HarvardX - Data Science - Choose You Own Project

David Forrester

23/12/2019

Overview

This project will look into the total stock price data for apple from listing on the stock exchange in 1985 to 2018. 4 different methods will be used to attempt to predict and forecast the close price each day. The data will be split into a training and test set for training and validating the methods.

A root mean square error calculation will be used to assess the performance of the predictions with a goal of returning the smallest value.

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

Dataset

The dataset is loaded into r as a comma separated value type from the git repo. This data is then split into a training and test set of 95% to 5%. There is no random split between the data set as it is a time series and therefore the previous rows are highly important in determining the next.

```
library(tidyverse)
library(lubridate)
library(stringr)
library(rvest)
library(XML)
library(tidytext)
library(wordcloud)
library(dslabs)
library(caret)
library(pracma)
library(MASS)
library(tseries)
library(forecast)
library(tsfknn)

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://
cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://
cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos =
"http://cran.us.r-project.org")
if(!require(tsfknn)) install.packages("tsfknn")

# relative path to file
file <- ("aapl.us.txt")

# Read csv file into environment
aapl <- read_csv(file)

# Split the data set by date ranges 80% train 20% test
aapl_train <- aapl[1:round(nrow(aapl) * 0.95),]
aapl_test <- na.omit(aapl[round(nrow(aapl) * 0.95) + 1:nrow(aapl),])
```

Methods and Analysis

Data Analysis

The first 6 rows and the summary below give insight into the structure of the data. It can be seen that each row represents a day of trading for the apple stock with the open, close, volume and intraday prices.

```
## # A tibble: 6 x 7
##   Date        Open  High   Low Close   Volume OpenInt
##   <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>   <dbl>
## 1 1984-09-07 0.424 0.429 0.419 0.424 23220030       0
## 2 1984-09-10 0.424 0.425 0.414 0.421 18022532       0
## 3 1984-09-11 0.425 0.437 0.425 0.429 42498199       0
## 4 1984-09-12 0.429 0.432 0.416 0.416 37125801       0
## 5 1984-09-13 0.439 0.441 0.439 0.439 57822062       0
## 6 1984-09-14 0.441 0.456 0.441 0.446 68847968       0
```

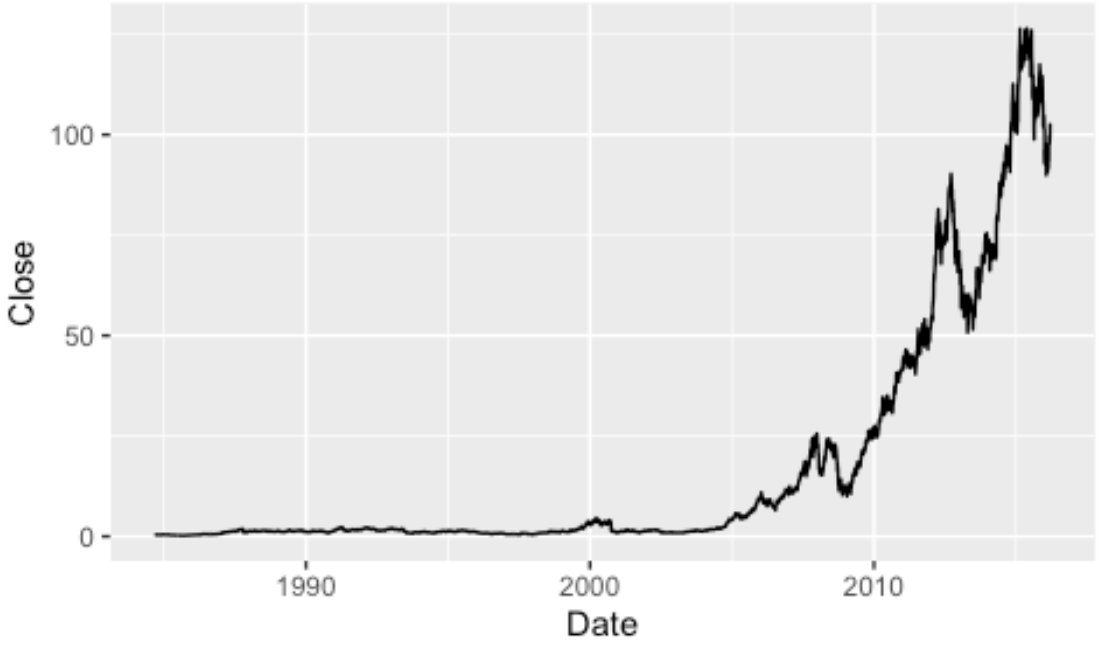
Looking at the summary of the training data set it can be seen that there is no missing data.

```
##       Date                 Open               High
##  Min.   :1984-09-07   Min.   :  0.2331   Min.   :  0.2356
##  1st Qu.:1992-07-16   1st Qu.:  1.1128   1st Qu.:  1.1371
##  Median :2000-05-28   Median :  1.5355   Median :  1.5660
##  Mean   :2000-06-05   Mean   : 16.8422   Mean   : 17.0179
##  3rd Qu.:2008-04-27   3rd Qu.: 16.6885   3rd Qu.: 17.0115
##  Max.   :2016-03-17   Max.   :127.7800   Max.   :127.8600
##       Low                Close              Volume
OpenInt
##  Min.   :  0.2305   Min.   :  0.2305   Min.   :0.000e+00   Min.
:0
##  1st Qu.:  1.0886   1st Qu.:  1.1153   1st Qu.:4.743e+07   1st
Qu.:0
##  Median :  1.5015   Median :  1.5369   Median :7.834e+07   Median
:0
##  Mean   : 16.6435   Mean   : 16.8330   Mean   :1.106e+08   Mean
:0
##  3rd Qu.: 16.3358   3rd Qu.: 16.6465   3rd Qu.:1.373e+08   3rd
Qu.:0
```

```
##  Max.   :125.4200   Max.   :126.4900   Max.   :2.070e+09   Max.
:0
```

