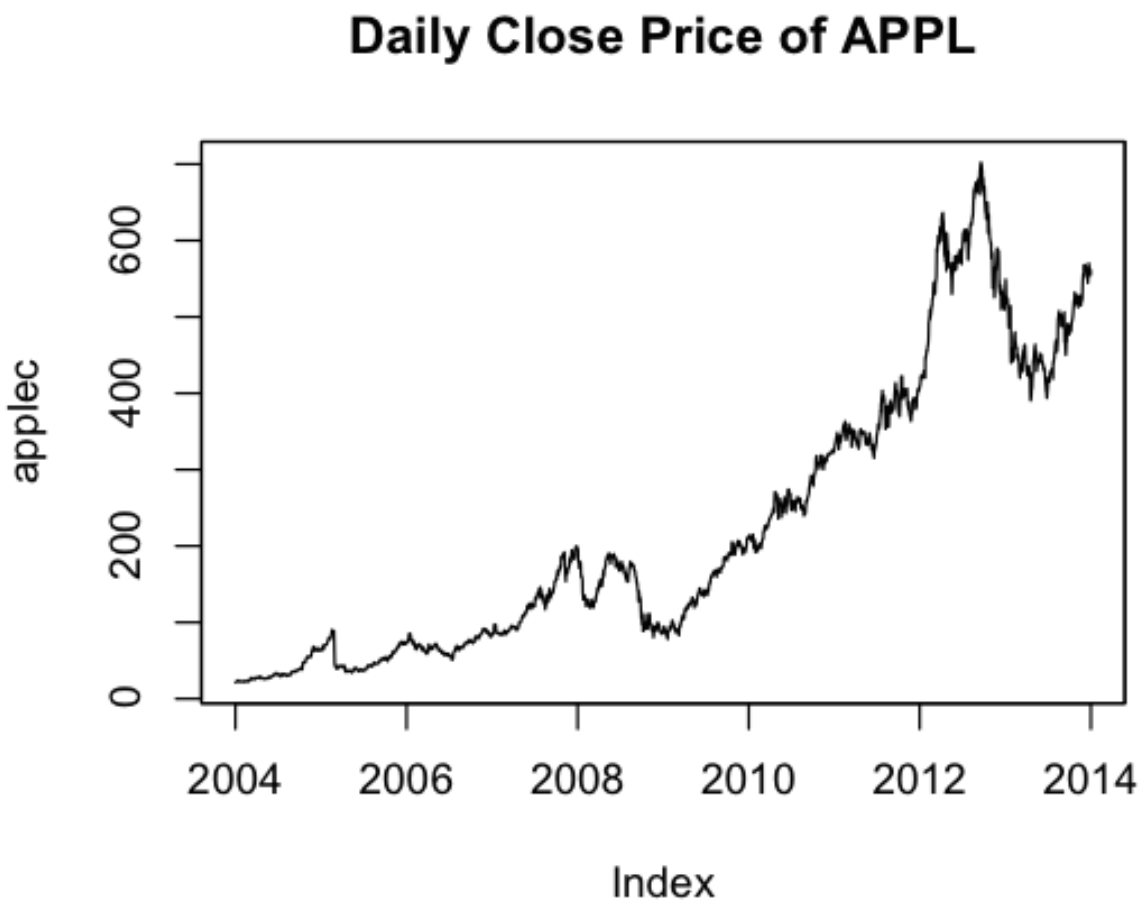
Next, we imported the data directly from Yahoo Finance using the command below:

```
# Downloads the AAPL historical price dataset from Yahoo Finance #
applec <- get.hist.quote(instrument = "aapl", start = "2004-01-01", end = "2014-01-01", quote = "Close")
```

The variable applec refers to the AAPL close price. The function 'get.hist.quote' is used to retrieve the daily historical data. The data imported contains the daily closing price of AAPL from January 2nd, 2004 to December 31st, 2013.

