PROJECT REPORT

Aim: To forecast closing stock prices of Google through time series analysis via ARIMA model.

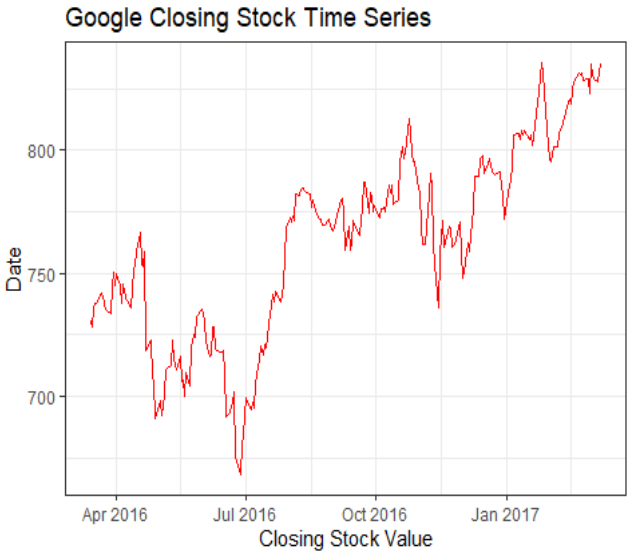
Exploring the dataset

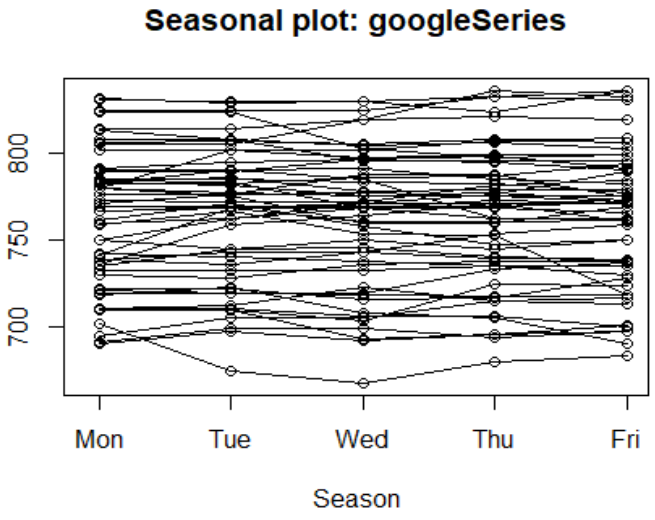
We have a dataset[6] that contains the stock price values for Google. It consists of 6 features: Date, Open, High, Low, Close, Volume. All the features except Date are continuous values that state the stock prices at various stages of a trading day. Date is a factor variable that holds dates from 14th March 2016 to 8th March 2017. It is converted into a Date variable with "%Y-%m-%d" format. We have 249 observations corresponding to the trading days between the dates. We have extracted only two columns: Date and Close in a new data frame (for easier processing) to perform time-series analysis on it. There are no missing values in the dataset and no duplicate values which is checked using *googledata[duplicated(googledata)].*

```
## 'data.frame':    249 obs. of  2 variables:
##  $ Date : Date, format: "2016-03-14" "2016-03-15" ...
##  $ Close: num  730 728 736 738 738 ...
```

