**Module 3: Week 3 Introduction to Enterprise Analytics**

**Author’s Name:** Shivani Sharma

**Course Number:** ALY6050

**Course Title:** Intro to Enterprise Analytics

**Academic Term:** Fall 2019 CPS Analytics

**Instructor’s Name:** Dr Alex Huang

**Assignment Completion Date:** 01-28-2020



**Introduction**

This assignment is based on the concept of regression and time series forecasting. A “Time Series” algorithm is based on the stream of historical data with an assumption that the factors contributing in the behavior or pattern of the value to be predicted will continue to do so. A time series can be bifurcated into components so that it can be systematically understood by someone in order to analyze it to build model on top of it for doing forecasting.

In order to create a time series in R there is a function ts(). We need to feed input data to this function.

The mathematical definition for a time series is as follows:

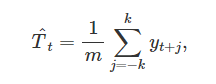
Let’s suppose a random variable “X” which is indexed with respect to time denoted by t, the observations denoted by the following expression is known as time series:



Where “N” is any integer value.

Now starting with the brief introduction of the various models that we have to use in this assignment.

“Moving Average” : In statistics world we talk about averages many time, and this model is related to averages only which means to analyze or study the data points by creating a series with averages with different subsets of parent data set. Order selection is very important in this technique. It can be represented with following mathematical formula:



**Exponential Smoothing Forecast:** This is basically applied to univariate data set which can be enhanced to support data with the help of systematic trend or seasonal component. The forecasted values produced by this are weighted averages of historic observations. This can be mathematically represented as follows:



Where alpha is the smoothing constant which is provided to us and it lies between 0 and 1 and t is the time period.

Overall to be very precise for time series , there are following components of a time series:

1. Trend: It is a regular increase or decrease in the units of data i.e there are consistent uptrends and downtrends.
2. Seasonality: This shows periodic effects in one complete observation cycle.
3. Random Events: This is the observation of events which are picked up on adhoc basis.

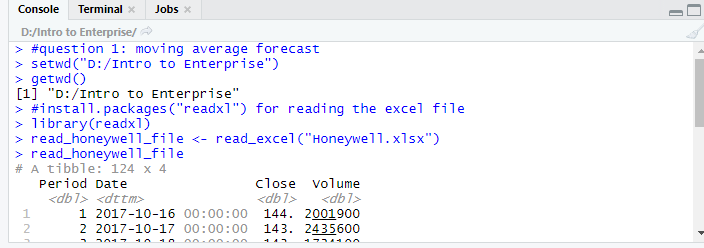
I would like to explain the problem statement with the logic in analysis section.

**Analysis**

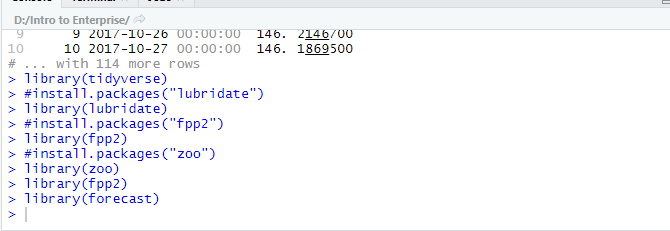
We have been provided with a data set containing historic stock prices of a company names “Honeywell” from a date range 10/15/2017 to 4/15/2018 and there are 3 problem statements in this assignment. Let’s start with the analysis of each one by one.

1. In this we have to perform the “Moving Average” forecast on the stock prices provided in the data set for date 4/16/2018. Following is the logic for the same:

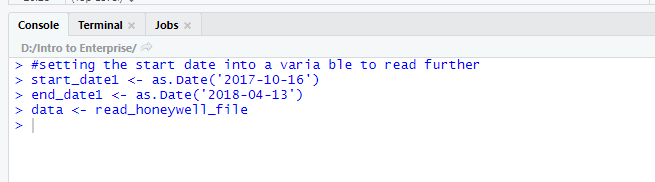
First of all I used the readxl library to directly read the excel sheet.



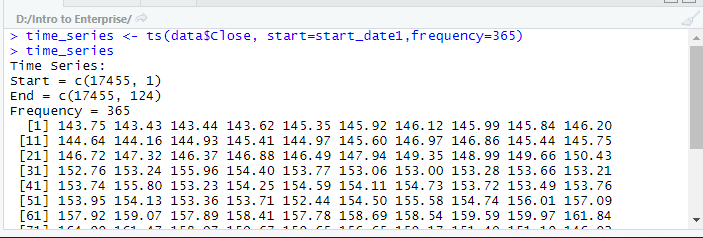
Following are the list of required libraries to be imported.

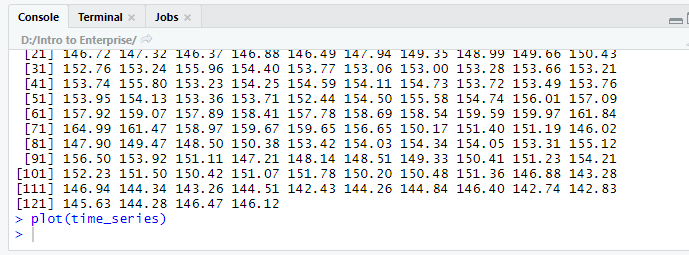


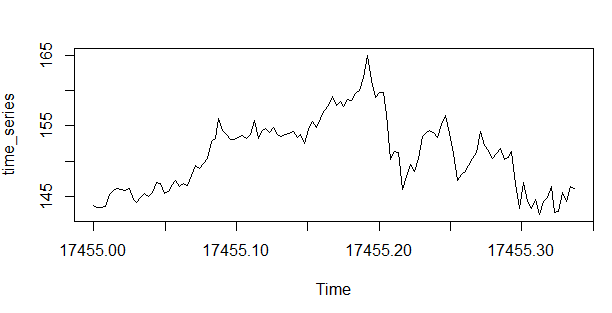
I have specified the start date and end dates as per the data set provided and stored them in different variables which are further going to be used.

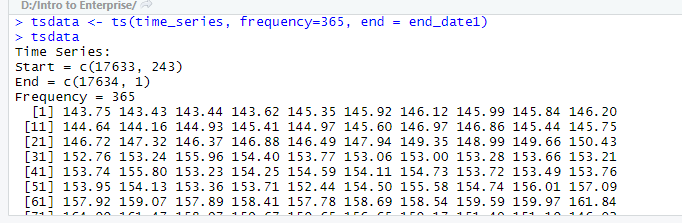


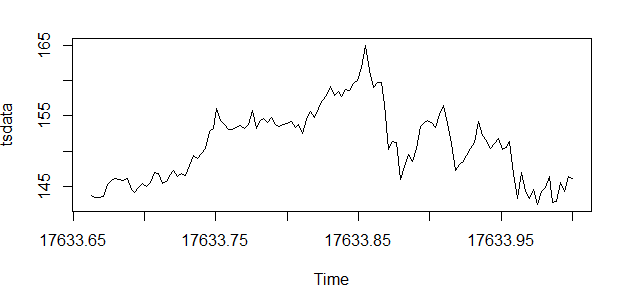
Before applying the moving average algorithm we need to create the time series for the data set and following is the logic for the same.







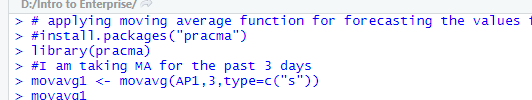


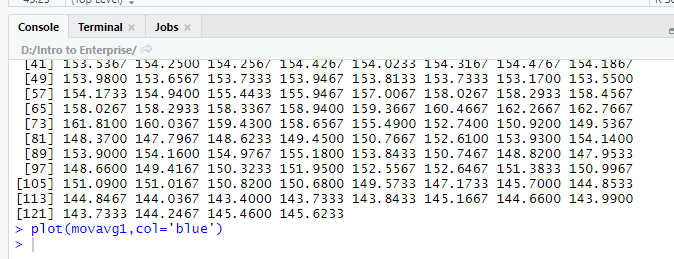


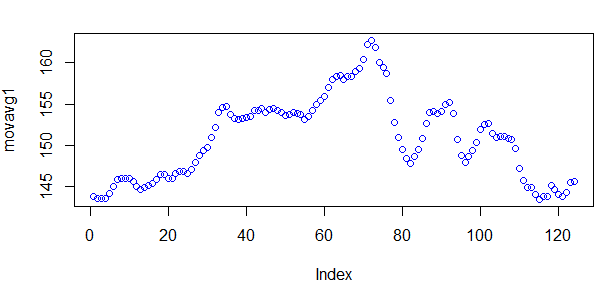
Now for moving average calculation the time series function needs to converted into the vector form as the “movavg()” reads the vector form only.