We then took a look at the data and plotted the original daily close price. The graph helped us visualize the data better, which shows the daily close price of AAPL.

```
# Summary of AAPL data #
summary(applec)

# Length of the AAPL data #
length(applec)

# Plot original daily close price #
plot(applec,type='l',main='Daily Close Price of APPL')
```



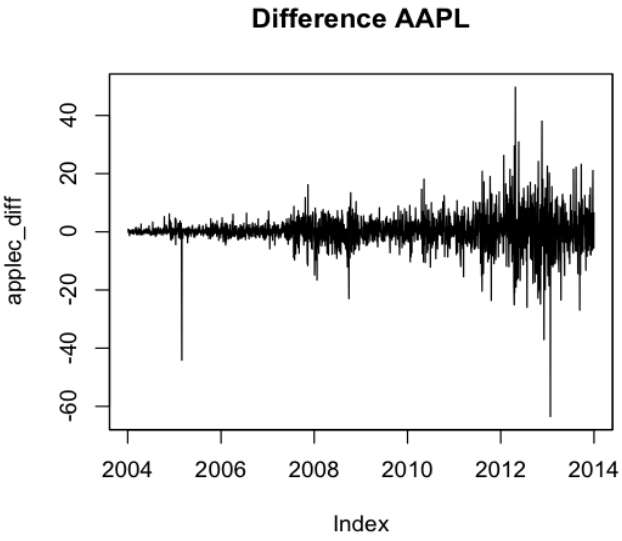
This plot shows that the close price of AAPL increases in general over the past ten years. We can also conclude that the variance of the stock price seems to increase slightly with time with the stock price showing an extreme change towards the end. Therefore, we can conclude that the plot shows non-stationary series and that we need to stabilize variance. In order to ensure that our visual analysis of the series is valid, we conducted the Augmented Dickey-Fuller Test.

```
# Augmented Dickey-Fuller Test to determine/confirm if the series is stationary or non-stationary #
adf.test(applec)
```

```
        Augmented Dickey-Fuller Test

data:  applec
Dickey-Fuller = -2.0204, Lag order = 13, p-value = 0.5697
alternative hypothesis: stationary
```

As seen above, the Augmented Dickey-Fuller Test confirms our analysis by clearly showing the result as "alternative hypothesis is stationary" implying that we need to make the series stationary. The reason why we make the series stationary is so the statistical properties remain the same for the future as they have in the past. This is achieved by first differencing the time series data which refers to converting non-stationary series to stationary series and then using log transformations on the original series and the differenced series which is the common technique that is used in the analysis of financial data. These steps are performed below:

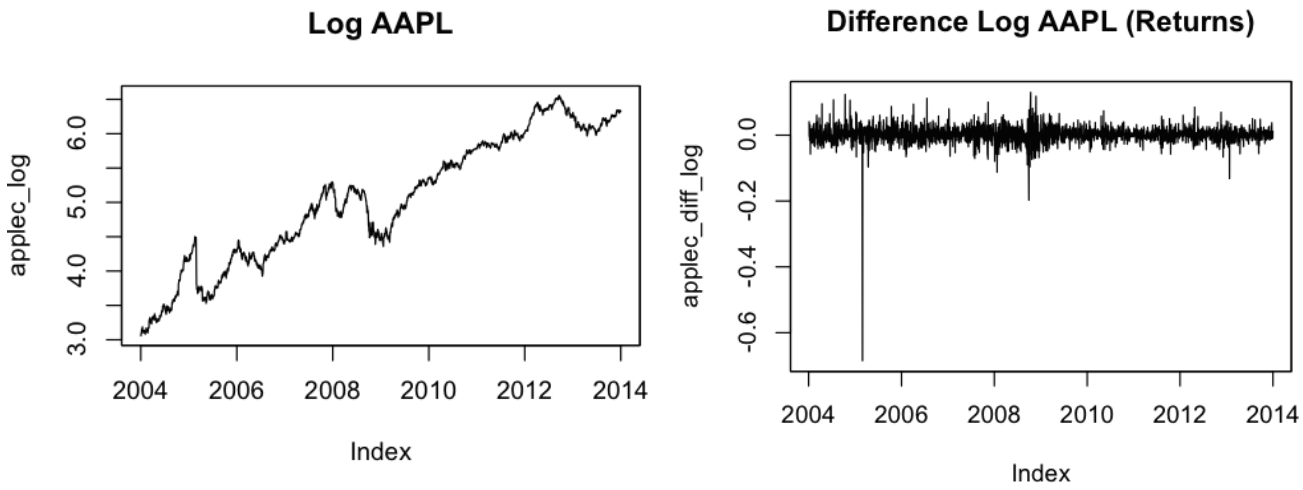
```
# Differencing and plotting the original series #
plot(applec)

# Differencing and plotting the original series #
applec_diff=diff(applec)
plot(applec_diff,type='l',main='Difference AAPL')
```