Data Pre-processing

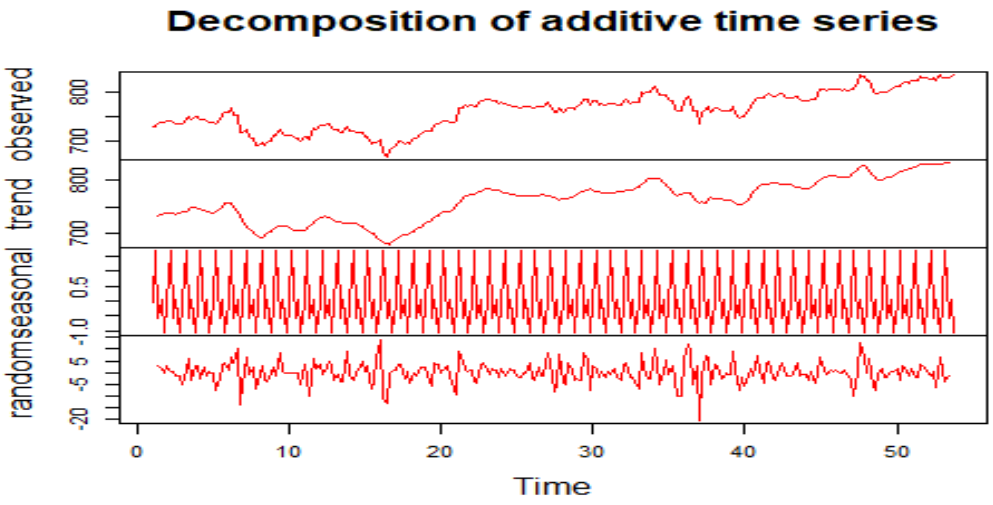
Many dates are missing in the dataset without any regularization. Therefore, data must be pre-processed to regularize it. Saturdays & Sundays are not present in the dataset as no trade happens on these days. However, there is no information about the holidays. These dates must be examined to have a frequency (equal intervals) in the dataset for seasonality. The data frame is merged with another data frame holding all calendar days of the period under investigation created using *seq.Date()* function. The missing days (NA values) are examined and imputed according to the Last Observation Carried Forward (LOCF) using *na.locf()* function. LOCF replaces each missing value with the most recent present value before it. Furthermore, a time series with a frequency of 20 to represent monthly data is created. [Each week has 5 trading days. Hence a month will have 20 trading days!]





The closing stock value appears to have a random pattern. The closing pattern appears to be increasing overall with time. There was a large drop in the middle of the year around June to September followed by an increase in the successive time of the year. The season plot of all weeks is put together for easy comparison using *seasonplot()* in the forecast package. All the weeks of dataset are put together against each other. On the x-axis we have five working days starting on Monday and ending on Friday. Y-axis contains respective closing prices in U.S dollars. The highest price for all these values was either Thursday or Friday. Both values seemed to be very close in that regard. As the lowest values here don't happen to be either Wednesday or Tuesday. Another interesting thing is the baseline which is nothing else than the mean of the corresponding series. We see that mean is lowest on Wednesday. The difference is not large, but in finance this can be a significant edge.

The series neither appear to be multiplicative nor additive. We can estimate the two components of a non-seasonal data, Trend and an Irregular component through decomposition of its additive series. The trend component could even by examined using *SMA().*



The trend component seems to be random with an increasing effect as stated earlier with a decrease near 10. The irregular component is depicted via the random component in the decomposition. Since we have non-seasonal data we will use the Non-Seasonal ARIMA model: ARIMA(p,d,q) to forecast the values.

1) p = It denotes periods to lag. It is the order of the auto regression part & can be calculated using the Partial Autocorrelation plot(PACF).

2) d = In an ARIMA model we transform a time series into stationary one (series without trend or seasonality) using differencing. d refers to the number of differencing transformations required by the time series to get stationary.

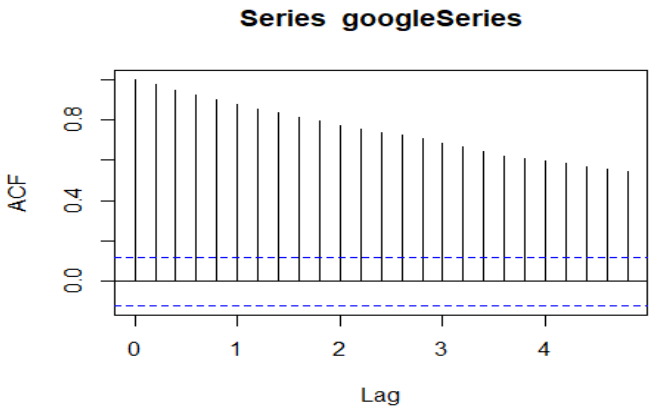
3) q = This variable denotes the lag of the error component, where the error component is a part of the time series not explained by trend or seasonality. It is the order of the moving average part & can be calculated using the Autocorrelation plot(ACF).

