AN EMPIRICAL ANALYSIS OF STOCK MARKET PRICE PREDICTION USING ARIMA.

AT

Xavier Institute of Management & Entrepreneurship

SUBMITTED BY

RAMPRASAD MOHAN

Roll no: **C03/061**

Under supervision of

Dr. Asraar Ahmed

A REPORT SUBMITTED IN PARTIAL

FULFILMENT OF THE REQUIREMENT OF PGDM (2019-21)

•Bangalore
•Chennai
•Kochi
Shaping Future Leaders

XAVIER INSTITUTE OF MANAGEMENT AND

ENTREPRENEURSHIP, CHENNAI

ACKNOWLEDGEMENT

I thank the God Almighty for giving me the courage and wisdom to take up the project and complete it successfully.

The help that I received from my mentors in all aspects is overwhelming and made me learn and accomplish the work entrusted to me with full satisfaction and complete my research project with ease and good learning. I am grateful to all the people in this organization for their support and for accommodating me in all their activities. I thank Prof. J. Philip, Chairman and Founder, XIME who laid the foundations of this initiative.

I would also like to thank Dr. David Jawahar, Director, XIME Chennai and Dr. Suresh Kumar, Dean (Academics), XIME Chennai for his support throughout the program. Thanks to all the other faculty members of XIME Chennai who made material efforts in preparing us for this journey and for all the support they provided us till the end.

My acknowledgement would be incomplete without mentioning the kind support and guidance provided by the faculty guide, Dr. Asraar Ahmed who gave me valuable inputs on my project. I express my profound gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout this study and through the process of researching and writing this research project. This accomplishment would not have been possible without them. I express my deep thanks and gratitude to each and every person who helped me and guided me in completing the project.

Xavier Institute of Management and Entrepreneurship

Chennai

Dr. Asraar Ahmed — Professor & Supervisor

Dr. Suresh Kumar — Dean (Academic)

CERTIFICATE

This is to certify that the Research project titled "An empirical analysis of stock market price prediction using ARIMA" at XIME was submitted by Mr. Ramprasad Mohan, Roll No.C03/061, Batch III, under the supervision of Dr. Asraar Ahmed. This has not been submitted to any other University or Institution for the award of any degree/diploma/certificate.

Date:

Place:

Signature of the Supervisor

Signature of the Dean

Mr. Ramprasad Mohan
Roll No: C03/061
Batch III
Xavier Institute of Management and Entrepreneurship
Chennai

DECLARATION

I hereby declare that this Project Report on Summer Internship Research Project titled "An empirical analysis of stock market price prediction using ARIMA" submitted to Xavier Institute of Management and Entrepreneurship, Chennai is a bonafide work carried by me under the guidance of Dr. Asraar Ahmed. This has not been submitted earlier to any other University or Institution for the award of any degree/diploma/certificate or published any time before.

Date:

Place:

Signature of the Student

EXECUTIVE SUMMARY

The research project titled ‘An empirical analysis of stock market price prediction using ARIMA’ is about how stock market or index of stock exchanges can be predicted with a predictive modelling algorithm like ARIMA using data science tools like R & R Studio. This project involves building the ARIMA model and evaluating the model’s accuracy and later forecasting the future time series values using the model build for stock indices.

In the past, several machine learning models and techniques had been developed for stock price prediction. Among them are artificial neural networks (ANN) model which are very popular due to its ability to learn patterns from data and predict/ infer solution from unknown data. ARIMA models are known to be robust and efficient in financial time series forecasting especially short-term prediction than even the most popular ANN techniques. It has been extensively used in field of economics and finance. ARIMA stands for **A**uto **R**egressive **I**ntegrated **M**oving **A**verage. ARIMA forecasts short-term future values based on the increasing or decreasing change in historical values, or inertia.

This project “An Empirical Study of Stock Price and Prediction” helped me in applying various tools and techniques and understanding its dependencies of the various methods. Through these researches I came to know about the positive and strong relationships using autocorrelation and using ARIMA, and also I came to know about various techniques that can be used to improve the current model using ANN, HWES, SARIMA, LSTM and SVM.

.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER: I
INTRODUCTION

A time series is a univariate sequence of numerical data points in successive order. In stock investment portfolio, a time series tracks the movement of the chosen data points, such as a share's opening and closing price, over a specified period of time with data points recorded at regular intervals. There is no minimum or maximum amount of time that must be included, allowing the data to be gathered in a way that provides the most information being sought after by the investor or analyst examining the stock price activity.

What is Time Series?

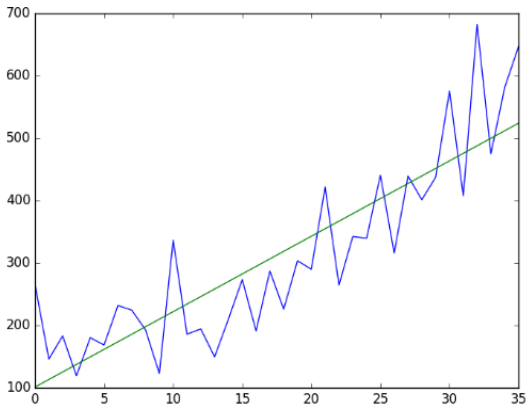
A time series can be taken on any continuous variable that changes over time (ratio scale). In stock trading, it is common to use a time series to track the price of a company's share over time. This can be tracked over the short term, such as the price of a share over the course/intraday of a business, or the long term, such as the price of a share at close on the last day of every month over the course of five years.

Time Series Analysis

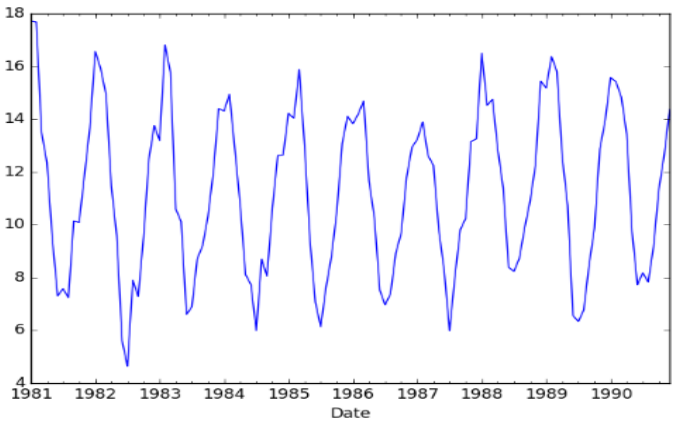
Time series analysis can be useful to see how a given asset, debentures, or economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

Time series analysis has the features to help us understand the underlying factors that lead to a specific trend in time series data points and thus help us predict data points. The first step towards time series analysis is to check if the time series data is stationary. The reasons behind non-stationary data are trend, seasonality and heteroscedasticity.

Trend: It is the general systematic linear or (most often) nonlinear component that changes over time and does not repeat. For example, the average number of car users is growing over time.



Seasonality: It refers to variations at a particular time-interval that repeats every year. Eg:- people might buy cars in a particular month because of salary increment or festival.

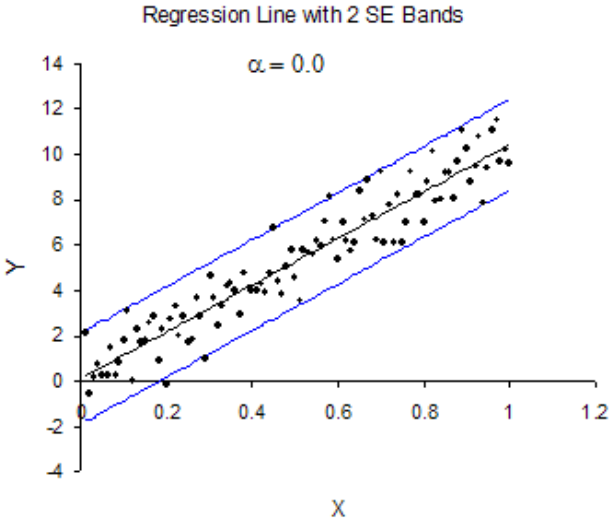


To check if a time series is stationary, the following tests are performed:

Plotting Rolling Statistics: Plots the moving average or variance and check if it varies with time. This is more like a visual representation.

Dickey-Fuller Test: Unlike the first one this a statistical test to check stationarity. In this test, null hypothesis considers time series as non-stationary. Results are the test-statistic (p- value) and some critical-values for different confidence levels. The series is said to be stationary if the null hypothesis gets rejected when the test-statistic becomes less than the critical-value.