Below is a plot of the training data set over time allowing for the trend to be easily seen.

```
aapl_train %>%
  ggplot() +
  geom_line(aes(Date, Close))
```



The below trend is of the test data which the models below hope to predict.

```
aapl_test %>%
  ggplot() +
  geom_line(aes(Date, Close))
```

Modelling Approach

The modelling approaches in this project will be assessed on their root mean square error to determine the most successful model. The goal will be to return the smallest RMSE when comparing the predictions with the testing data set.

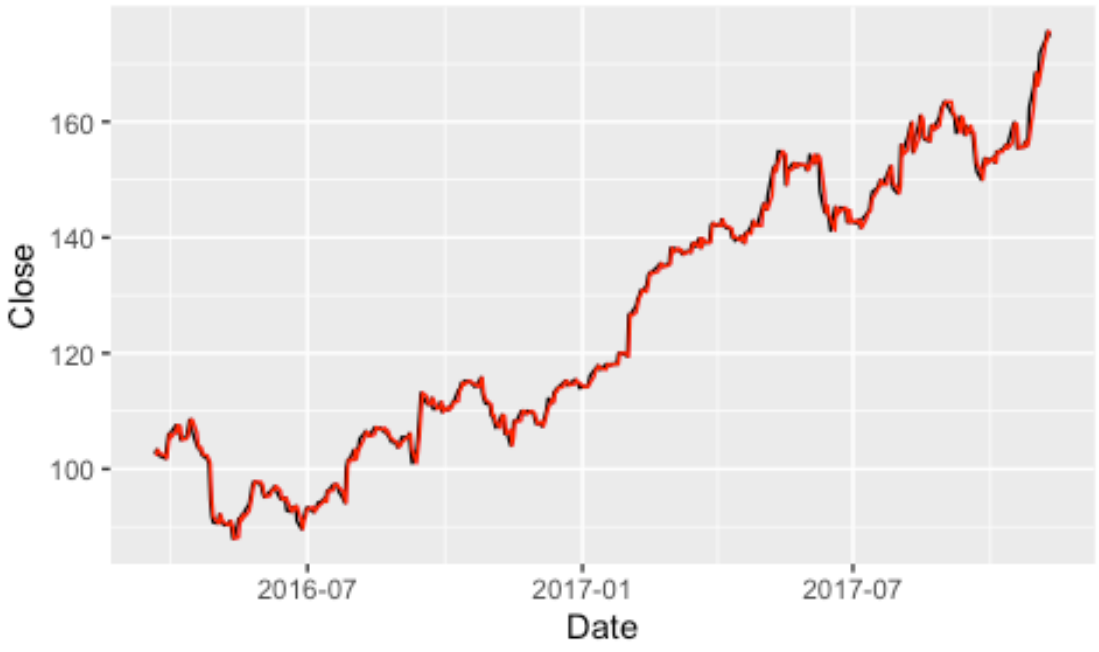
Model 1: Previous Close

The first modelling approach is to simply use the previous close value as the prediction for the current close value. This is typically used as a starting point for comparing future models when looking at stock prices.

```
aapl_test <- mutate(aapl_test, PrevClose = lag(Close))
aapl_test <- na.omit(aapl_test)
```