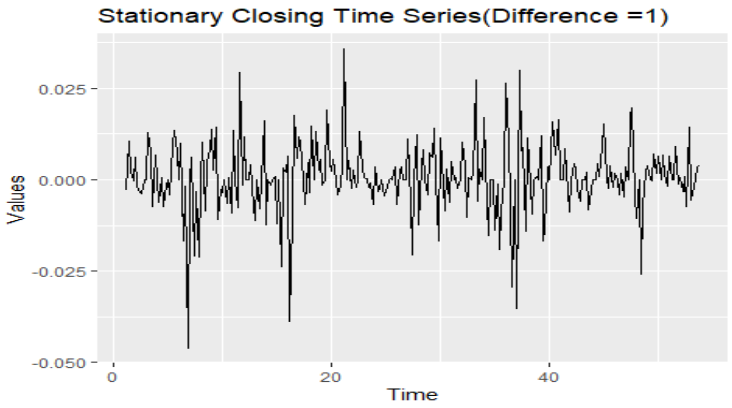
Assumptions of Time Series Analysis using ARIMA(p,d,q)

1) Univariate Data: We have only a single feature in the data, “Close” on which we want to perform time series analysis with its past values. Hence, the assumption is met.

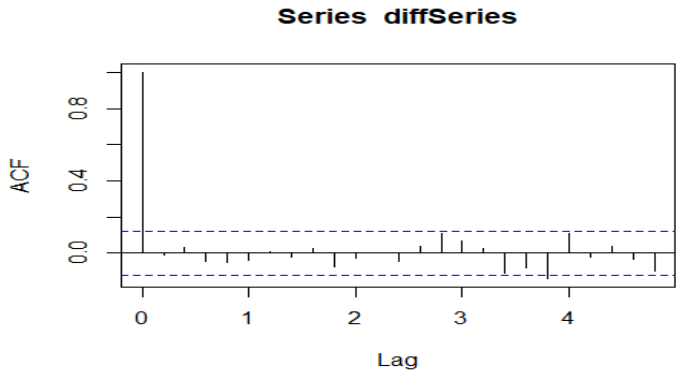
2) Stationary Data: Through visual inspection the data does not seem to be stationary as mean & variance are not constant over time. The stationarity of the data can further be tested using the Dickey-Fuller test and Autocorrelation Plot. The test analysis yields a p-value of 0.2382(>0) is not significant. Thus, the data is not stationary. Differencing of log transformation on the series with difference = 1 is performed. Transformations such as logarithms help to stabilize the variance of a time series. Differencing helps stabilize the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality. The differenced time series on the Dickey-Fuller test yields p-value of 0.01~0, thus the null hypothesis (series is not stationary) is rejected and the series is now stationary. The d parameter in the ARIMA model for differencing required to get a stationary series is set to 1. Therefore, the assumption is met with d=1.