Heteroscedasticity: Heteroscedasticity means unequal scatter and in regression analysis, we talk about heteroscedasticity in the context of the residuals or error term. It is a systematic change in the spread of the residuals over the range of measured values. Heteroscedasticity causes problem because ordinary least squares (OLS) regression assumes that all residuals are drawn from a population that has a constant variance.



The main purpose is to reduce these features from the time series by estimating the trend and seasonality in the series. The following techniques can be useful to model or estimate trend and seasonality:

- Aggregation: Considers averages for a time period like month/week.
- Smoothing: Considers rolling averages.
- Polynomial Fitting: Fits a regression model.

According to different problem solving, any of the techniques can be used. After the estimation, trend and seasonality can be reduced by using the following methods:

Differencing: It is the most common way to reduce non-stationary features by taking the difference of observation between a particular instant with a previous instant. Trend and seasonality reduction can be improved by changing the order of differencing.

Decomposing: It separately models the trend and seasonality of the time series and the rest of it is returned so that the residuals can be modeled. After these steps, forecasting techniques can be applied on the non-stationary series. In final step, trend and seasonality constraints are applied back to convert the predicted values into the original scale.

Prediction of time series data can be done using various methods like ARMA, ARIMA, GARCH, etc. But in this report Auto-Regressive Integrated Moving Averages (ARIMA) had been used to predict the price of Sensex Index. But before we understand ARIMA, we need to understand the AR, MA and ARMA methods.

Mostly there are eleven types of classical time series analysis, but in this research project only four are mentioned due to ease of understanding and applicability of the algorithm in R and R Studio.

1. **Auto-regression (AR)**

The auto-regression (AR) method models the next step in the sequence as a linear function of the observations at prior time steps. The notation for the model involves specifying the order of the model p as a parameter to the AR function, e.g. AR (p). For example, AR (1) is a first-order auto-regression model. The method is suitable for univariate time series without trends and seasonality.

R Code using ar (autoregression) function :-

```
        # load lh dataset and use for autoregression
ar(lh)
ar(lh, method = "burg")
ar(lh, method = "ols")
ar(lh, FALSE, 4) # fit ar(4)

(sunspot.ar <- ar(sunspot.year))
predict(sunspot.ar, n.ahead = 25)
## try the other methods too

ar(ts.union(BJsales, BJsales.lead))
## Burg is quite different here, as is OLS (see ar.ols)
ar(ts.union(BJsales, BJsales.lead), method = "burg")
```

2. **Moving Average (MA)**

The moving average (MA) method models the next step in the sequence as a linear function of the residual errors from a mean process at prior time steps. A moving average model is different from calculating the moving average of the time series. The notation for the model involves specifying the order of the model q as a parameter to the MA function, e.g. MA (q). For example, MA (1) is a first-order moving average model. The method is suitable for univariate time series without trend and seasonal components.

R Code:-

```
MAmodel <-  sma(rnorm(118,100,3),order=12,h=18,
                        holdout=TRUE,interval="p")

 # SMA of arbitrary order for moving average
ourModel <- sma(rnorm(118,100,3),h=18,holdout=TRUE,interval="sp")

summary(ourModel)
forecast(ourModel)
plot(forecast(ourModel))
```

3. Autoregressive Moving Average (ARMA)

The Autoregressive Moving Average (ARMA) method models the next step in the sequence as a linear function of the observations and residual errors at prior time steps. It combines both Auto regression (AR) and Moving Average (MA) models. The notation for the model involves specifying the order for the AR (p) and MA (q) models as parameters to an ARMA function, e.g. ARMA (p, q). An ARIMA model can be used to develop AR or MA models. The method is suitable for univariate time series without trend and seasonal components.

R Code:-

```
data(tcm)
r <- diff(tcm10y)
summary(r.arma <- arma(r, order = c(1, 0)))
summary(r.arma <- arma(r, order = c(2, 0)))
summary(r.arma <- arma(r, order = c(0, 1)))
summary(r.arma <- arma(r, order = c(0, 2)))
summary(r.arma <- arma(r, order = c(1, 1)))
plot(r.arma)

data(nino)
s <- nino3.4
summary(s.arma <- arma(s, order=c(20,0)))
 summary(s.arma
 <- arma(s, lag=list(ar=c(1,3,7,10,12,13,16,17,19),ma=NULL)))
acf(residuals(s.arma), na.action=na.remove)
pacf(residuals(s.arma), na.action=na.remove)
summary(s.arma
  <- arma(s, lag=list(ar=c(1,3,7,10,12,13,16,17,19),ma=12)))
summary(s.arma
  <- arma(s, lag=list(ar=c(1,3,7,10,12,13,16,17),ma=12)))
```

4. Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) method models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps. It combines both Auto

regression (AR) and Moving Average (MA) models as well as a differencing pre-processing step of the sequence to make the sequence stationary, called integration (I). The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA(p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models. The method is suitable for univariate time series with trend and without seasonal components.

R Code:-

```
        # Fit model to first few years of AirPassengers data
          air.model <-
Arima(window(AirPassengers,end=1956+11/12),order=c(0,1,1),
                  seasonal=list(order=c(0,1,1),period=12),lambda=0)
         plot(forecast(air.model,h=48))
         lines(AirPassengers)

         # Apply fitted model to later data
     air.model2 <- Arima(window(AirPassengers,start=1957),model=air.model)

        # Forecast accuracy measures on the log scale.
        # in-sample one-step forecasts.
         accuracy(air.model)
        # out-of-sample one-step forecasts.
         accuracy(air.model2)
        # out-of-sample multi-step forecasts
         accuracy(forecast(air.model,h=48,lambda=NULL),
         log(window(AirPassengers,start=1957)))
```

Differencing :- Transformations such as logarithms can help to stabilize the variance of a time series. Differencing can help 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 series is the change between consecutive observations in the original series, and can be written as

$$y'_t = y_t - y_{t-1}$$

The differenced series will have only T−1 values, since it is not possible to calculate a difference y′1 for the first observation. Sometimes data still not stable after applying first order differencing so we can apply some function like Log with the differencing

$$y''_t = y'_t - y'_{t-1}$$

$$= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

$$= y_{t-2} y_{t-1} + y_{t-2}.$$

Need for time series

A time series is a series of data points listed/indexed in the order of time. This project will help us to predict the prices of stocks, shares and many other factors which will be related to stock market and with help of this we can get the seasonality, pattern recognition of stocks. Time series analysis comprises of the methods for analyzing time series data in order to extract the meaningful statistics and other characteristics of data. Time series forecasting is the use of model to predict the future value based on the previously observed values and correlating it with the international stock exchange and getting the impact of various international/national stocks on the subsidiaries

Objective

The objective of the project is to forecast the future stock index by using ARIMA model and with past data on Indian Stock exchange (Sensex) to train the model then test its accuracy.

Scope

The project mainly comprises of BSE Sensex index having stocks of Indian companies which are the major players enlisted in stock market and other subsidiaries which are related to similar streams.

CHAPTER: II
Literature Review

In this chapter some research paper with respect to stock market and prediction methods are being discussed

Time Series Data Analysis for Stock Market Prediction using Data Mining Techniques with R (Angadi August 2015)

A stock exchange market depicts savings and investments that increase the effectiveness of the national economy. The future stock returns have predictive relationships with the already available information in the present and historical stock market indices data. ARIMA is a statistical time series forecast model which is known to be efficient especially for short-term forecasts. In this paper, the authors Mahantesh C. Angadi and Amogh P. Kulkarni propose a model for forecasting the stock market trends based on the technical analysis using historical stock market data and ARIMA model. It will automate the process of selecting the direction of future stock price indices and provides guidance for financial specialists to choose the best times when to purchase or sell stocks. The results are shown as visualizations using R programming language. The experimental results obtained demonstrated the potential of ARIMA model to predict the stock price indices on short-term basis. This could guide the investors in the stock market portfolio to make profitable investment decisions whether to buy, sell or hold a share. With the results obtained ARIMA model can compete reasonably well with other emerging forecasting techniques for short-term prediction.