Below the test data set is plotted along with the previous value to give insight into the modelling approach. As it can be seen this approach lags the actual value and is therefore very unlikely to match up.

```
aapl_test %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = Close), colour = "black") +
  geom_line(aes(y = PrevClose), colour = "red")
```



Putting the actual and predicted values into the RMSE formula a results of 1.51 is returned. This will be used as a starting point and a baseline for comparing the other models.

```
naive_rmse <- RMSE(aapl_test$Close, aapl_test$PrevClose)

rmse_results <- data_frame(method = "Previous Value Prediction", RMSE
= naive_rmse)

## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.

rmse_results

## # A tibble: 1 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Previous Value Prediction  1.51
```

Model 2: Moving Average

The next modelling approach is to use the moving average of previous close values to determine the next. when selecting a moving average typically a window (n) of previous values is selected. This can range from 2 to infinite.

```
windows <- seq(2, 20, 1)
```

Setting up a window between 2 and 20 at 1 value increments the resulting RMSE for the training data can be tested.

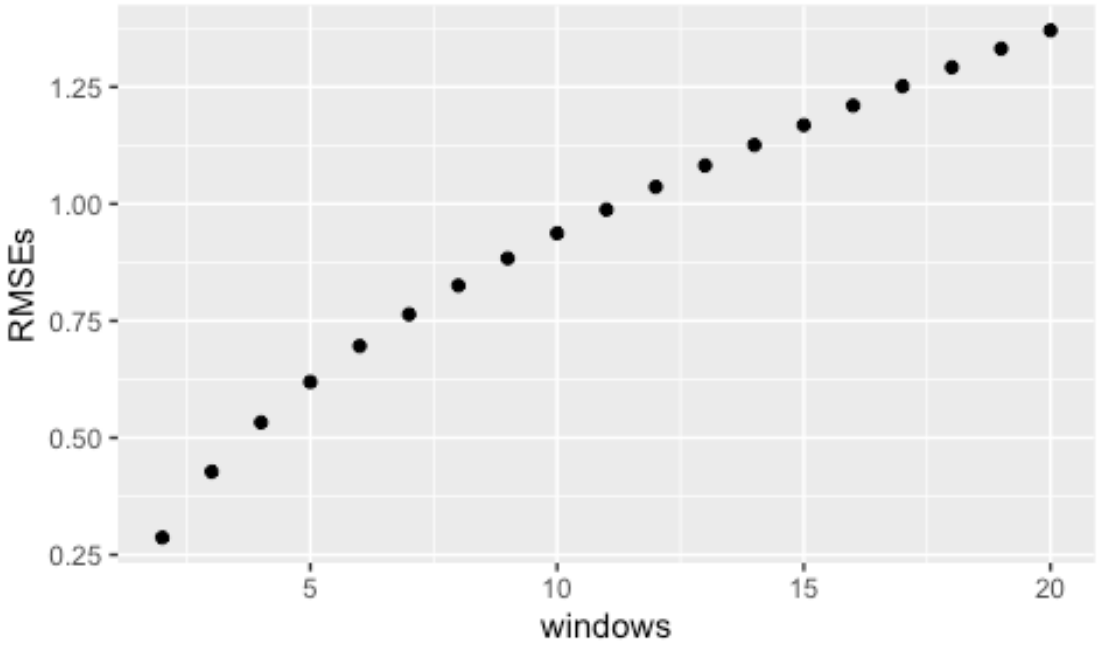
```
RMSEs <- sapply(windows, function(w){

  aapl_train <- mutate(aapl_train, MovingAvg =
movavg(aapl_train$Close, w, type=c("s")))

  return(RMSE(aapl_train$Close, aapl_train$MovingAvg))
})
```

The results are plotted below and as it can be seen the window of 2 gives the most accurate result. This is expected as the larger the window the less accurate the moving average becomes in terms of short term predictions and the better it becomes are predicting long term trends.

```
qplot(windows, RMSEs)
```



Taking this window and applying it to the test data is conducted below in this code.