The graph on the left is the original time series of Apple stock price from 01/02/2004 to 12/31/2013, showing exponential growth while the graph on the right, shows the differences of apple stock prices. It can be seen that the variance of the series increases as the level of original series increases, and therefore, it is not stationary.

```
# Take log of original series and plot the log price #
applec_log=log(applec)
plot(applec_log,type='l',main='Log AAPL')

# Differencing log price and plotting it #
applec_diff_log=diff(applec_log)
```

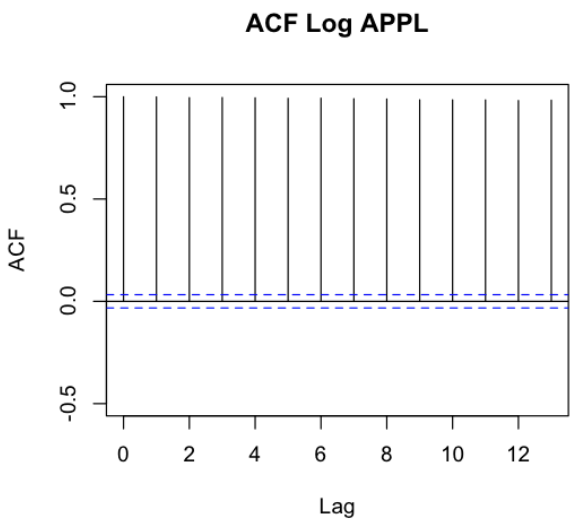
```
plot(applec_diff_log,type='l',main='Difference Log AAPL (Returns)')
```

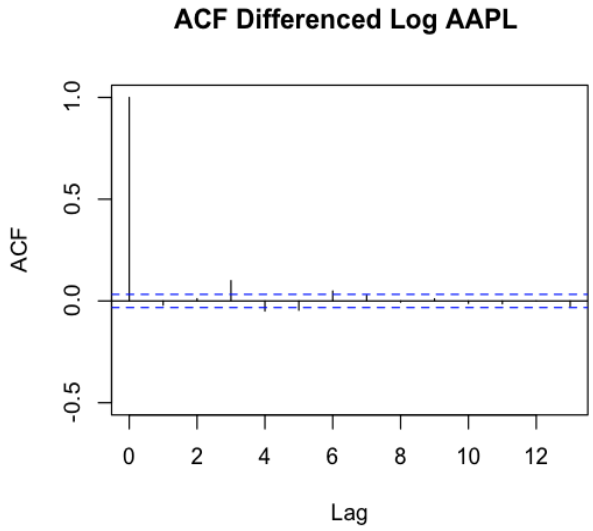


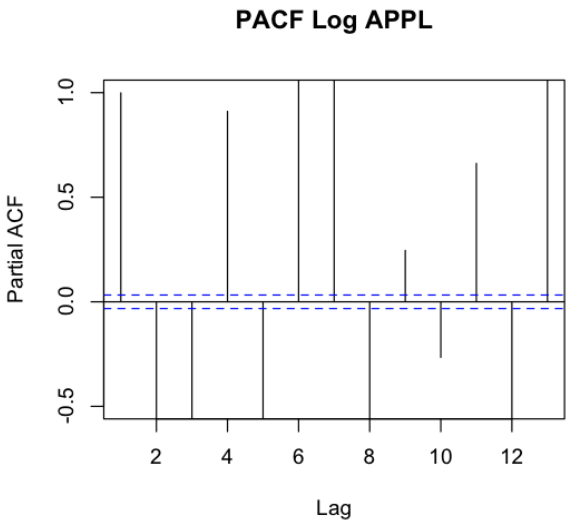
The graph on the left shows the log price of Apple. The series is more linear compared to the original "Daily Close Price of AAPL' while the graph on the right shows the differences of log price of Apple. This series seems more mean- reverting, and variance is constant and does not significantly change as level of original series changes.