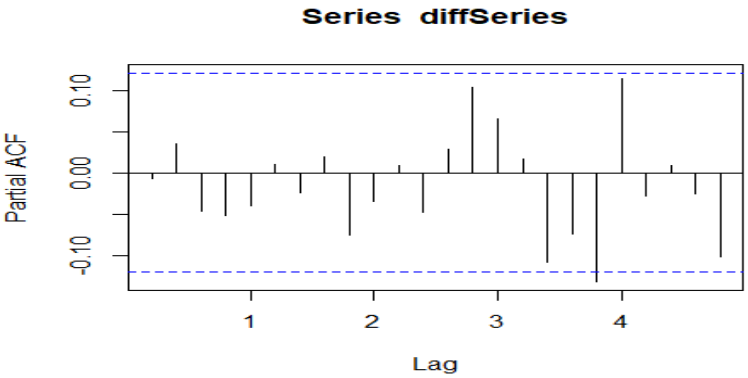
```
## Augmented Dickey-Fuller Test
## data:  googleSeries
## Dickey-Fuller = -2.8018, Lag order = 0, p-value = 0.2382
## alternative hypothesis: stationary
```






```
##  Augmented Dickey-Fuller Test
## data:  diffSeries
## Dickey-Fuller = -16.256, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```





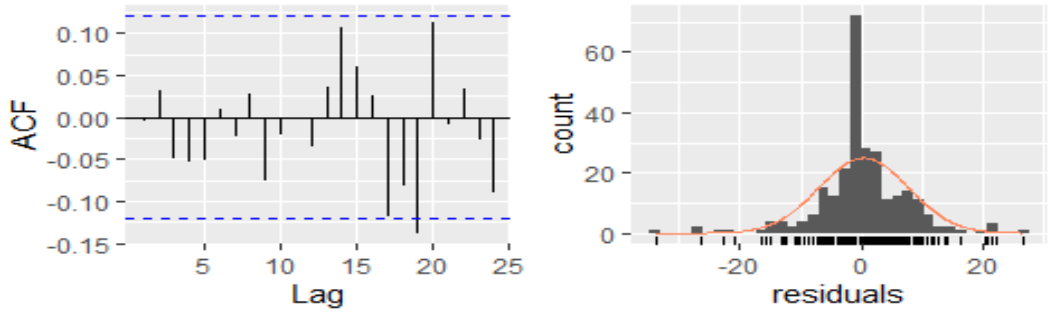

```
##  Fitting models using approximations to speed things up...
##  ARIMA(2,1,2)(1,0,1)[5] with drift         : Inf
##  ARIMA(0,1,0)           with drift         : 1804.589
##  ARIMA(1,1,0)(1,0,0)[5] with drift         : 1812.515
##  ARIMA(0,1,1)(0,0,1)[5] with drift         : 1807.951
##  ARIMA(0,1,0)                              : 1803.322
##  ARIMA(0,1,0)(1,0,0)[5] with drift         : 1809.5
##  ARIMA(0,1,0)(0,0,1)[5] with drift         : 1805.898
##  ARIMA(0,1,0)(1,0,1)[5] with drift         : 1811.562
##  ARIMA(1,1,0)           with drift         : 1807.513
##  ARIMA(0,1,1)           with drift         : 1806.632
##  ARIMA(1,1,1)           with drift         : 1808.771
##  Now re-fitting the best model(s) without approximations...
##  ARIMA(0,1,0)                              : 1807.323
##  Best model: ARIMA(0,1,0)