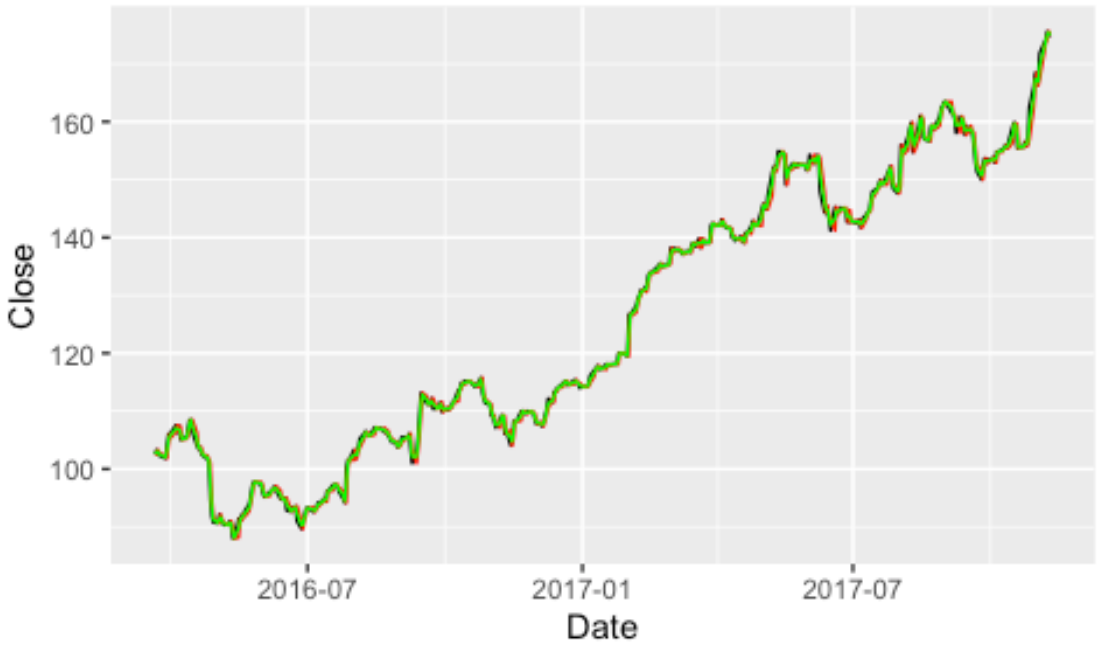
```
best_window <- windows[which.min(RMSEs)]
best_window
```

```
## [1] 2
```

```
aapl_test <- mutate(aapl_test, MovingAvg = movavg(aapl_test$Close,
best_window, type=c("s")))
```

Overlaid in green the new prediction for the close values can be seen. From the plot it can be determined that the moving average approach smooths out the sharpe predictions from the previous value model.

```
aapl_test %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = Close), colour = "black") +
  geom_line(aes(y = PrevClose), colour = "red") +
  geom_line(aes(y = MovingAvg), colour = "green")
```



The moving average model provides a significant improvement in the RMSE value as can be seen in the table below.

```
naive_rmse <- RMSE(aapl_test$Close, aapl_test$MovingAvg)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Moving Average
Prediction",
                                     RMSE = naive_rmse))
rmse_results
```

```
## # A tibble: 2 x 2
##   method                     RMSE
##   <chr>                     <dbl>
## 1 Previous Value Prediction 1.51
## 2 Moving Average Prediction 0.755
```