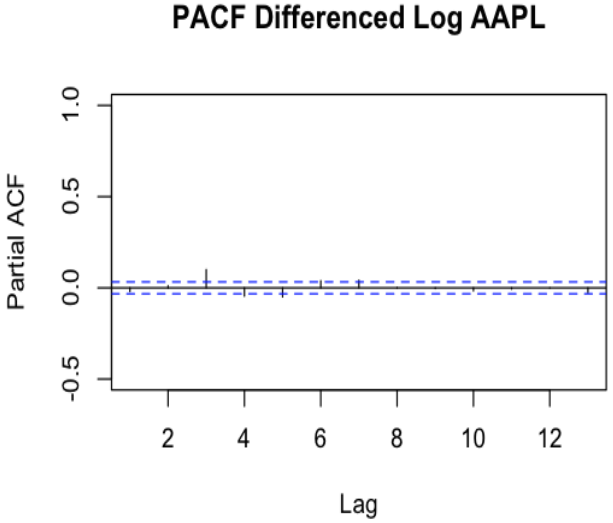
Next, we use autocorrelation and partial autocorrelation to obtain the input parameters for the 'arima' function to fit the ARIMA model. We also plot the graphs for them, which are shown below:

```
# Autocorrelation and Partial autocorrelation - ACF and PACF of applec #
acf(applec_log,main='ACF APPL',lag.max=13, na.action = na.pass, ylim=c(- 0.5,1))
pacf(applec_log,main='PACF AAPL',lag.max=13, na.action = na.pass, ylim=c(- 0.5,1))

# Autocorrelation and Partial autocorrelation - ACF and PACF of differenced logged
AAPL #
acf(applec_diff_log,main='ACF Differenced Log AAPL ',lag.max=13, na.action =
na.pass, ylim=c(- 0.5,1))
pacf(applec_diff_log,main='PACF Differenced Log AAPL',lag.max=13, na.action =
na.pass, ylim=c(- 0.5,1))
```









The upper left graph shows the ACF of Log Apple stock price where the ACF stays constant which shows that the model was differenced where as, the upper right graph shows the ACF of differences of log Apple with not many significant lags.

The lower left graph shows the PACF of Log Apple, indicating significant value at multiple lags where as, the lower right shows the PACF of differences of log Apple, showing no significant lags. The model for differenced log Apple series is thus a white noise. Another way to check and identify the model is using AICs. AIC refers to Akaike Information Criterion and according to the model, we select the lowest AIC. Therefore, from the autocorrelation and partial autocorrelation as well as AIC method, we used the various p and q values and the differencing value d =1 (since the series was stationary after one differencing action was performed) to build the ARIMA model.