Forecasting nifty bank sector stocks price using ARIMA model (Mohamed Ashik Nov 2017)

In this research paper, the Nifty 50 information of 2015 is analyzed and ARIMA model is utilized for forecasting. National Stock Exchange is widest and full automatic trading system in India. Nifty 50 is one of the stock value financial specialist and 50 organizations were invested in the traders. Autoregressive Integrated Moving Average model is one of the most accepted models and a vital area of the Box-Jenkins approach to time series analysis. In this paper introduced by Mohamed Ashik and Senthamarai Kannan, the Nifty 50 stock market price were assessed and anticipated the pattern of up and coming exchanging days securities exchange fluctuations by utilizing Box-Jenkins procedure. From their investigation, it was discovered that R-Square value is (94%) high and Mean Absolute Percentage Error is exceptionally little for the fitted model. Thus the prediction accuracy is more suitable for the Nifty 50 closing stock price. It concluded that closing stock price of Nifty 50 taken in the present study shows slow decreasing fluctuations trend for upcoming trading days.

Application of ARIMA Model in 2014 Shanghai Composite Stock Price Index (Renhao Jin, Sha Wang August 2015)*

In order to study the changes of Shanghai Composite Stock Price Index (SCSPI) and predict the trend of stock market fluctuations, this paper constructed a time-series analysis. A non-stationary trend is found, and an ARIMA model is found to fit the data. A short trend of Shanghai composite stock price index is then predicted using the established model. This paper does a study on 2014 Shanghai Composite Stock Price Index (SCSPI). In the process of model building, the original SCSPI data is found to be un-stationary, but the first order differencing data of original SCSPI data is stationary. By comparing with several models, ARIMA with p,d,q values (1, 1, 1) is chosen as the final model and it succeeds in predicting three steps trends of SCSPI. Considering the fluctuations of SCSPI, the model can be applied in finance practices. The fluctuations of stocks are non-rational, and it is influenced by several factors. No model can include all these factors. Although the predicted values from the suggested model is a little different from actual value, the increasing trend is agreed in the three steps, which is enough for financial practice. In the financial practice it is very perfect to make an investment to assure a coming profit.

An Empirical Analysis of Stock Market Price Prediction using ARIMA and SVM

Autoregressive Integrated Moving Average (ARIMA) model is the most acceptable and applied model in the terms of time series forecasting mechanism. Although, this model has its own kind of parametric limitations to capture the nonlinear patterns in the case of stock market prediction. Support vector machines (SVM) which is a novel neural network technique, can easily solve these problems compared to ARIMA model. Maximum number of the available reviewed paper and chapter has its own kind of limits as they concentrate on the particular application of financial market or explores machine learning tools and techniques that was applied on the particular dataset. This study provided a comparative study of some relevant existing tools and the techniques applied in the area of the financial market analysis. The main aim of the research paper was to perform a comparative study of ARIMA and SVM models in R-language and review the fundamental challenges and futuristic challenges of the models.

Indian Share Market Forecasting with ARIMA Model (December 2014)

Artificial Neural Networks (ANNs) are flexible deep learning algorithms and provides estimates that can be applied to a wide range of time series forecasting problems with a high degree of accuracy for predicting the futuristic value in share market and give a better future scope for investment. But the artificial neural

network is not satisfactory among analyst as it includes both theoretical and empirical results. The combination of different models can be an effective way of improving upon the predictive performance, if the models in ensembles are quite different. In this paper, a novel hybrid model of artificial neural network is proposed using the autoregressive integrated moving average (ARIMA) model to produce more accurate forecasting method than artificial neural network. On this context, the authors Swapnil Jadhav, Saurabh Kakade and Kaivalya Utp collected data on monthly closing stock indices of BSE Sensex and have tried to develop an appropriate model that would help them to forecast the future unknown values of Indian stock market indices using ARIMA. The analysis of the performance of the Indian stock market for six years with respect to time presented them a suitable time series ARIMA model (1,0, i) which helped them in predicting the approximate values of the future stock price. Out of the initial six different models, the authors choose ARIMA (i ,0, i) as the best model based on the fact that it satisfies all the conditions.

CHAPTER: III
RESEARCH METHODOLOGY

Objectives of this study

The main objective of this research is to empirically conduct analysis and predict future stock indices using ARIMA model.

Research Methodology

For this research project data science toolbox R & R studio is used to build the ARIMA model and evaluate the same. The research is conducted on stock index data collected from authentic source. Implementation of the methods require various tools on hardware which helps to get some observations. The software and hardware requirements are listed below.

Hardware Requirement: To create this observation and programming platform a powerful device is required to run various software and other program respectively.

- Workstation with Core i5 or Core i7 for faster processing
- RAM of minimum 8 GB to run R Studio and R framework effectively

Software Requirement:

R – Studio: R Studio is an open source integrated development environment for R, a programming language for statistical computing and graphics.

R core libraries: R is a free, open-source software and programming language developed in 1995 at the University of Auckland as an environment for statistical computing and graphics. R version 4.0.2 and above is required for the project.

Data source & Collection Methods:

Secondary Data:

Secondary data is that data which already existed. This is indirect collection of data from sources containing past or recent past information. The data for our analysis is fetched from below website.

- Money control

Time frame of the study:

11 years of BSE Sensex monthly stock index data is being selected for this study that is from 2009-2020

CHAPTER: IV
DATA MODELLING, ANALYSIS AND INTERPRETATION

Time-series analysis is a basic concept within the field of statistical learning that allows the analyst to find meaningful information in data points collected over time. To demonstrate this technique, we'll be applying it to BSE Sensex Stock Index (2009 – 2020) in order to find the best model to predict future stock values.

Step 1: First step is to load the required packages in R studio to perform the analysis. The project will make use of 'tseries' and 'forecast' packages in CRAN that have functions to execute auto.arima() and forecast() methods. 'ggplot2','ggfortify' and 'plotly' will be loaded for visualization and 'MLmetrics' package to evaluate the accuracy of the ARIMA model.

Step 2: Load the data

The next step is to get the stock index from secondary data sources. Here the data for BSE Sensex is fetched from money control for the year 2009 till 2020. The data is a monthly frequency indicating the closing price of stock indices each month. Below is the first few rows of the extracted data.

Figure 1: Dataset

Date	Price	Date	Price	Date	Price
01-01-2009	9,424.24	01-01-2013	19,894.98	01-01-2019	36,256.69
01-02-2009	8,891.61	01-02-2013	18,861.54	01-02-2019	35,867.44
01-03-2009	9,708.50	01-03-2013	18,835.77	01-03-2019	38,672.91
01-04-2009	11,403.25	01-04-2013	19,504.18	01-04-2019	39,031.55
01-05-2009	14,625.25	01-05-2013	19,760.30	01-05-2019	39,714.20
01-06-2009	14,493.84	01-06-2013	19,395.81	01-06-2019	39,394.64
01-07-2009	15,670.31	01-07-2013	19,345.70	01-07-2019	37,481.12
01-08-2009	15,666.64	01-08-2013	18,619.72	01-08-2019	37,332.79
01-09-2009	17,126.84	01-09-2013	19,379.77	01-09-2019	38,667.33
01-10-2009	15,896.28	01-10-2013	21,164.52	01-10-2019	40,129.05
01-11-2009	16,926.22	01-11-2013	20,791.93	01-11-2019	40,793.81
01-12-2009	17,464.81	01-12-2013	21,170.68	01-12-2019	41,253.74
01-01-2010	16,357.96	01-01-2014	20,513.85	01-01-2020	40,723.49
01-02-2010	16,429.55	01-02-2014	21,120.12	01-02-2020	38,297.29
01-03-2010	17,527.77	01-03-2014	22,386.27	01-03-2020	28,440.32
01-04-2010	17,558.71	01-04-2014	22,417.80	01-04-2020	31,743.08
01-05-2010	16,944.63	01-05-2014	24,217.34	01-05-2020	32,424.10
01-06-2010	17,700.90	01-06-2014	25,413.78	01-06-2020	34,731.73

Step 3: Data clean up

Next step we need to check for discrepancies in data such as null values and perform data clean-up operations like scaling, normalization etc. Here since we have only two variables with one being date field and the other stock indices, scaling is not required for ARIMA model. But for few others like ANN, LTVM models of prediction scaling or normalizing data may be required. The dataset is checked for any null and missing values using is.na() function in R

Step 4: Exploratory data analysis

Next we perform exploratory data analysis to find outliers and identify the structure of the data frame. We want to get a feel for our data in order to decide which models may be appropriate for our forecast. To do this, we will plot our data and diagnose for trend, seasonality, heteroskedasticity and stationarity.