Model 3: KNN Forecasting

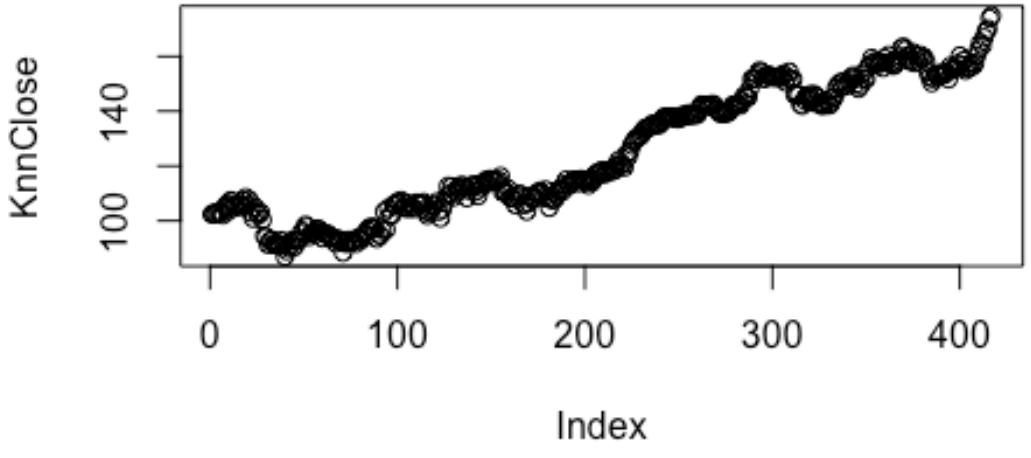
This next method looks to use k nearest neighbours to predict the closing value. The approach has implemented a for loop to predict the next value in the sequence as using just the training data to predict 416 days out caused issues with the scaling. This came in the form of the predictions not exceeding 110 as values higher than this aren't experienced in the training data set.

```
KnnClose <- aapl[7946,]$Close

for (n in seq(1, 416, 1)) {
  PredKnn <- knn_forecasting(aapl[1:7946 + n,]$Close, h = 1, lags =
1:2, k = 2)
  KnnClose <- append(KnnClose, PredKnn$prediction)
}
```

The plot below shows the predicted values using the KNN model over the 417 days.

```
plot(KnnClose)
```



The KNN method returns a less than desirable RMSE score of 2.5. This is likely due to the stock trend of Apple continues upwards from bottom right to top left. There is therefore minimal repeating prices in the historical data used to predict the next value.

```
df <- data.frame(aapl_test$Close, KnnClose)
naive_rmse <- RMSE(df$aapl_test.Close, df$KnnClose)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="KNN Prediction",
                                     RMSE = naive_rmse))
rmse_results
```

```
## # A tibble: 3 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Previous Value Prediction 1.51
## 2 Moving Average Prediction 0.755
## 3 KNN Prediction            2.59
```

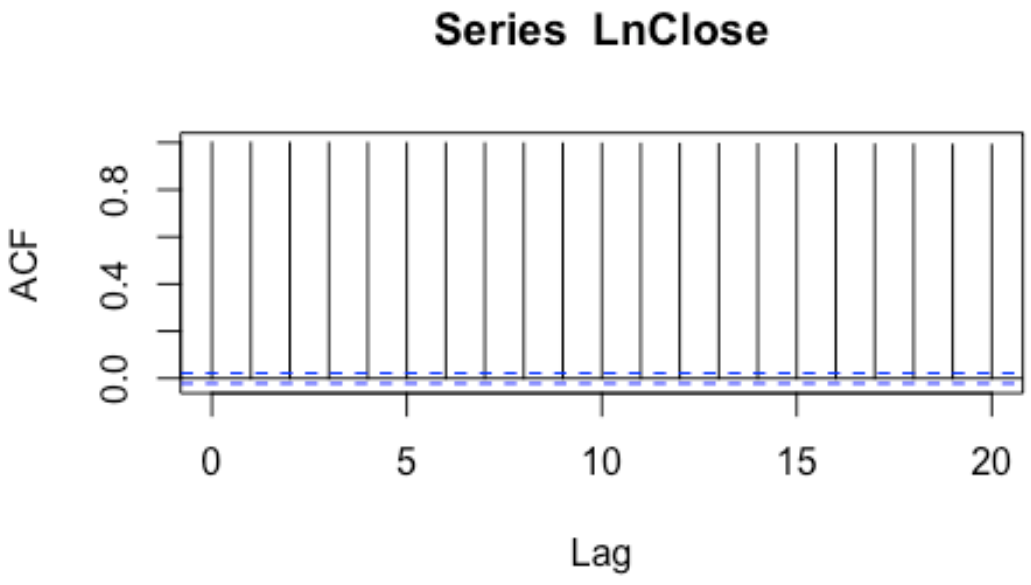
Model 4: ARIMA

The final model used to predict the is auto regressive integrated moving average or also known as ARIMA. This is very commonly used in time series forecasting and should therefore fit the use case well. 3 parameters are used to tune the model however, in this case auto ARIMA will be used to determine p, q and d.

```
LnClose = log(aapl_train$Close)
```