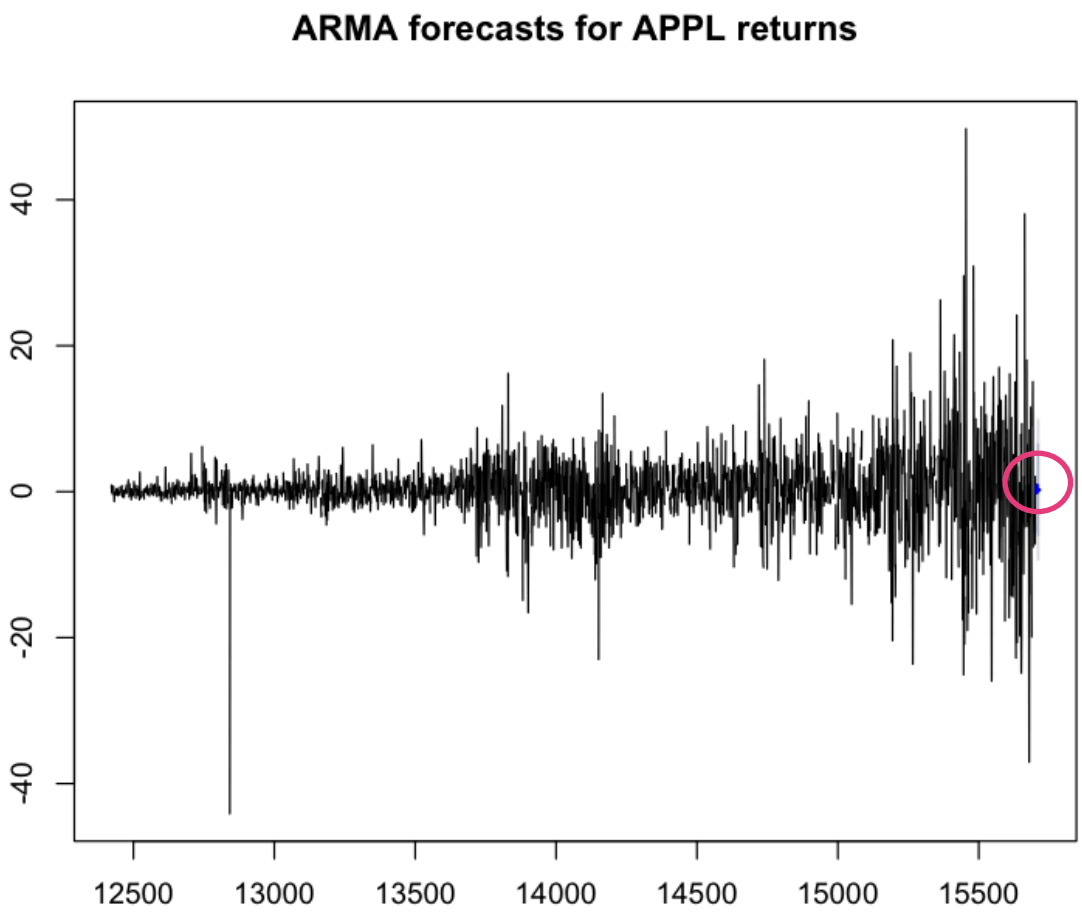
We begin by first building the ARMA model. The Autoregressive-moving-average (ARMA) models are based on two polynomial functions; one for the auto regression (AR) and a second for the moving-average (MA). We build the model by dividing the data into a training and testing dataset and then fit the model with 'arima' function and then plot the graph for the ARMA forecast. Once that has been done, we measure the accuracy of the ARMA model by using the 'accuracy' function as shown below.

```
# Build train and test dataset for ARMA model #
applec_diff_train <- applec_diff[1:(0.9 * length(applec_diff))]  # Train dataset
applec_diff_test <- applec_diff[(0.9 * length(applec_diff) + 1):length(applec_diff)]  #
Test dataset


# Fit the model with ARIMA function #
fit <- arima(applec_diff_train, order = c(2, 1, 2))
arma.preds <- predict(fit, n.ahead = (length(applec_diff) - (0.9 *
length(applec_diff))))$pred
arma.forecast <- forecast(fit, h = 10)
summary(arma.forecast)
```

From the summary that is obtained, only the autoregressive portion of the ARMA model is significant.

```
# Plot ARMA forecasts for APPL returns #
plot(arma.forecast, main = "ARMA forecasts for APPL returns")

# Accuracy of the ARMA model for APPL #
accuracy(arma.preds, applec_diff_test)
```

```
                    ME     RMSE      MAE      MPE     MAPE
Test set -0.1589597 8.56877 6.09036 101.4306 103.8242
```



Therefore, the blue and grey regions in the right side of the graph (which isn't clearly visible in the screenshot provided) provide us the 99% and 95% confidence level for the forecasts respectively. An intrinsic shortcoming of ARMA models, which is evident from the plot above, is the assumption of mean reversion of the series, which means that the model over a long period of time, takes the average stock prices and therefore is a poor choice for long-term predictions. Also, the performance/accuracy of the model is represented by the RMSE value is 8.56 which is obtained from the 'accuracy' function. The lower the accuracy value, the better the performance of the model. We will compare the performance/accuracy of ARMA and ARIMA once we obtain the RMSE value for the ARIMA model.