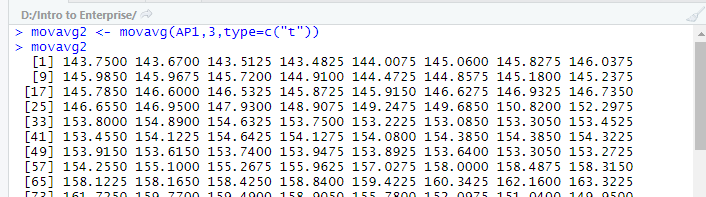


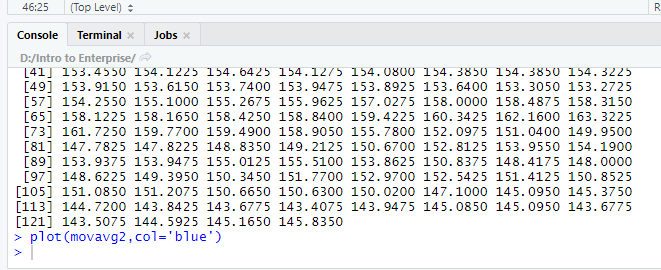
Now for moving average function I have takes the duration as 3 days and for each type of moving average category I am calculating the moving averages for the data set and following is the logic and output for the same. I have also plotted graphs for each type and there is a difference in the behavior of each type in the last section of the graph. Initially it is approximately same for all.