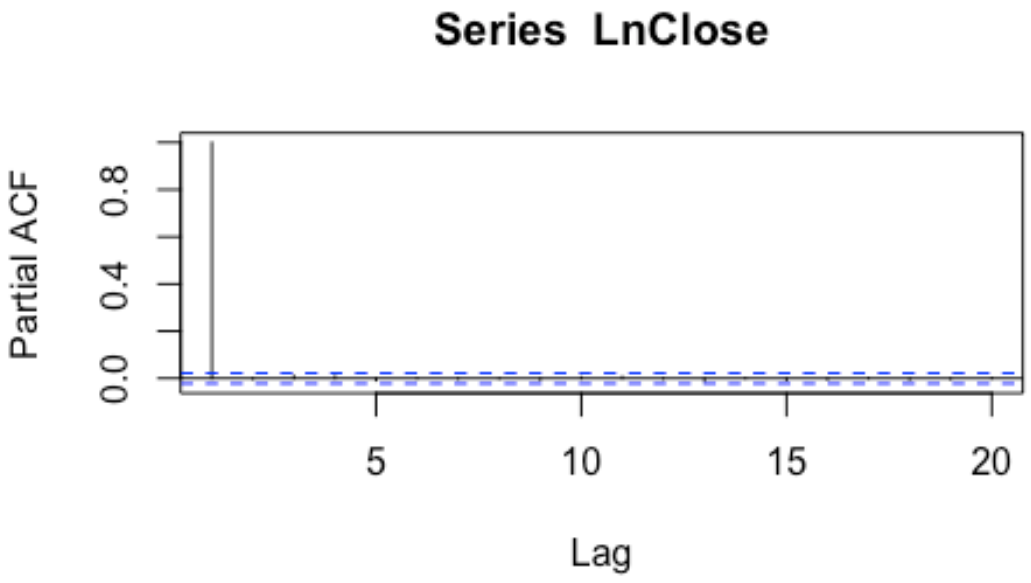
From the plot below it can be seen that the Auto - correlation function for the time series is highly correlated even out to a lag of 20 previous values. The 2 plots allow for a deeper understanding of the time series and whether it is more auto regressive or moving average and which order to use them in.

```
acf(LnClose, lag.max = 20)
```



From the plots above and below it can be determined that the apple stock price time series is more auto regressive.

```
pacf(LnClose, lag.max = 20)
```



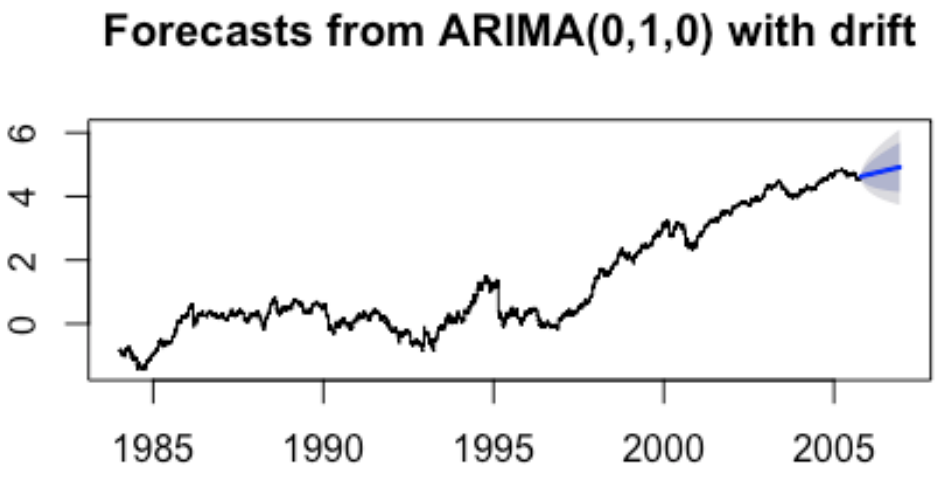
The next step is to set the starting point and the frequency of the predictions. The close price is then pasted into the auto ARIMA function.

```
CloseArima <- ts(LnClose, start = c(1984, 09), frequency = 365)
FitCloseLn <- auto.arima(CloseArima)
FitCloseLn
```

```
## Series: CloseArima
## ARIMA(0,1,0) with drift
##
## Coefficients:
##       drift
##       7e-04
## s.e.  3e-04
##
## sigma^2 estimated as 0.0008698:  log likelihood=16722.29
## AIC=-33440.57   AICc=-33440.57   BIC=-33426.61
```