## Series: googleSeries
## ARIMA(0,1,0)
## sigma^2 estimated as 54.62:  log likelihood=-902.65
## AIC=1807.31   AICc=1807.32   BIC=1810.88
```

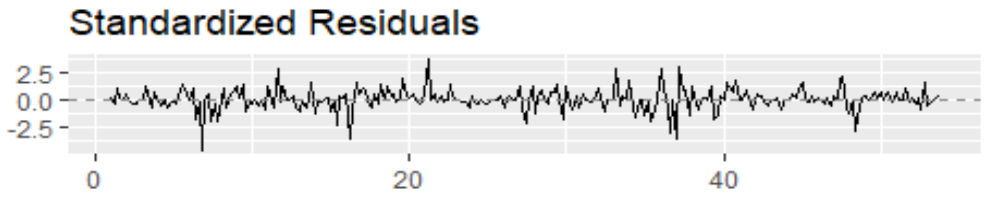
The result matches the one which we estimated. Our candidate model is selected as the best model to fit the time series because it gives the least Akaike's Information Criterion(AIC) value of 1807.31 among the rest of the configurations. Since we got only the non-seasonal component in the model, therefore our series is non-seasonal.

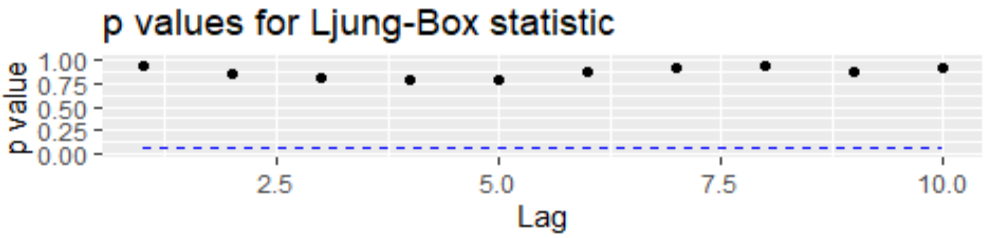
Model Residual Analysis:

The residuals from the model need to be white noise i.e. there should be no autocorrelation in the residuals. The plots are made using *qqnorm(), checkresiduals(), ggtsdiag(), and autoplot()* functions in R.

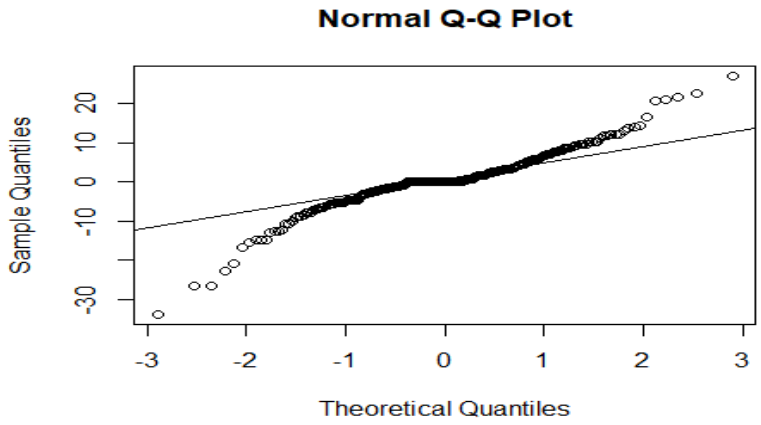
```
## Ljung-Box test
## data:  Residuals from ARIMA(0,1,0)
## Q* = 4.5121, df = 10, p-value = 0.9213
## Model df: 0.   Total lags used: 10
```







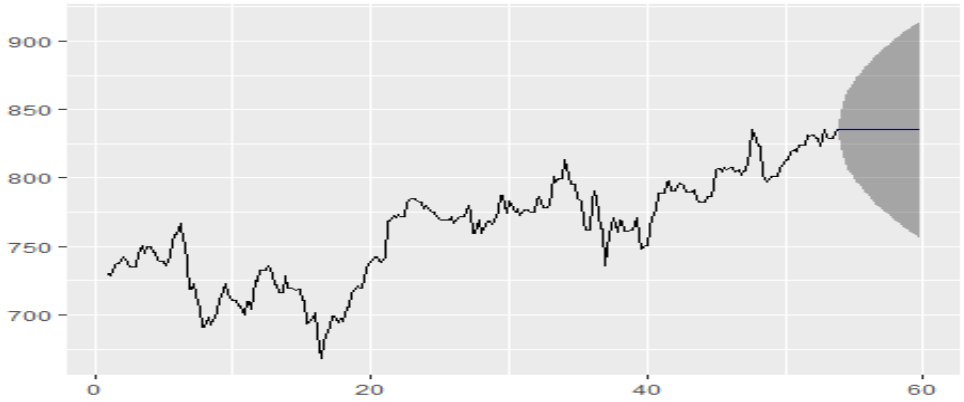
1) Plotting Standardized Residuals: The residuals are centered around 0. Therefore, the mean is close to zero. The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data, apart from one outlier outside -3, and therefore the residual variance can be treated as constant.
2) Box-Ljung Test: The test reveals the autocorrelation in the series. The null hypothesis is that the residuals from the ARIMA model do not have autocorrelation at a 95% significance level. The test reveals a p-value of 0.9213(>0.05) is non-significant. Moreover, the p-values for the Ljung-Box Q test all are well above 0.05, indicating "non-significance." Therefore, there is no autocorrelation in the residuals. So, our residuals are white noise.
3) ACF of Residuals: The ACF of the residuals shows no significant autocorrelations. Hence, no autocorrelation in residuals.



4) Q-Q plot & Histogram: Both these graphs depict the normality of the residuals. The residuals are centered around 0 and are quite normal. There is a tail on the left side even if the outlier is removed. Consequently, forecasts will probably be quite good, but prediction intervals that are computed assuming a normal distribution may be inaccurate.

Forecasting:

As all the graphs are in support of the assumption that there is no pattern of autocorrelation in the residuals, therefore, our model ARIMA(0,1,0) is a good fit and we can use it to forecast closing stock values using *forecast()* function in R.