In the next step, we build the ARIMA model and then compare it with the ARMA model obtained above.

```
# Build ARIMA model #
arima(applec_log)       #ignores mean#
ArimaModel <- arima(applec_log, order = c(2,1,2), include.mean=TRUE)
ArimaModel
```

```
Call:
arima(x = applec_log, order = c(2, 1, 2), include.mean = TRUE)

Coefficients:
          ar1      ar2     ma1     ma2
      -0.7348  -0.7367  0.7442  0.6864
s.e.   0.1358   0.2886  0.1568  0.3004

sigma^2 estimated as 0.0005149:  log likelihood = 5653.33,  aic = -11296.65
```

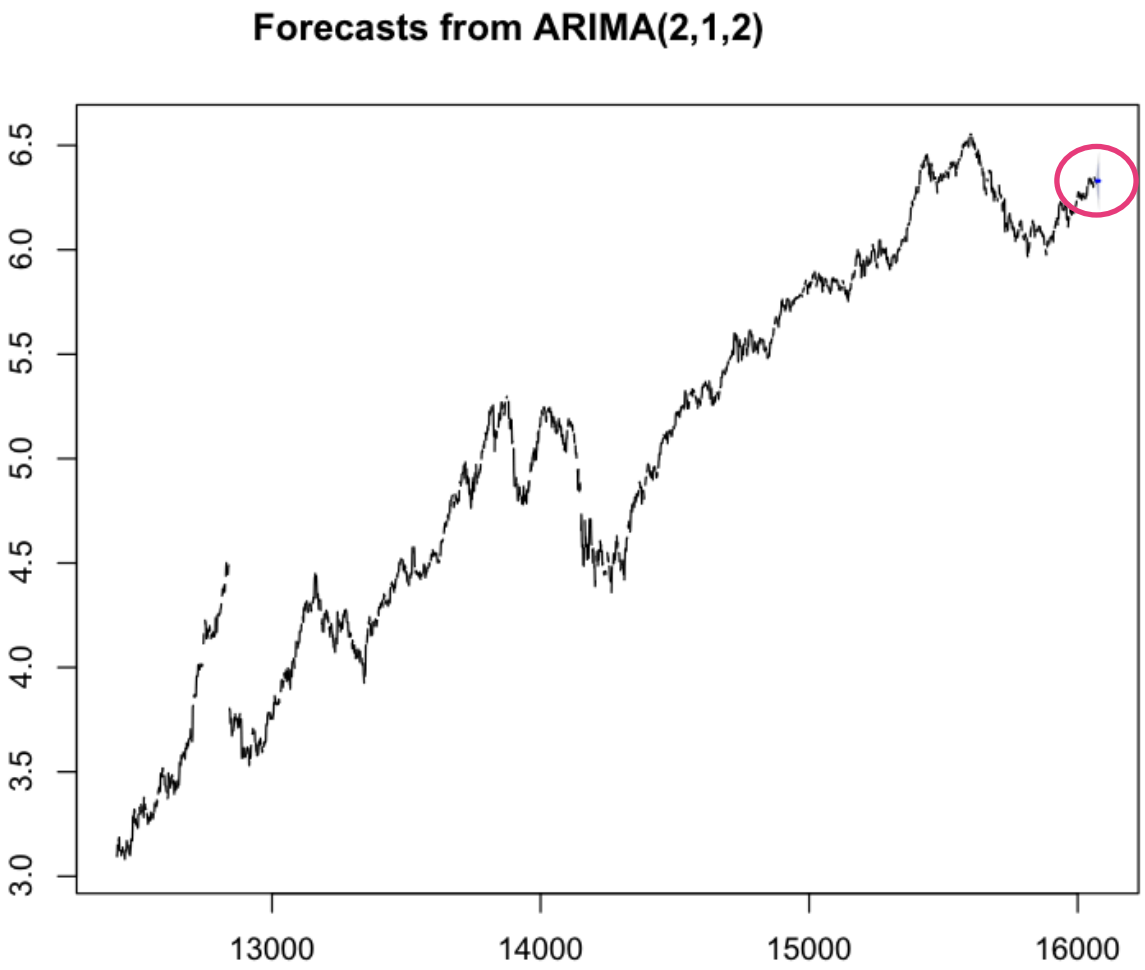
When the model is run with various values for the p and q parameters, the AICs are outputted with the AIC for values (2,1,2) being the lowest at -11296.65. We have also included a table below for to show the comparisons.