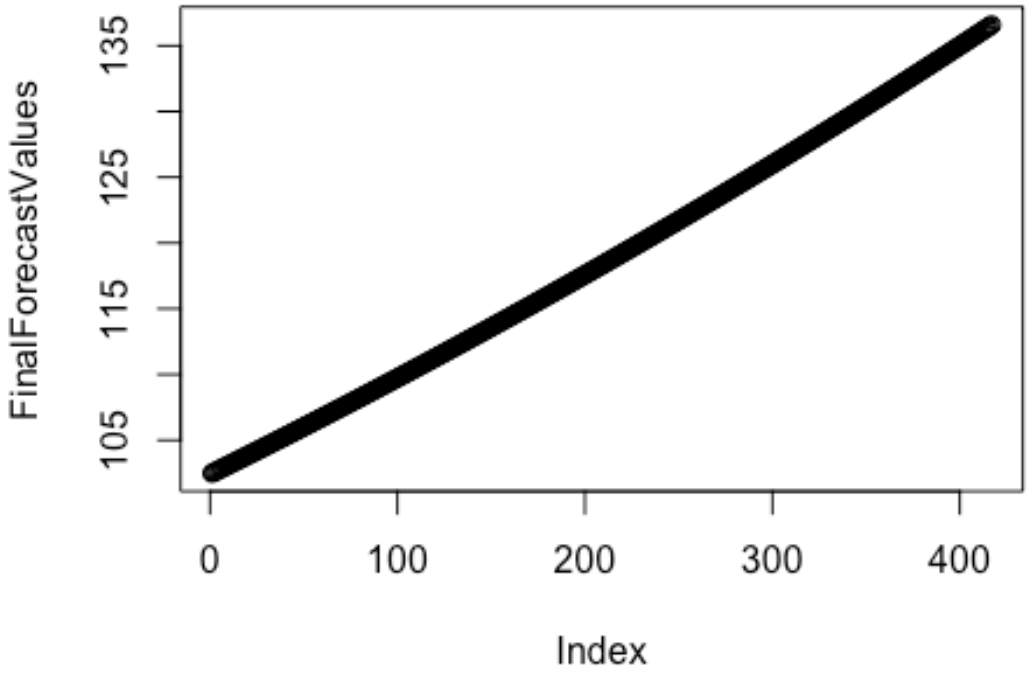
A forecast can then be generated and forecast out for the range of the testing data set.

```
ForecastValueLn = forecast(FitCloseLn, h = 417)
plot(ForecastValueLn)
```


Forecasts from ARIMA(0,1,0) with drift
0
2
4
6
1985
1990
1995
2000
2005

As it can be seen in the above and below charts the forecast is very linear and will therefore likely return a poor RMSE.

```
ForecastValuesExtracted = as.numeric(ForecastValueLn$mean)
FinalForecastValues = exp(ForecastValuesExtracted)
plot(FinalForecastValues)
```



This is confirmed when the ARIMA forecast data is compared to the actual test data. This is due to ARIMA being great for determining the trend of the stock price, however, not the day to day fluctuations.

```
df <- data.frame(aapl_test$Close, FinalForecastValues)
naive_rmse <- RMSE(df$aapl_test.Close, df$FinalForecastValues)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="ARIMA Prediction",
```

```
                                     RMSE = naive_rmse))
rmse_results
```

```
## # A tibble: 4 x 2
##   method                        RMSE
##   <chr>                        <dbl>
## 1 Previous Value Prediction  1.51
## 2 Moving Average Prediction  0.755
## 3 KNN Prediction             2.59
## 4 ARIMA Prediction          16.3
```

Results

The results below present the RMSE for the 4 different modelling approaches used to attempt to predict the future values of the apple stock price.

```
rmse_results
```

```
## # A tibble: 4 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Previous Value Prediction  1.51
## 2 Moving Average Prediction  0.755
## 3 KNN Prediction             2.59
## 4 ARIMA Prediction          16.3
```

Conclusion

Overall the moving average prediction model was the most accurate and would therefore be used to build upon in future investigation. Further improvements could be to use linear regression and even incorporate the other columns into the model such as the range of prices which the stock traded in the day or the volume of stock traded.