Step 5: Create Time series data object:

The BSE Stock Index data should be in the form of a time series object type; this means that our data exists over a continuous time interval with equal spacing between every two consecutive measurements. The data collected may be in the form of dataframe or tables hence by using the ts() function in R, we would be able to convert it to time series objects.

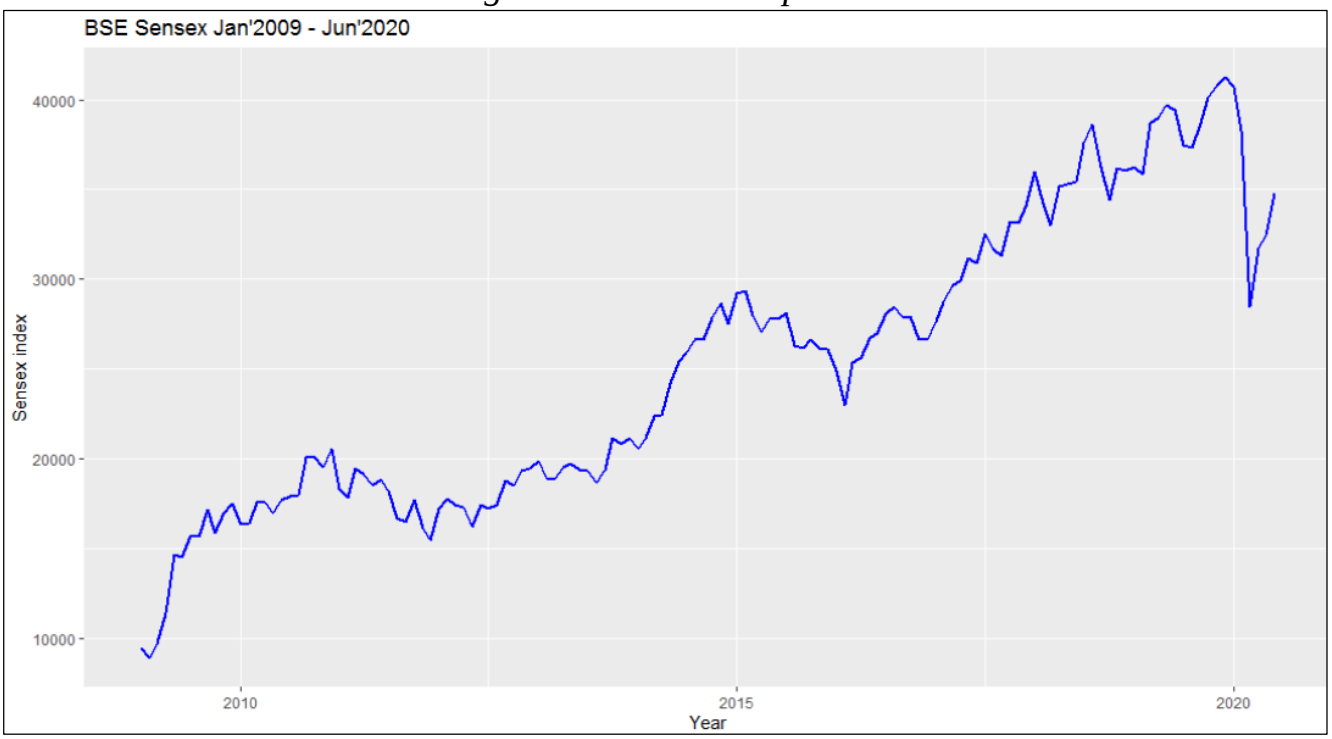
Step 6: Plot the time series:

Plotting data is the most critical step in the exploratory analysis phase. Visualizing our time-series data enables us to make inferences about important components, such as trend, seasonality, heteroskedasticity, and stationarity. Here is a quick summary of each:

- **Trend**: We say that a dataset has a trend when it has either a *long-term increase* or *decrease*.
- **Seasonality**: We say that a dataset has seasonality when it has patterns that repeat over known, fixed periods of time (e.g. monthly, quarterly, yearly).
- **Heteroskedasticity**: We say that a data is *heteroskedastic* when its variability is not constant (i.e., its variance increases or decreases as a function of the explanatory variable).
- **Stationarity**: A stochastic process is called *stationary* if the mean and variance are constant (i.e., their joint distribution does not change over time).

Below is the plot obtained using the autoplot() function :

Figure 2: Time series plot

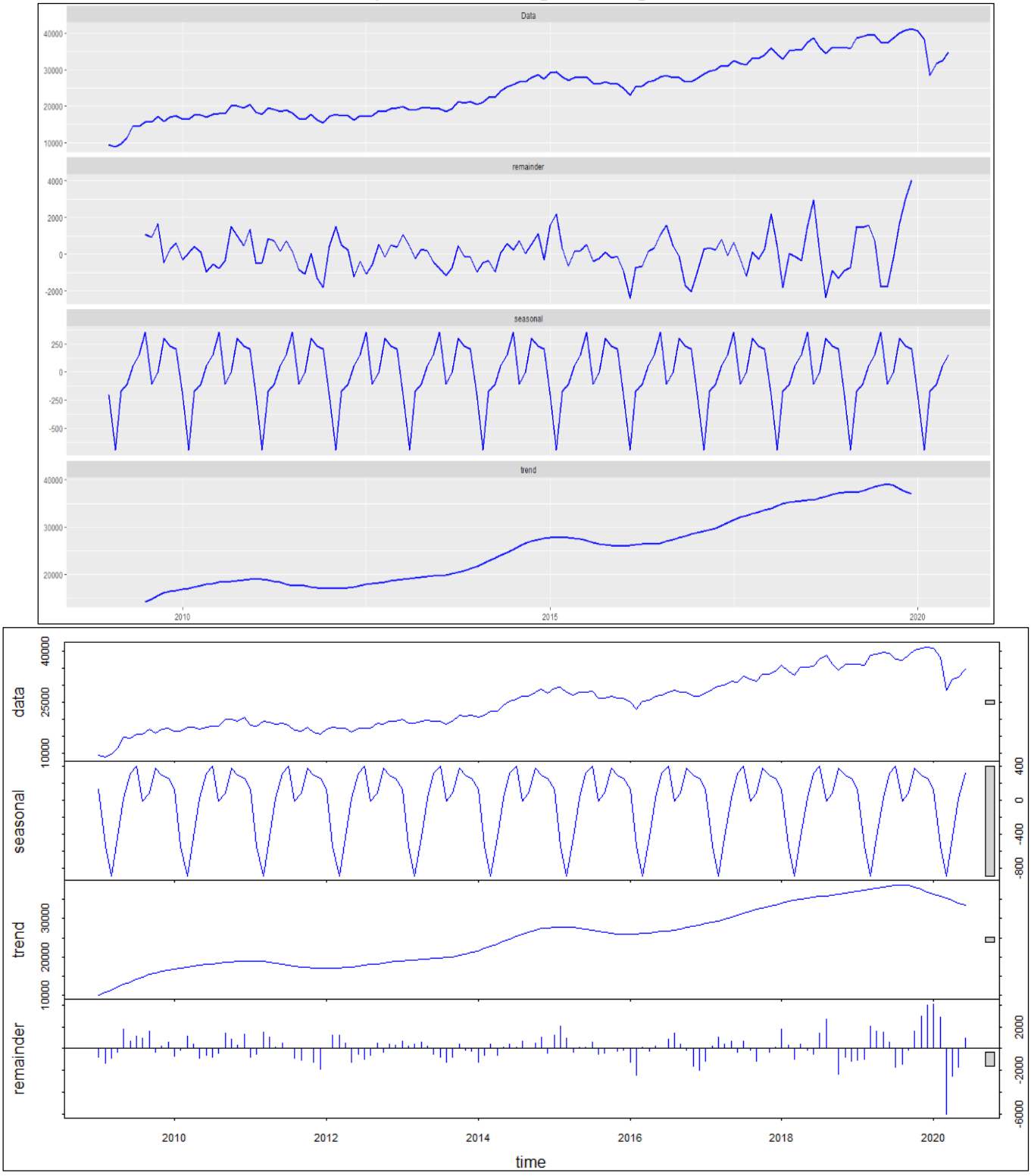


We can easily see that our time series has instances of both positive and negative trend. Overall, it is very volatile, which tells us that we will have transform the data. The time series also has seasonal fluctuations that repeat each year.