Model	AICs
0 1 0	-11296.05
1 1 0	-11294.53
0 1 1	-11294.55
1 1 1	-11294.26
0 1 2	-11294.7
1 1 2	-11294.1
2 1 0	-11294.51
2 1 1	-11294.21
2 1 2	-11296.65

```
# Summary and Accuracy of the ARIMA model #
summary(ArimaModel)
```

```
                ME        RMSE         MAE         MPE       MAPE
0.001114902 0.02268806 0.01520043 0.02448598 0.3233534
```

```
# Display all elements of the output, in case we want to extract specific parts of it for
further computation and then index each element with the $ and name #
names(ArimaModel)
ArimaModel$coef

# Forecasting future time points (10 in this case but you can change this by
changing the h value) #
forecast(ArimaModel, h = 10)
plot(forecast(ArimaModel))
```



Therefore when we compare our initial daily close price graph (page 4) to the forecast table (we take the first point estimate as shown below) and the ARIMA forecast graph shown above, we can see that in the latter, the time series is stationary and the rather small interval (shaded region to the right of the graph that is not clearly visible in the screenshot) around the points estimate (h = 10) is reasonable given the small fluctuations of the existing series. The blue shading represents the 80% confidence interval while the gray shading represents the 95% interval.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
16071	6.330122	6.300966	6.359278	6.285532	6.374713

When the ARMA model is compared to the ARIMA model, the blue and grey regions of the ARMA model provide us the 99% and 95% confidence level for the forecasts respectively however, the assumption of mean reversion of the series from the ARMA forecast plot indicates that the model is poor for long-term predictions unlike the ARIMA model.

Lastly, when we also compare the performance/accuracy of the ARMA and the ARIMA models, the ARIMA model performs better which is evident from the lower RMSE (0.022) of the model compared to the RMSE (8.56) of the ARMA model. As a result, we can conclude that our model successfully forecasts the returns since the forecasted return percentage is well within our 95% confidence interval and very close to the lower limit.

Conclusions

The time series analysis and forecasting of financial data (stocks) provided us with a wide variety of information through the different models that we used. We learnt how we could analyze the information, which is easily obtainable from Yahoo Finance and then build our own models with varying parameters to help forecast financial data as per our needs. These techniques would not just be useful in the financial industry but in various other industries because of the flexibility to build our own models based on one's own needs and requirements. We should accept the ARIMA model because it performed the better than the ARMA model due to the lower RMSE value. It also had very small fluctuations in the existing series and its prediction was well within the 95% confidence interval.

In the next steps, one can build the ARCH/GARCH model which is a method to measure volatility of the series, or more specifically, to model the noise term of ARIMA model. This model would incorporate new information and analyze the series based on conditional variances where users can forecast future values with up-to-date information. The forecast interval for the ARCH/GARCH model would be closer than that of ARIMA only model. As a result, the project material can be easily applied to business problems by simply downloading the most up-to-date data and setting up the parameters that provide the best forecasting results along with the best performance and also take it a step further by building the ARCH/GARCH model. These financial forecasts obtained are an estimate of future income and expenses for a business over the next year and are used to develop projections of profit and loss statements, balance sheets, and most critically the cash flow forecast.

References

Zhao, Y. (2013). R and Data Mining Examples and Case Studies. S.l.: Academic Press.

Starkweather, J. (2015). Time Series Analysis: Basic Forecasting. (Vol. 04). Benchmarks RSS Matters.

Forecasting stock returns using ARIMA model with exogenous variable in R. (2013, April 28). Retrieved May 3, 2015, from http://programming-r-pro-bro.blogspot.in/2013/04/forecasting-stock-returns-using-arima.html

ARMA Models for Trading. (2012, August 21). Retrieved May 3, 2015, from http://www.r-bloggers.com/arma-models-for-trading/