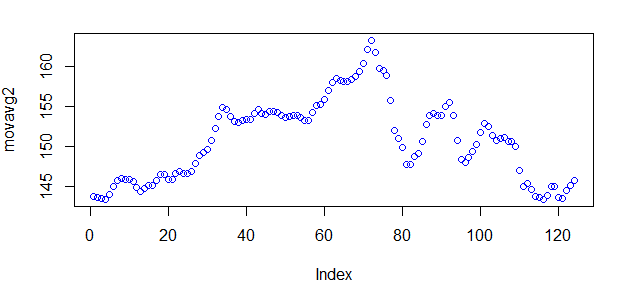


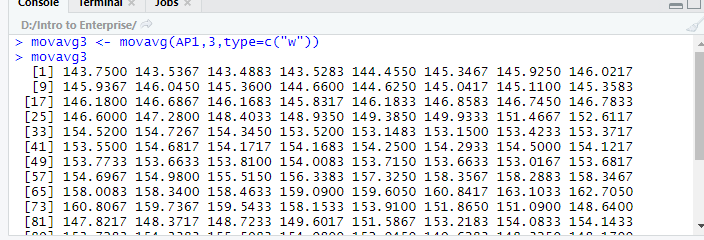


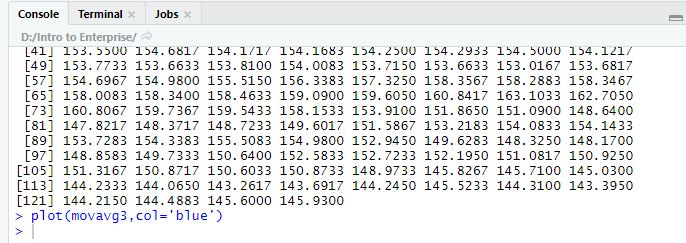


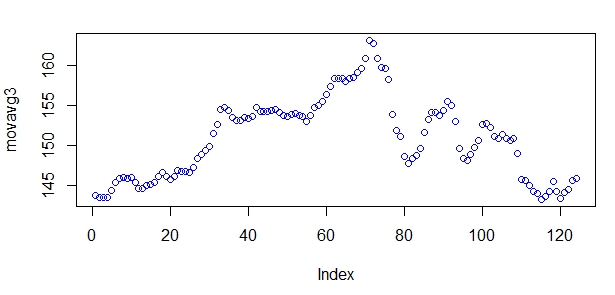


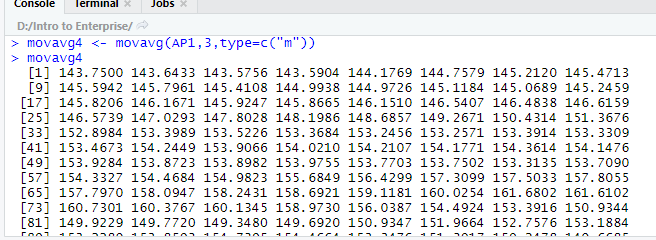


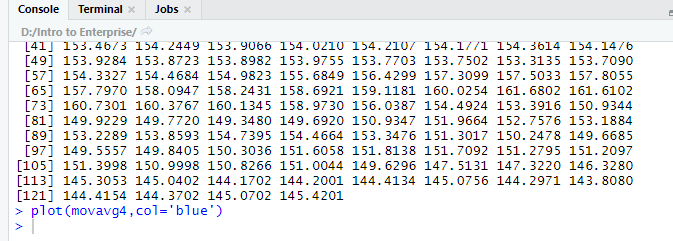


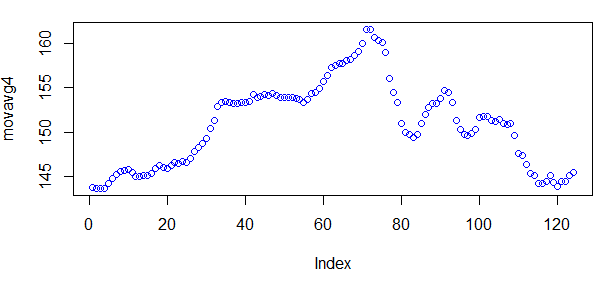


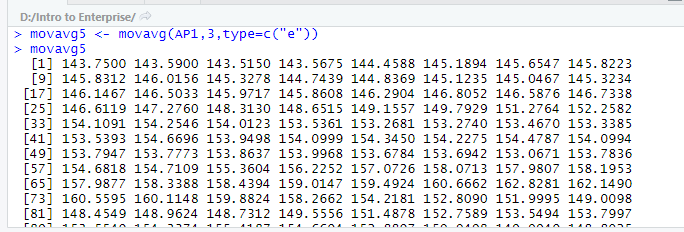


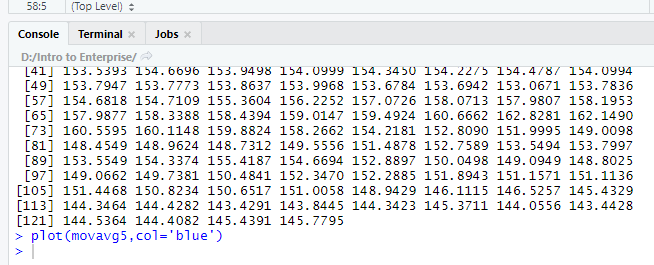


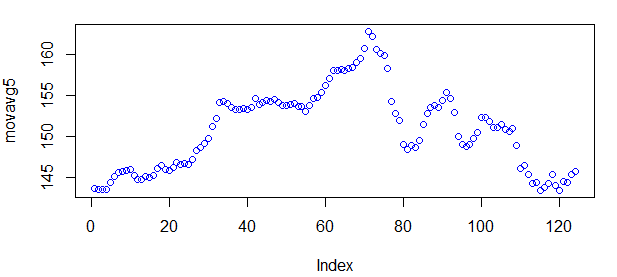


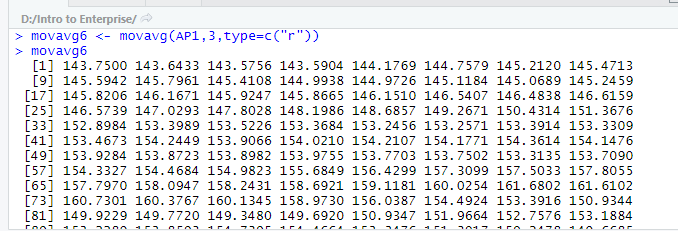


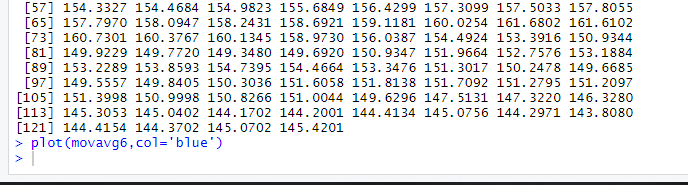


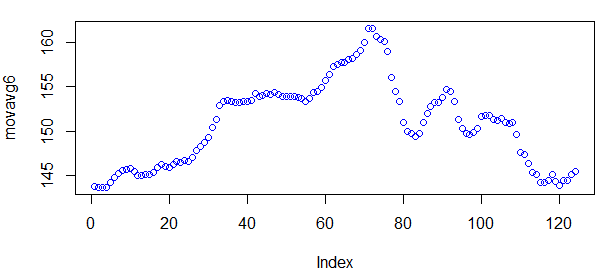






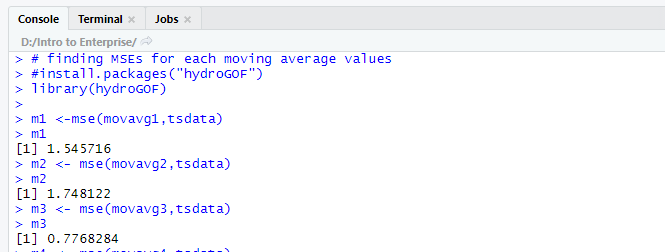


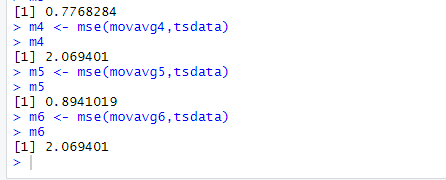




From the above patterns we can conclude that the stock prices are showing same pattern initially and then showing the difference for each of the above MA values.

Now further to be very precise about the results let’s calculate the MSE values for each



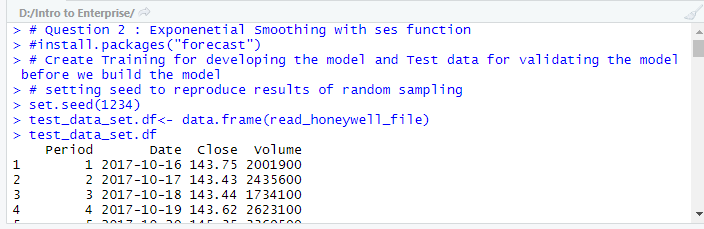


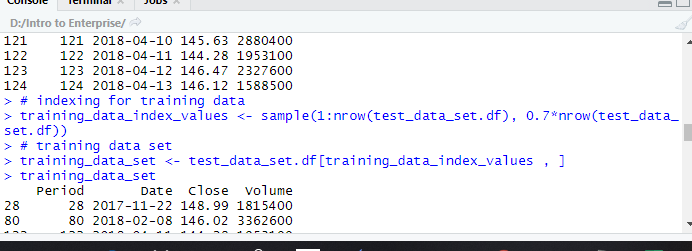
Since bigger MSE values implies that the information esteems are scattered generally around its focal minute (mean), and a littler MSE implies in any case and it is certainly the liked or potentially wanted decision as it shows that your information esteems are scattered near its focal minute (mean); which is typically incredible. And from the above results we can see that the smallest value belongs to 3rd one so this works better than the others i.e the one with type “w”.

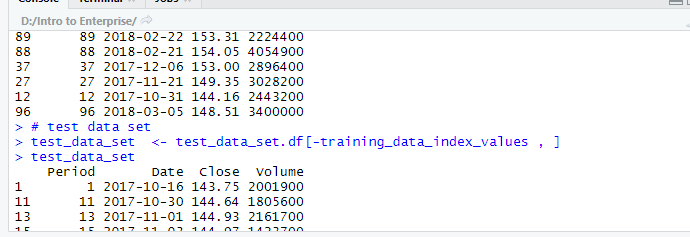
1. In this we have to apply exponential smoothing forecast algorithm on the data set for the same date and we have been provided with the different values of smoothing parameter for each iteration. And here is the logic for the same. I have used “ses()” for the same and the values of alpha i.e smoothing parameter are provided in the problem statement as 0.15,0.35,0.55 and 0.75 respectively. Before building the exponential smoothing time series model we need to divide the data set into training and test data sets first and then we need to further do the calculations on training data and test data sets and finally then compare the results and accuracy for each respectively.

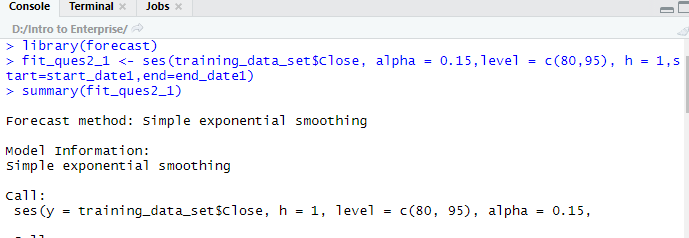
Following is the step by step logic for the same:

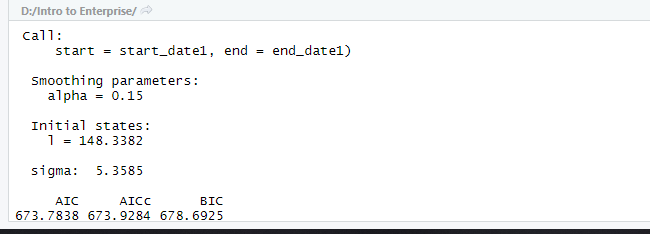
First I have divided the data set into training and test data sets i.e 70 percent for training and 30 for testing data set.