Step 8: Decompose the time series

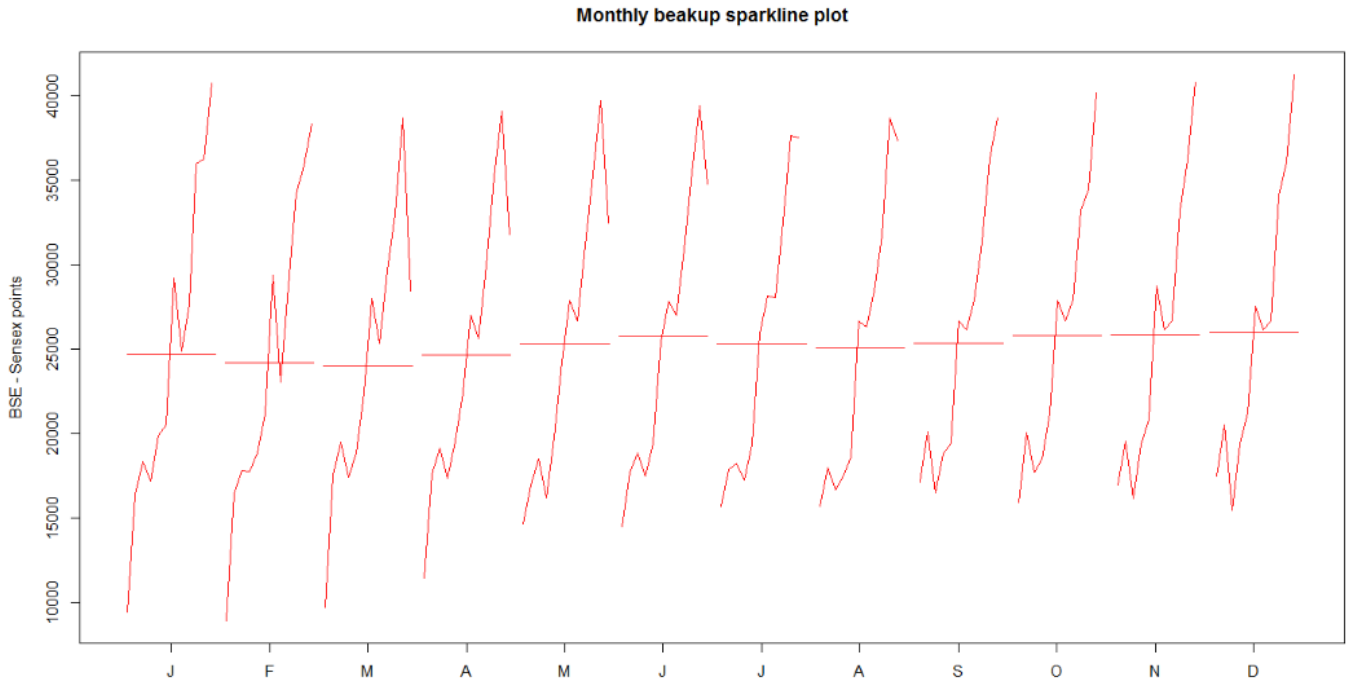
Beyond understanding the trend of the time series, we can further understand the anatomy of the data. For this reason, we would break down our time series into its seasonal component, trend, and residuals. Using decompose() function in R. Below is the plot after decomposition.

Figure 3: Decomposition plot



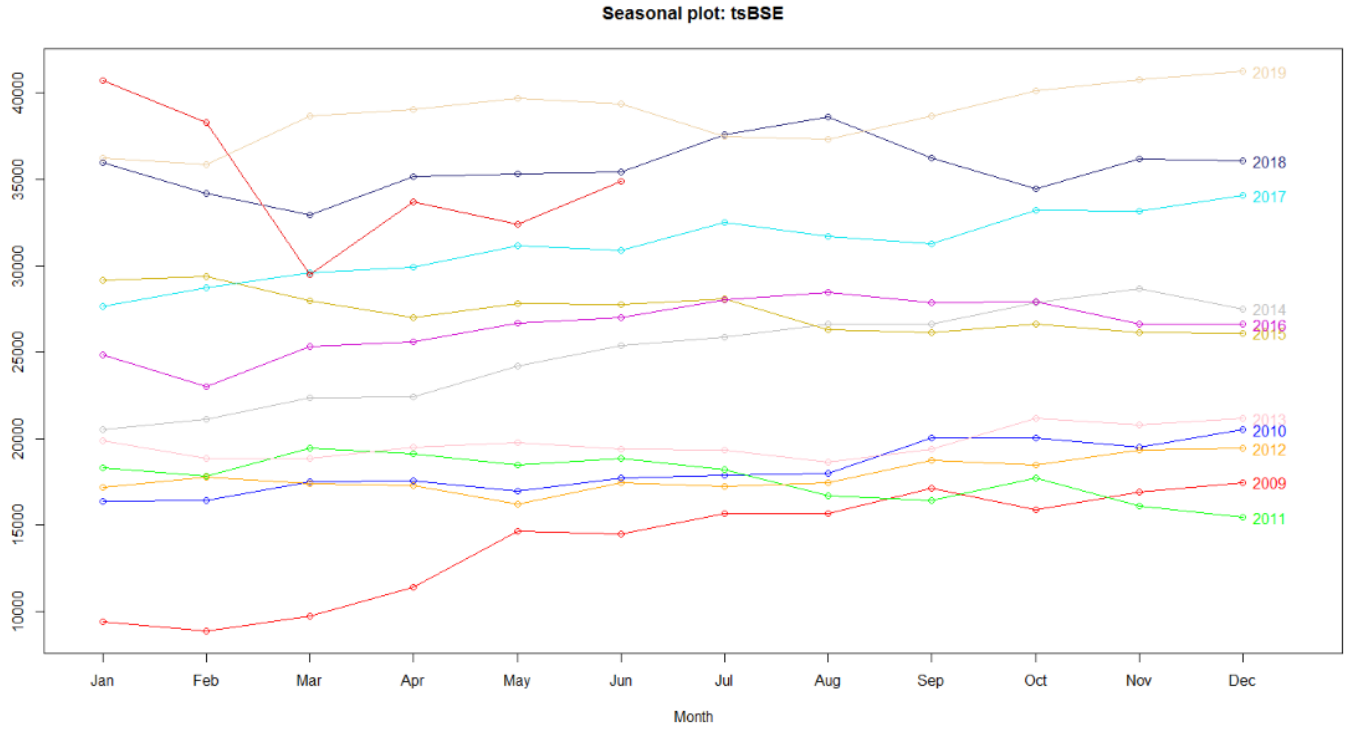
The monthly plot below performed using monthplot() function displays the subseries for each month (all January values connected, all February values connected, and so on), along with the average of each subseries. From this graph, it appears that the trend is increasing for each month in a roughly uniform way. Additionally, the greatest increase in Sensex points appears in January due to New Year stock purchases. March sees a dip at the end of the month due to budget session or annual financial release of companies.

Figure 4: Monthly plot



The seasonal plot (below figure) done using seasonplot() displays the subseries by each year. Each color displays each year pattern for various months.

Figure 5: Seasonal plot



Step 7: Test for stationarity

The next step is to test for stationarity in our data, the **Augmented Dickey-Fuller Test** for stationarity will be used. The null hypothesis states that large *p* values indicate non-stationarity and smaller *p* values indicate stationarity. (We will be using 0.05 as our alpha value. Below is the output for the BSE data set:

```
Augmented Dickey-Fuller Test

Data:  tsBSE
Dickey-Fuller = -2.1251, Lag order = 5, p-value = 0.5244
Alternative hypothesis: stationary
```

We could see that the p value for the ADF test conducted for BSE Sensex data is greater than 0.05 which means that the null hypothesis of non-stationarity will stay and will not be rejected. Hence we have statistically proved that the data is non – stationary and we have to further transform the data to stationary one.

Step 9: Diagnosing the ACF and PACF Plots of Our Time-Series Object

ACF stands for "autocorrelation function" and PACF stands for "partial autocorrelation function." The ACF and PACF diagnosis is employed over a time-series to determine the order in which we are going to create our model using ARIMA. In other words, a time series is stationary when its mean, variance, and autocorrelation remain constant over time.These functions help us understand the correlation component of different data points at different time lags. Lag refers to the time difference between one observation and a previous observation in a dataset. The dashed line delimits the significant and insignificant correlations: values above the line are significant. (The height of the dashed line is determined by the amount of data). From the below plot there is a significant correlation at lag 0.0 that decreases after a few lags.