```
##                  MPE      MAPE
## Training set 0.04618084 0.6335156
```

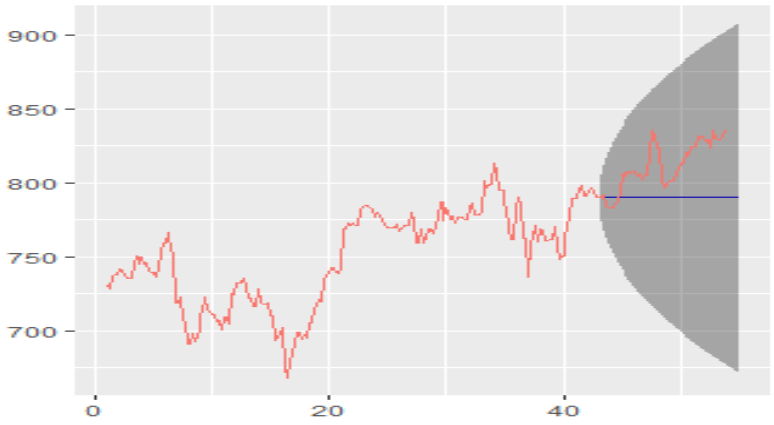
We have a straight line close to the last observation. There is no seasonality in the forecast as seen in the ARIMA model. The forecast is slightly above the last observation which was also higher than the previous one. The model considers several last observations in making the forecast. The grey area represents the 95% confidence interval of the forecast. It goes wider with time. The mean average percentage error (MAPE) of the model calculated using *accuracy()* function in R is quite small 0.63% indicating a good model. Therefore, the closing stock values are likely to be higher than the previous ones. Therefore, it is a definite profit to trade with Google.

Validation via Train and Test sets:

We divide the series into a train and test series using *subset().* We split the series in 80:20 ratio of train:test. The forecast of the training was compared with the actual trend in the series or the values if the test set. The results seem to be quite good. The actual trend is within the confidence interval computed by the forecast. Therefore, our model was validated using the information we had & it proved to be a good model.

```
##       Point Forecast    Lo 95    Hi 95
## 43.20         789.91 774.6998 805.1202
## 43.40         789.91 768.3995 811.4205
. . .

## 54.60         789.91 674.0724 905.7476
## 54.80         789.91 673.0780 906.7420
## 55.00         789.91 672.0921 907.7279
```

Forecasting on trainingset:



Conclusion:

1) The time series analysis can be used to forecast any future values for any univariate data. Non- seasonal ARIMA(p,d,q) model was used to forecast the non-seasonal time series of Google stock from March 2016 to March 2017. Stationarity holds an important aspect in time series & the tests seems to be satisfied with these assumptions on our data
2) The diagnostic tests on residuals, Box Ljung, ACF plots, QQ plots for the model, and model validation on train & test sets revealed good results.
3) The stock value predicted for future months in 2017 was also compared with the actual values online using "finance.yahoo"[4] and results were similar. Therefore, successful forecasting was implemented.
4) Since, forecast predicted an overall increase in the closing values for the next 7 weeks, there will be an overall profit in pursuing trade with Google. Generally, trading with Google is good for on Thursdays & Fridays. Its Wednesday market goes down a bit.
5) Future scope of the project can be moving on to more complicated time-series datasets, models and analyzing it & making significant comparisons and conclusions from it.

References:

1. https://otexts.com/fpp2/
2. https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/
3. http://rstudio-pubs-static.s3.amazonaws.com/311446_08b00d63cc794e158b1f4763eb70d43a.html
4. https://finance.yahoo.com/quote/GOOGL/history?period1=1490227200&period2=1492905600&interval=1d&filter=history&frequency=1d
5. https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html
6. https://www.kaggle.com/jamesbasker/goog-ticker-stock-data