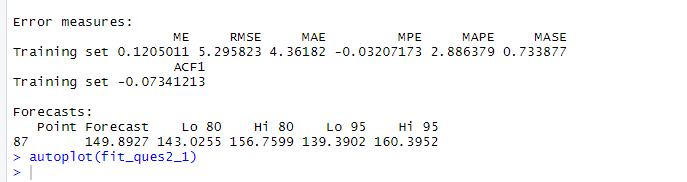


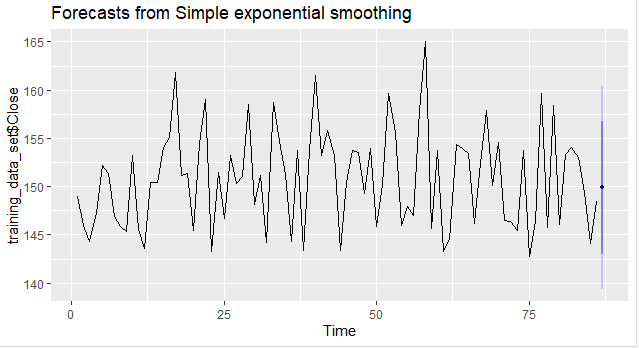


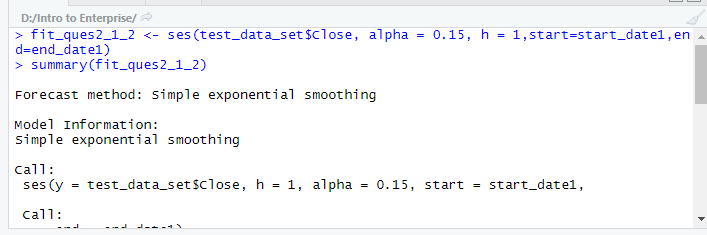


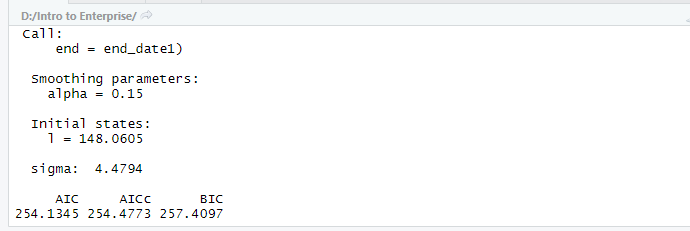


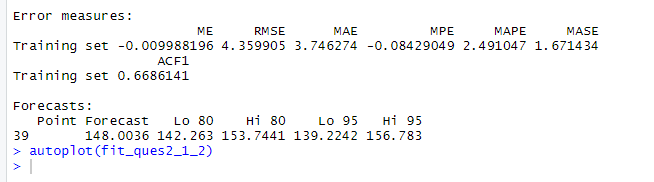


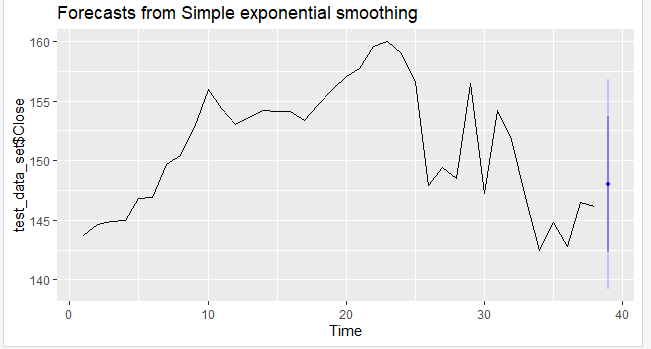






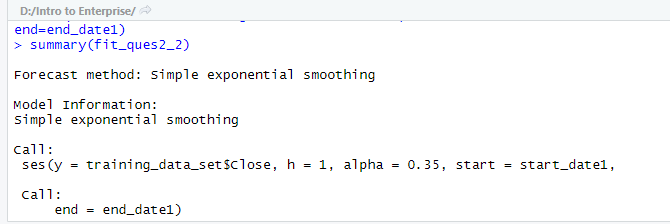


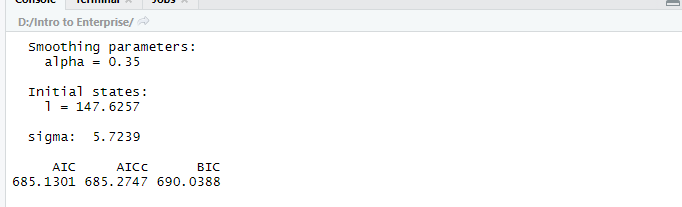


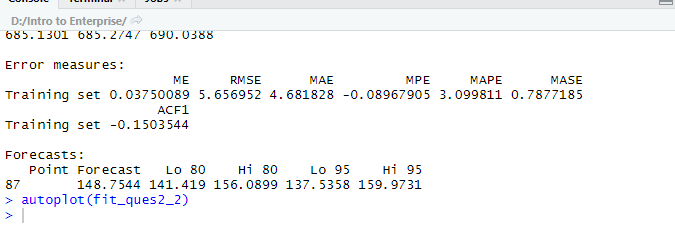


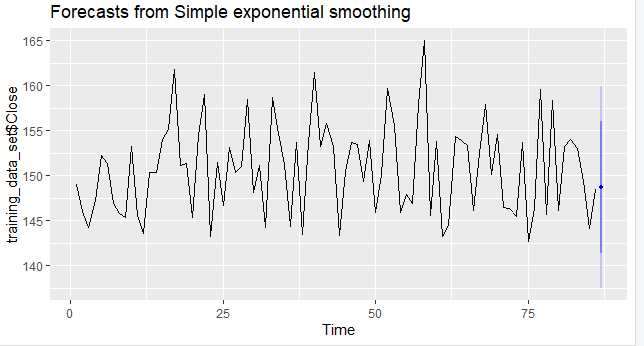
From the above results we can see that there is a very small difference between the MSE values and the graph is also showing a similar trend only the prices differ for very few values.

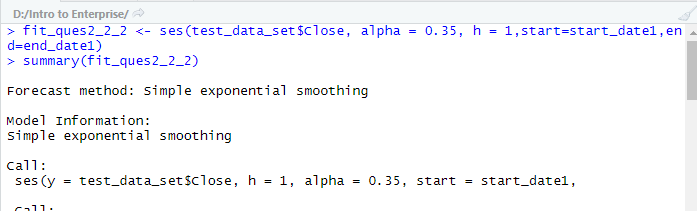
Similarly, I am plotting the results for each of the alpha values provided to us.