Figure 6: ACF plot

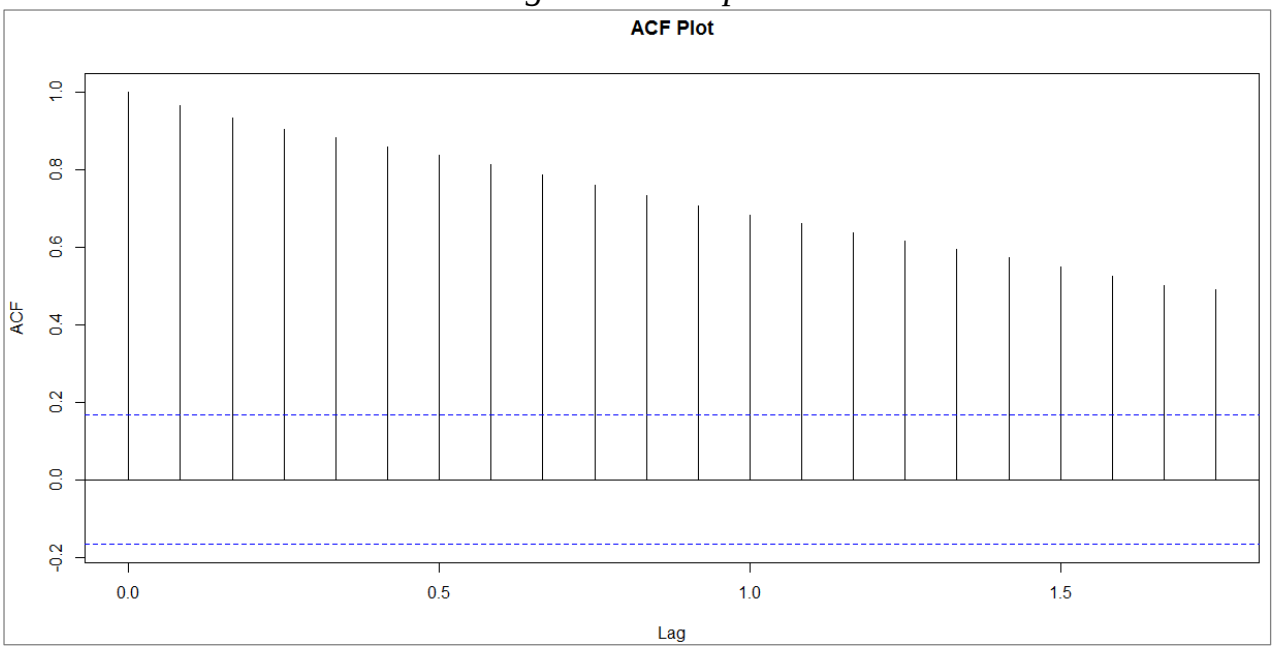
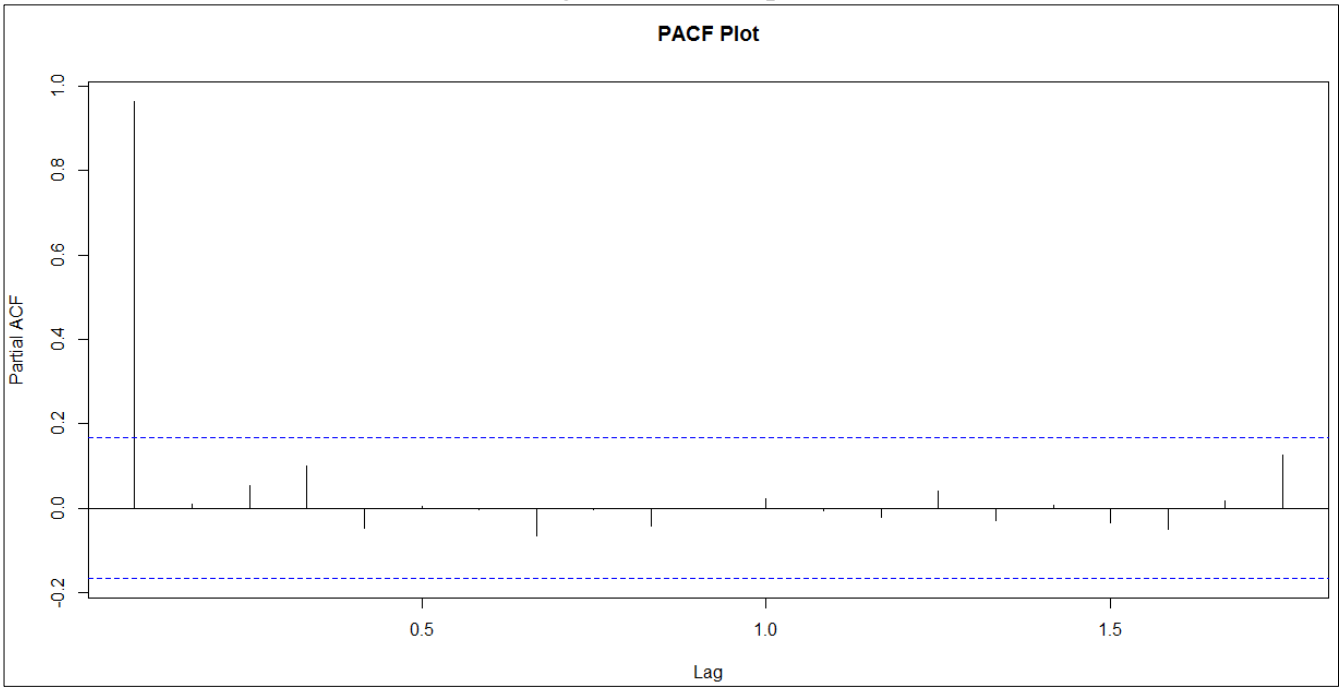


Figure 7 PACF plot



The partial autocorrelation function is a measure of the correlation between observations of a time series that are separated by k time units (yt and yt–k), after adjusting for the presence of all the other terms of shorter lag (yt–1, yt–2, ..., yt–k–1). The above plot shows there is a significant correlation at lag 0.1 followed by correlations that are not significant (vertical lines are below the blue dotted line). This pattern indicates an autoregressive term of order 0.

Step 9: Transform Data to Adjust for Non-Stationarity

Based on our visual inspection of the time-series object and the statistical tests used for exploratory analysis, it is appropriate to difference our time-series object to account for the non-stationarity. A way to make a time series stationary is to find the difference across its consecutive values. This helps stabilize the mean, thereby making the time-series object stationary.

Step 10: Build ARIMA model using auto.arima()

Next we build the ARIMA model. Though there are many methods to build the model, in this project we would be using the auto.arima() function in 'forecast" package which will automatically select a best ARIMA model. The function iterates and selects an ARIMA model with best p,d,q values. These are values that minimize the AIC criterion over a large number of possible models.

- P is the number of autoregressive terms
- D is the number of non-seasonal differences needed for stationarity
- Q is the number of lagged forecast errors in the prediction equation

By default, auto.arima limits p and q to the range $0 \leq p \leq 5$ and $0 \leq q \leq 5$. If we believe that the model needs more coefficients, we can use max.p and max.q to expand the search limits.

The output in R below corresponds with (p,d,q)(P,D,Q)[m] where 'm' stands for number of periods in season, i.e. months in year and in below case its 12.

```
ARIMA(2,1,2)(1,0,1)[12] with drift         : 2380.578
 ARIMA(0,1,0)              with drift        : 2380.213
 ARIMA(1,1,0)(1,0,0)[12] with drift         : 2376.491
 ARIMA(0,1,1)(0,0,1)[12] with drift         : 2378.664
 ARIMA(0,1,0)                               : 2380.536
 ARIMA(1,1,0)              with drift        : 2382.12
 ARIMA(1,1,0)(2,0,0)[12] with drift         : Inf
 ARIMA(1,1,0)(1,0,1)[12] with drift         : 2377.227
 ARIMA(1,1,0)(0,0,1)[12] with drift         : 2378.721
 ARIMA(1,1,0)(2,0,1)[12] with drift         : Inf
 ARIMA(0,1,0)(1,0,0)[12] with drift         : 2374.715
 ARIMA(0,1,0)(2,0,0)[12] with drift         : 2374.302
 ARIMA(0,1,0)(2,0,1)[12] with drift         : 2376.155
 ARIMA(0,1,0)(1,0,1)[12] with drift         : 2375.466
 ARIMA(0,1,1)(2,0,0)[12] with drift         : 2376.058
 ARIMA(1,1,1)(2,0,0)[12] with drift         : Inf
 ARIMA(0,1,0)(2,0,0)[12]                    : 2375.246

 Best model: ARIMA(0,1,0)(2,0,0)[12] with drift
```

We could see that the auto.arima() after running few iterations, chooses the best fit model as (0,1,0)(2,0,0) [i.e (p,d,q)(P,D,Q)] based on lowest AIC values.