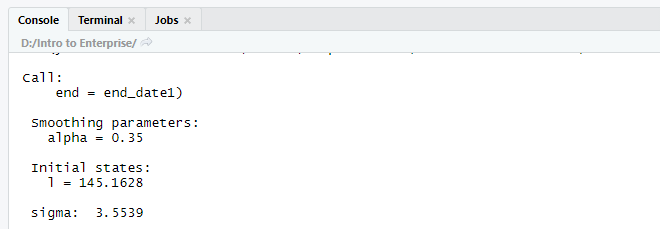


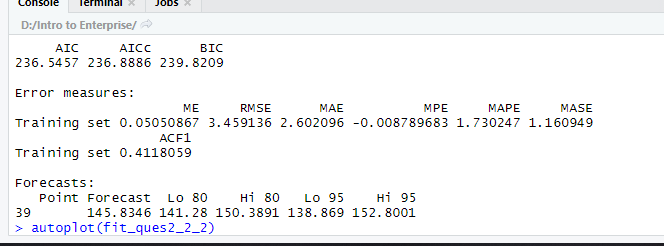


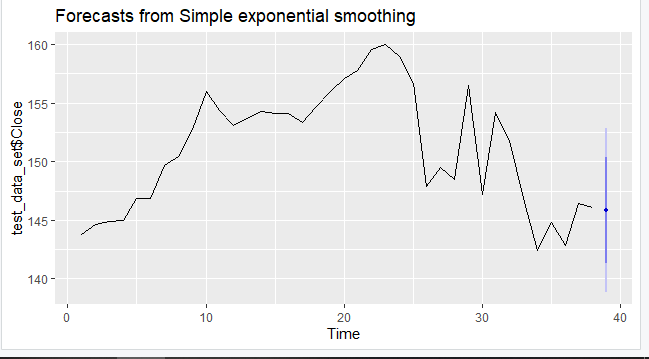


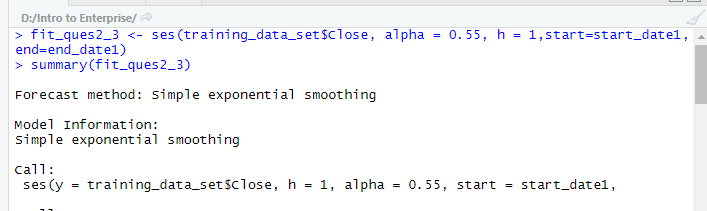


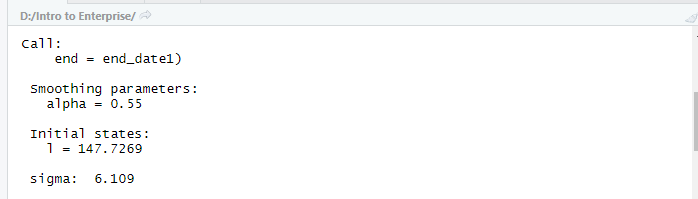


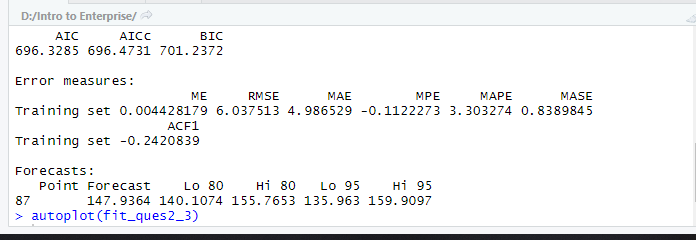


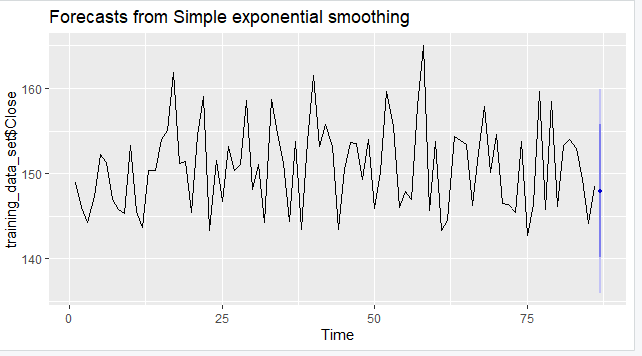


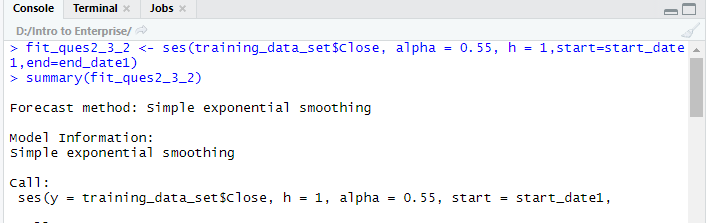


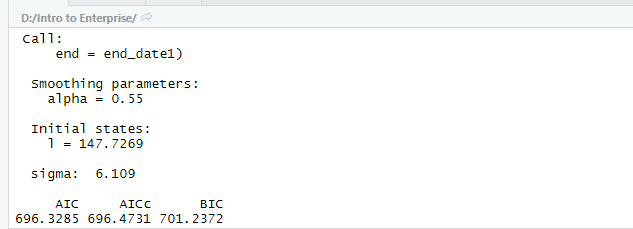


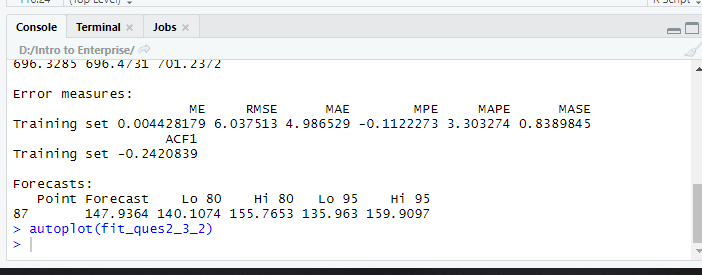


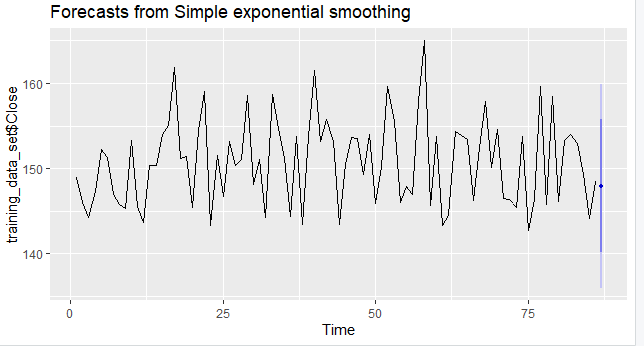


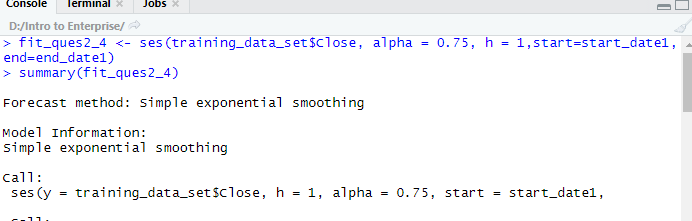


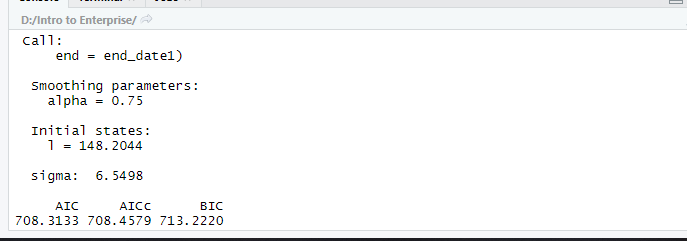


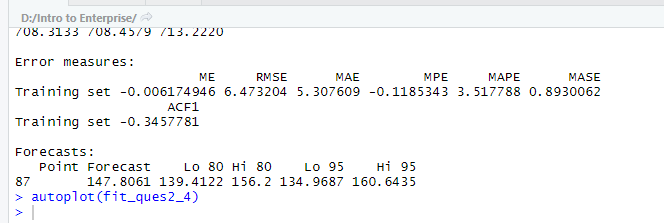


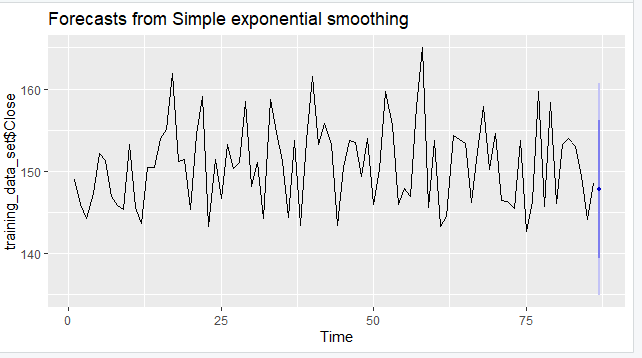


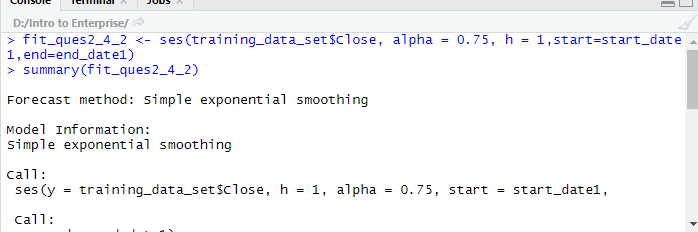


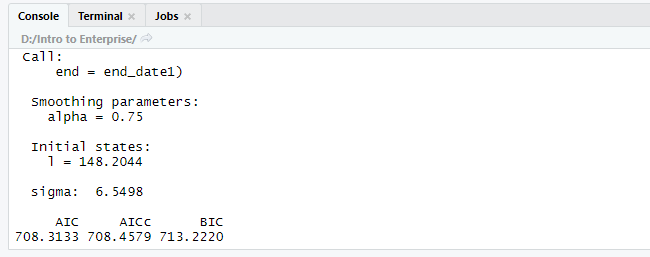


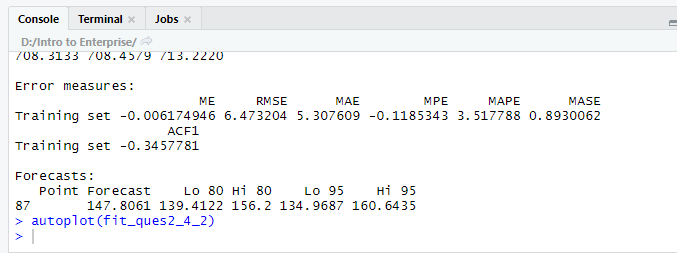


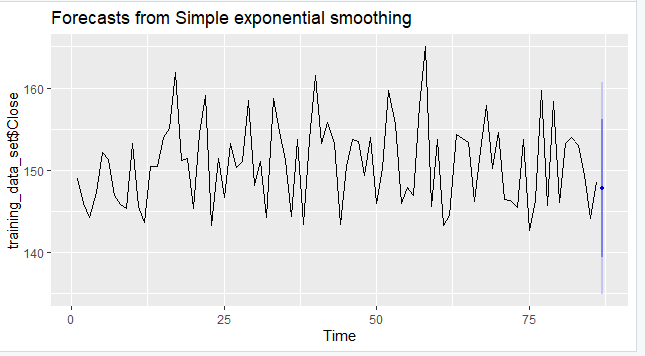




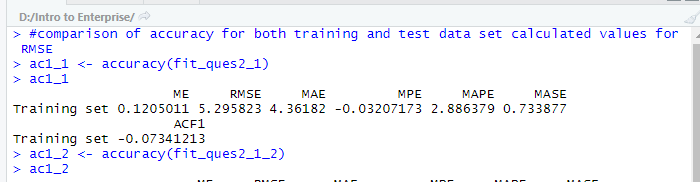


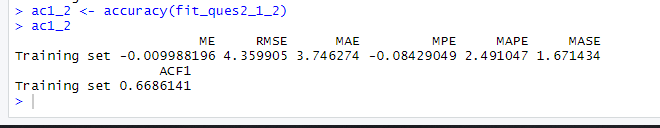


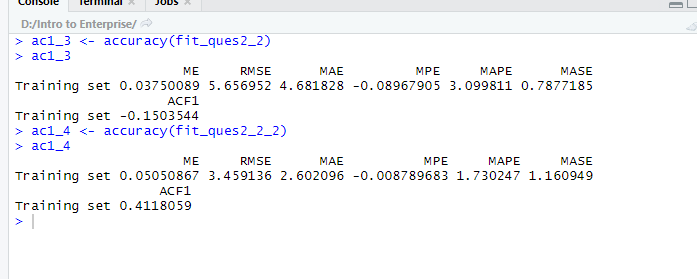


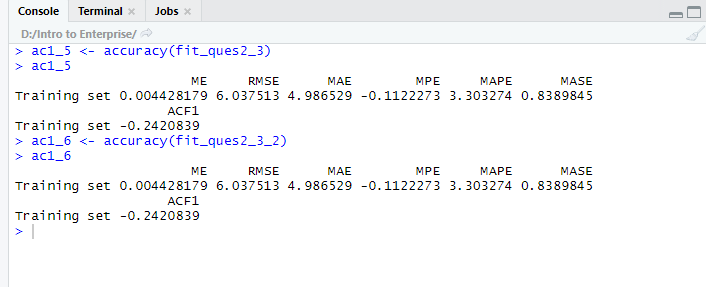


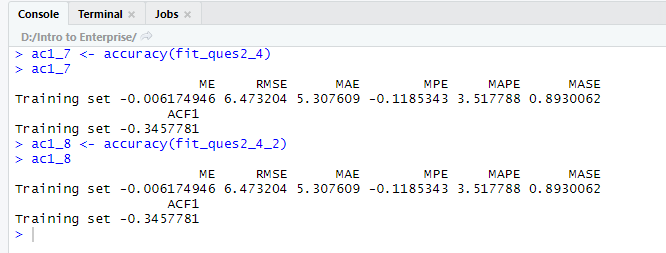
Now after calculating the error measures for all the provided alpha values for both training and test data sets now we will find the accuracy for each of them and then compare the values





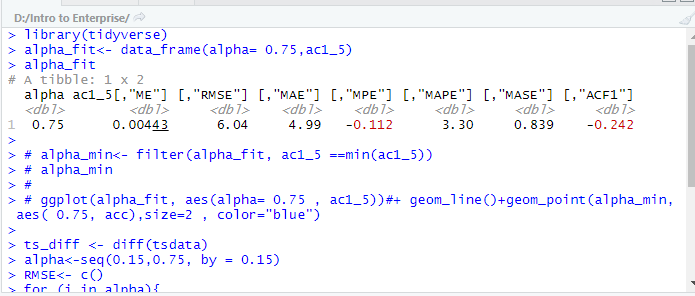


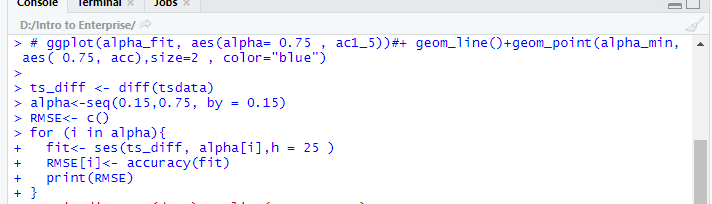




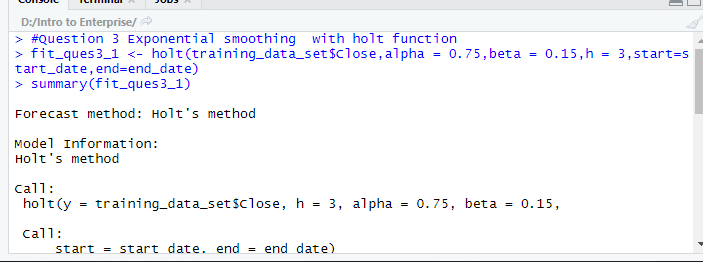
From the above results we can see that the value of alpha=0.15 yielded the best results as the RMSE is the lowest for this value of alpha. We can also do the reverse engineering in order to validate the same i.e based on the RMSE values we can calculate the alpha values and whichever value of RMSE will provide the approximate value of alpha is the best one.

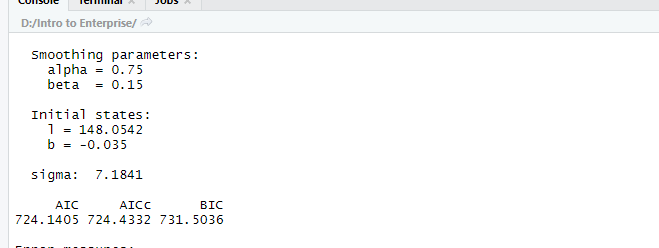
Following is the logic for the same:

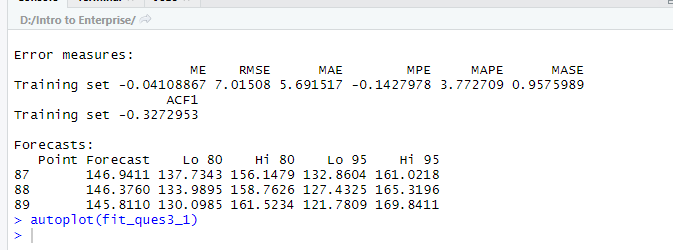


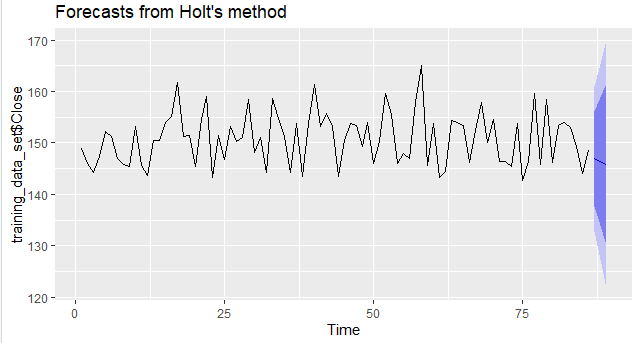


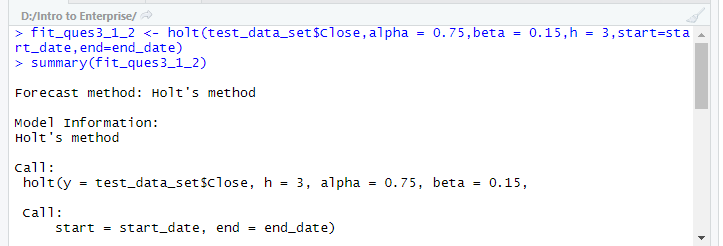
1. In this problem we have to again apply the exponential smoothing forecast with the provided with a single value of smoothing parameter i.e alpha which is 0.75 but different values of trend parameter for each iteration successively. Here I have used “holt()” for calculating the same. Here is the logic for the same:

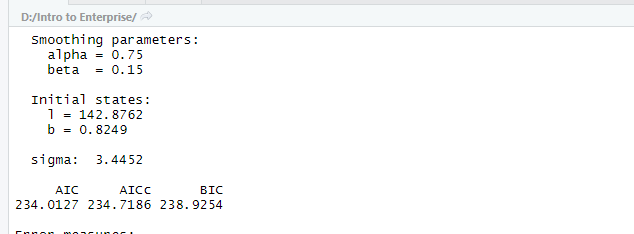


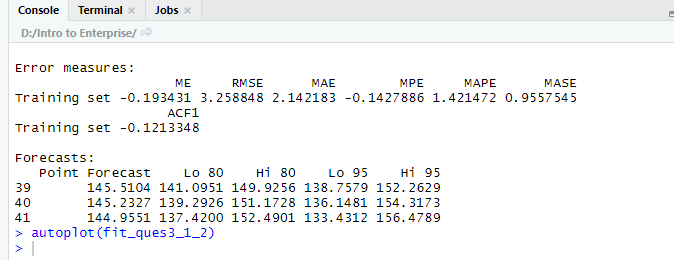


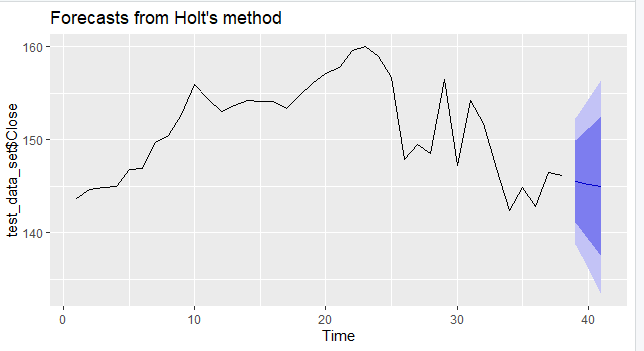


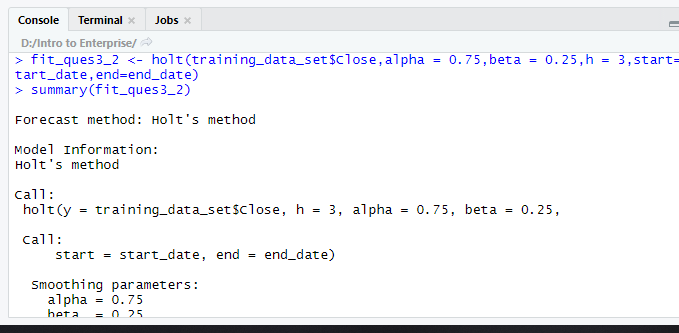


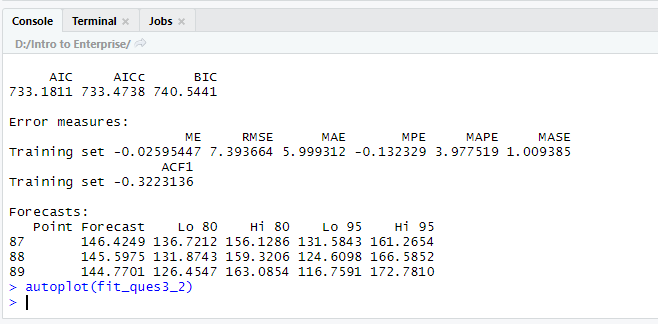


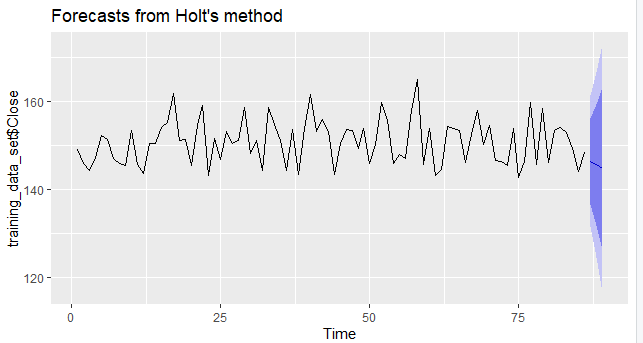


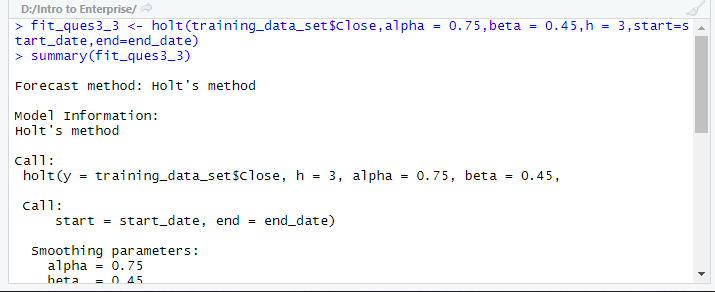


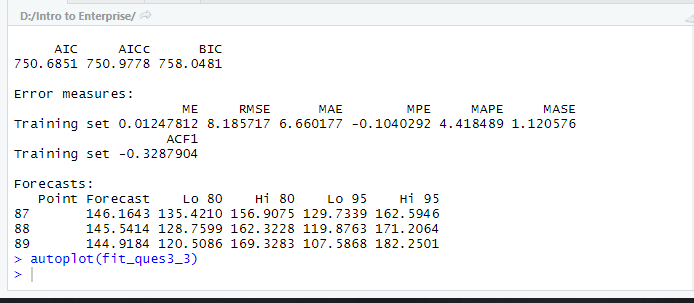


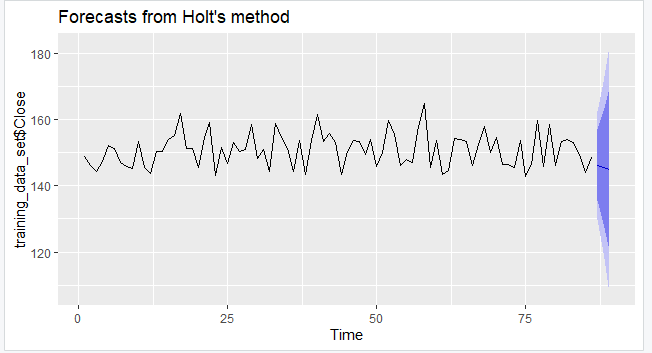


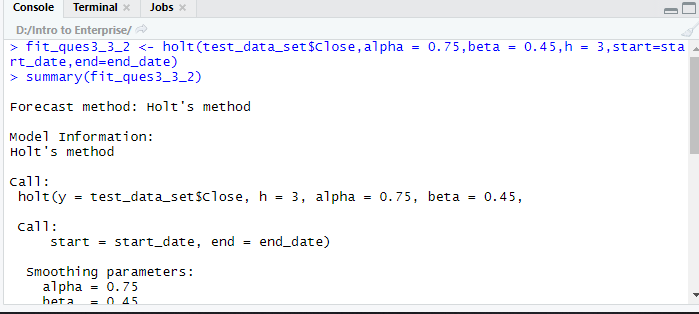


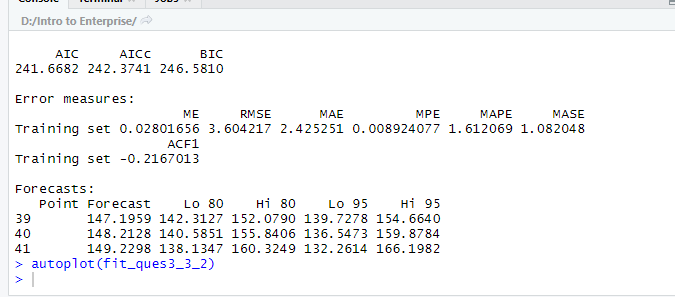


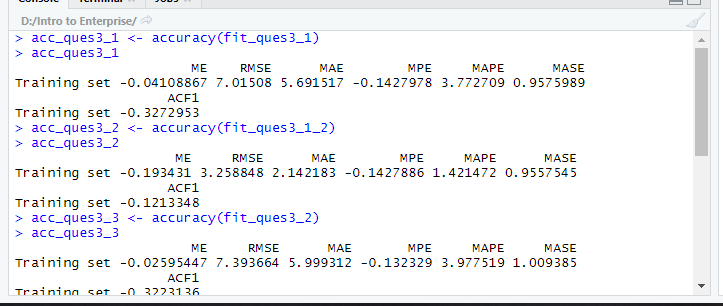


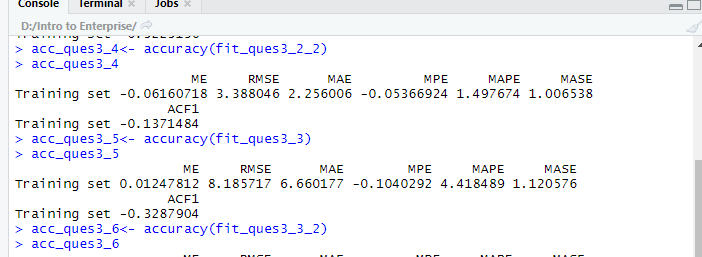


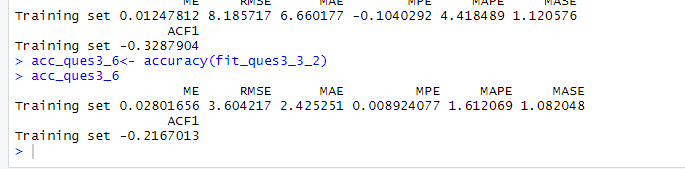












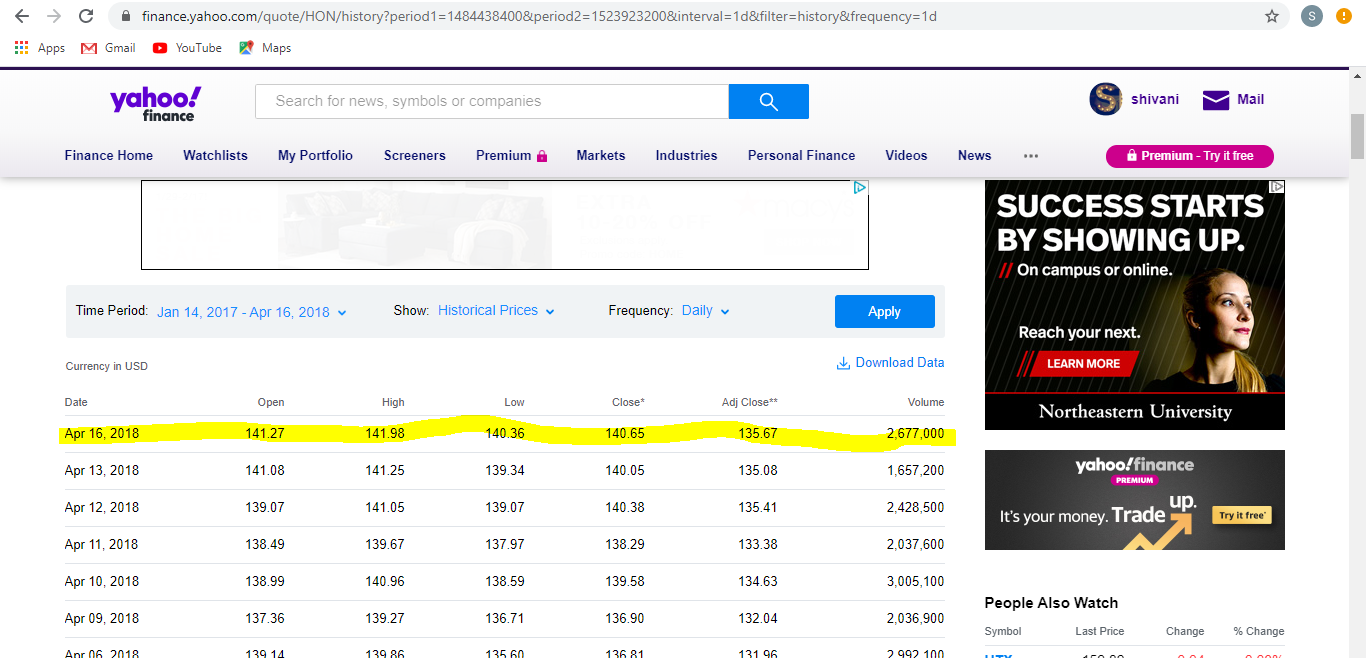
From the above results we can observe that the RMSE values are lowest for beta=0.15 so it is providing the most accurate forecast for the stock prices. Like in previous problem I have mentioned that we can do the reverse engineering i.e by passing the values of RMSE we can calculate the beta values and the one with most approximate value will be the correct one.

1. Final problem statement I would like to mention in the conclusion section as this one is whole learning of this project.

**Conclusion**

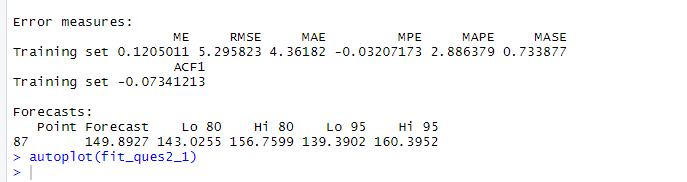
I tried to find out the actual data of the stock market for Honeywell for 16 april 2018 from the historical data available on the website “<https://finance.yahoo.com/quote/HON/history?period1=1484438400&period2=1523923200&interval=1d&filter=history&frequency=1d> “

And here is the data available for the same on website:

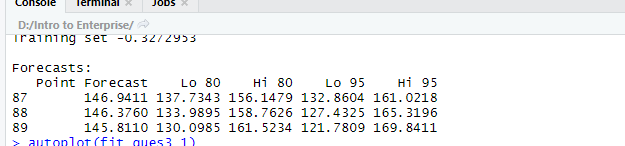


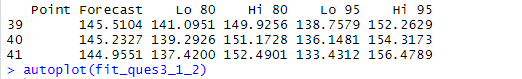
And the results that I have obtained from the applied time series algorithms are approximately nearby. Let’s discuss one by one:

1. From moving average the values predicted are 143.75 which is very close approximation to the actual value i.e 141
2. Now coming to the results of exponential smoothing with ses(). From the plots we have created we can conclude that the value is lying between 140-145 and for the most significant value of alpha i.e 0.15 the predicted value are as follows which is also nearly close to approximation if we see the lowest value.



1. Now coming to final value i.e the exponential smoothing with beta values so the forecasted value is as follows:





Which is also nearby close.

So, from the above discussion we have observed that the exponential smoothing with holt function provided the most accurate results for the forecast.

# References

1. Prabhakaran, Selva. “Eval(ez\_write\_tag([[728,90],'r\_statistics\_co-Box-3','ezslot\_4',109,'0','0']));Time Series Analysis.” *Time Series Analysis With R*, r-statistics.co/Time-Series-Analysis-With-R.
2. Kabacoff, Robert. “Missing Data.” *Quick-R: Missing Data*, [www.statmethods.net/input/missingdata.html](http://www.statmethods.net/input/missingdata.html).
3. “Moving Averages.” *Moving Averages · AFIT Data Science Lab R Programming Guide*, afit-r.github.io/ts\_moving\_averages.
4. “Forecasting: Principles and Practice.” *8.4 Moving Average Models*, otexts.com/fpp2/MA.html.
5. “Time Series Analysis Using R.” *DataCamp Community*, [www.datacamp.com/community/tutorials/time-series-r](http://www.datacamp.com/community/tutorials/time-series-r).
6. Seo, Jae Duk. “Trend, Seasonality, Moving Average, Auto Regressive Model : My Journey to Time Series Data with...” *Medium*, Towards Data Science, 5 June 2018, towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b.
7. “Forecasting: Principles and Practice.” *6.2 Moving Averages*, otexts.com/fpp2/moving-averages.html.
8. Brownlee, Jason. “A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python.” *Machine Learning Mastery*, 21 Aug. 2019, machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/.
9. Stephanie. “Exponential Smoothing: Definition of Simple, Double and Triple.” *Statistics How To*, 2 June 2019, [www.statisticshowto.datasciencecentral.com/exponential-smoothing/](http://www.statisticshowto.datasciencecentral.com/exponential-smoothing/).
10. <https://s3.us-east-1.amazonaws.com/blackboard.learn.xythos.prod/5a3148150d016/19571386?response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27Week%25203%2520R%2520Example%2520-%2520Jupyter%2520Notebook.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20200127T211406Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=AKIAIL7WQYDOOHAZJGWQ%2F20200127%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=037f418498a37a6f56495483dcfff55c0ff11ec0933d8a59970439ce20c67fd1>
11. <https://stackoverflow.com/questions/39732029/finding-mean-squared-error>