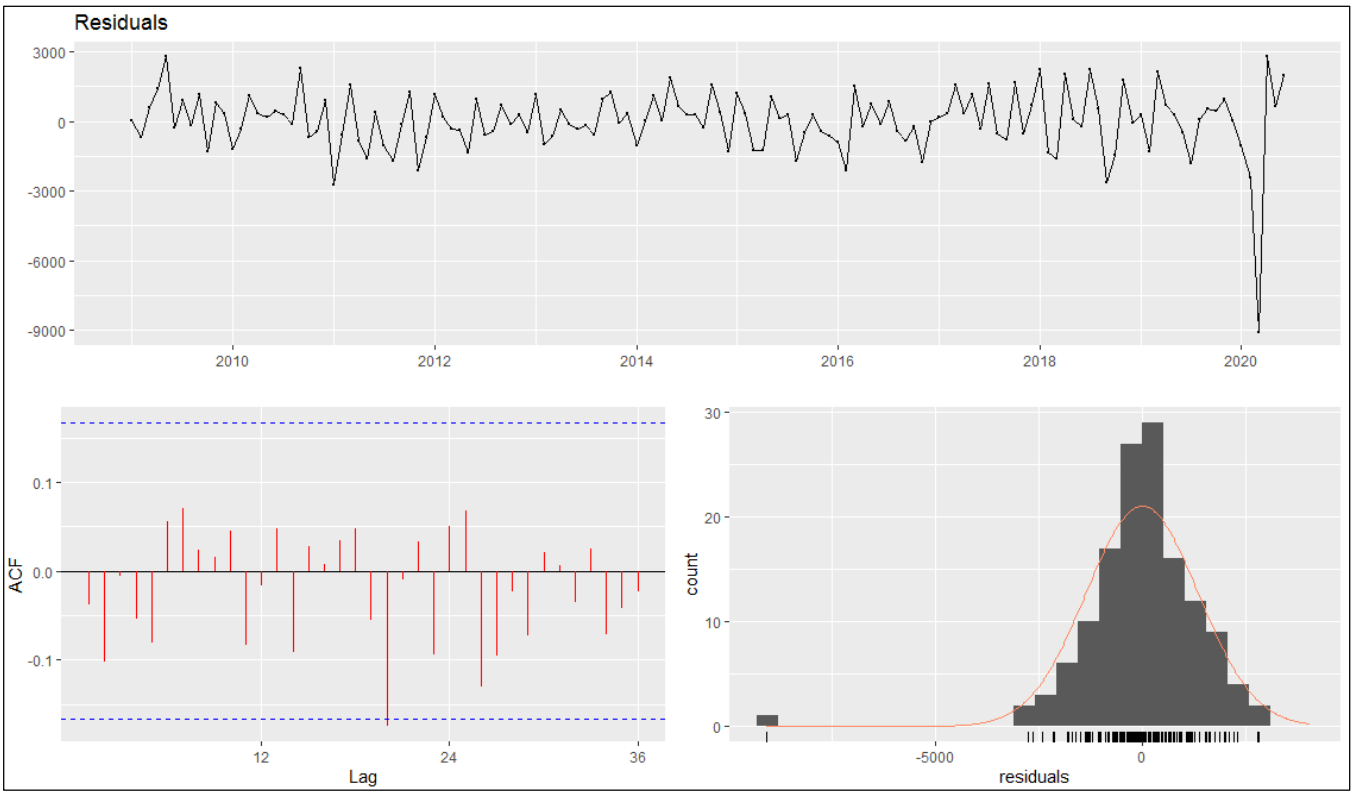
The summary of the ARIMA model is shown below :

```
Coefficients:
          sar1     sar2      drift
      -0.2714   0.1833   187.8167
s.e.   0.1080   0.1160   105.3365

sigma^2 estimated as 1867858:  log likelihood=-1183.15
AIC=2374.3   AICc=2374.6   BIC=2385.98

Training set error measures:
                     ME      RMSE       MAE          MPE      MAPE      MASE         ACF1
Training set 2.864798 1346.743 928.2619 -0.02910001 3.938919 0.2703027 -0.03740878
```

Two of the most widely used output are Akaike information criteria (AIC) and Bayesian information criteria (BIC). These criteria are closely related and can be interpreted as an estimate of how much information would be lost if a given model is chosen. When comparing models, one wants to minimize AIC and BIC.

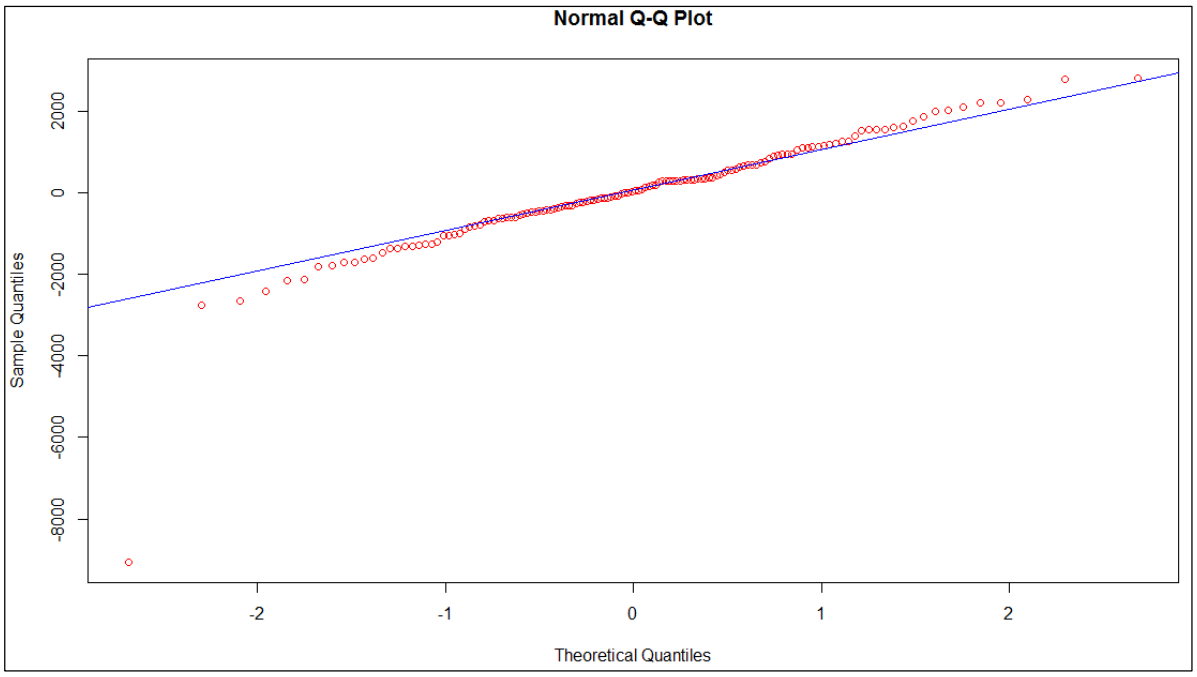
Next step is to analyze the residuals of the ARIMA model. The residuals are the errors in prediction and by plotting the values we can visualize the accuracy in prediction of p,d,q values by the model. The checkresiduals() function in forecast package is used to get the residual plots.

Figure 8: Residual plots



If the model is appropriate, the residuals should be normally distributed with mean as zero, and the autocorrelations should be zero for every possible lag. In other words, the residuals should be normally and independently distributed (no relationship between them). The qqnorm() and qqline() functions produce the plot below. Normally distributed data should fall along the line. In this case, most of the results are along the line hence it is normally distributed and our model is appropriate.

Figure 9: Q – Q plot



Step 10: Test for stationarity after building ARIMA model

Once the best fit ARIMA model is determined using auto.arima(), the **Augmented Dickey-Fuller Test** for stationarity will be performed again on residuals to prove for stationarity to proceed with forecast. The null hypothesis states that large *p* values indicate non-stationarity and smaller *p* values indicate stationarity. (We will be using 0.05 as our alpha value). Below is the output for the ARIMA model:

```
        Augmented Dickey-Fuller Test

Data:  ts(tsmodel$residuals)
Dickey-Fuller = -4.9099, Lag order = 5, p-value = 0.01
Alternative hypothesis: stationary
```

We could see that the p value for the ADF test conducted for ARIMA residuals data is very less than 0.05 which means that the null hypothesis of non-stationarity will be rejected. Hence we have statistically proved that the data is stationary now and we can proceed with forecasting the time series.

Step 11: Forecast the future stock indices

Using forecast() function available in package loaded, the future values are forecasted and plotted. The argument or parameter to be passed to the function includes future time periods in months (h value) and the confidence interval (95% C.I) below is the future forecasted value by the ARIMA model along with the 95% confidence interval values. The below values are one year forecast by passing h value as 12.

```
forecast(tsmodel,h = 12,level = c(95))
```

	Forecast	Lo 95	Hi 95
Jul 2020	35855.60	33176.92	38534.27
Aug 2020	36290.58	32502.37	40078.80
Sep 2020	35689.52	31049.91	40329.12
Oct 2020	35169.94	29812.59	40527.29
Nov 2020	35515.11	29525.41	41504.80
Dec 2020	35571.55	29010.16	42132.94
Jan 2021	35954.34	28867.24	43041.45
Feb 2021	36745.79	29169.36	44322.23
Mar 2021	40139.53	32103.51	48175.56
Apr 2021	39513.30	31042.58	47984.01
May 2021	39657.97	30773.81	48542.13
Jun 2021	39177.48	29898.28	48456.68

The forecasted values are then plotted using autoplot() function, the forecast is done for 1 year, 5 years and 10 years period and below are the plot results.

Figure 10: One Year forecast plot

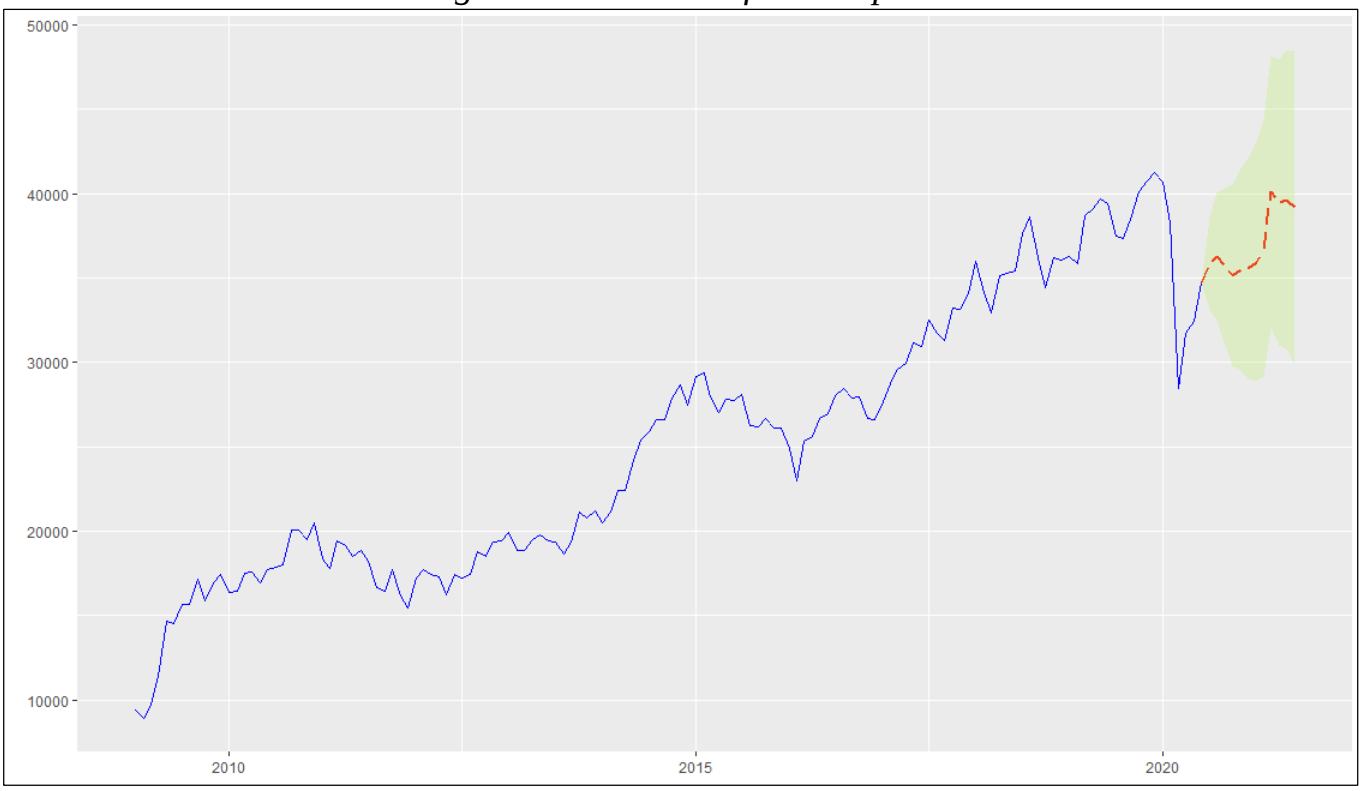


Figure 11: Five year forecast

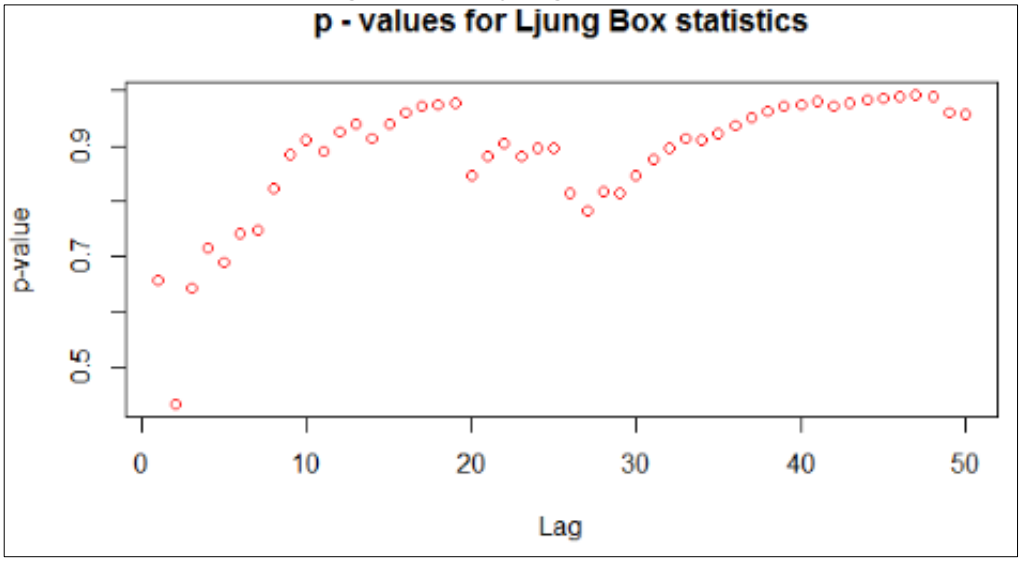
Figure 12: 10 Year Forecast

We could see that the forecast was accurate for 1 year period and as the future forecast period increases the accuracy of the model output decreases and becomes exponential smoothing forecast based on trend. In the 10 year forecast we could see that from the year 2022, the forecast value smoothens and becomes exponential smoothing value with increasing trend. Thus it is clear that the ARIMA model is best suited for short – term forecast.

Step 12: Ljung Box test

Ljung-Box statistic is used to test whether a time series are random and independent. If observations are not independent, one observation can be correlated with a different observation k time units later, a relationship called autocorrelation. Autocorrelation can decrease the accuracy of a time-based predictive model, such as time series plot, and lead to misinterpretation of the data. Using Box.test() function in R the statistical test is performed and plotted.

Figure 13: Ljung Box Plot



We could see that for lag values from 1 to 50, all the p-value of the test are greater than 0.05 which accepts null hypothesis that the residuals for our time series model are independently distributed.

Step 13: Evaluating the Accuracy of the model

The last step of the ARIMA modeling is to check the accuracy and evaluate the model. We are creating a new data frame with all the predicted stock values by the ARIMA model from 2009 to 2020 and the actual value in the original data set. We are plotting the same using ts.plot() function and below represents the plot with actual and predicted values. We could see that the prediction is nearly accurate and traces the actual values with few error points/residuals in few data points.

Figure 14: Actual vs Predicted plot

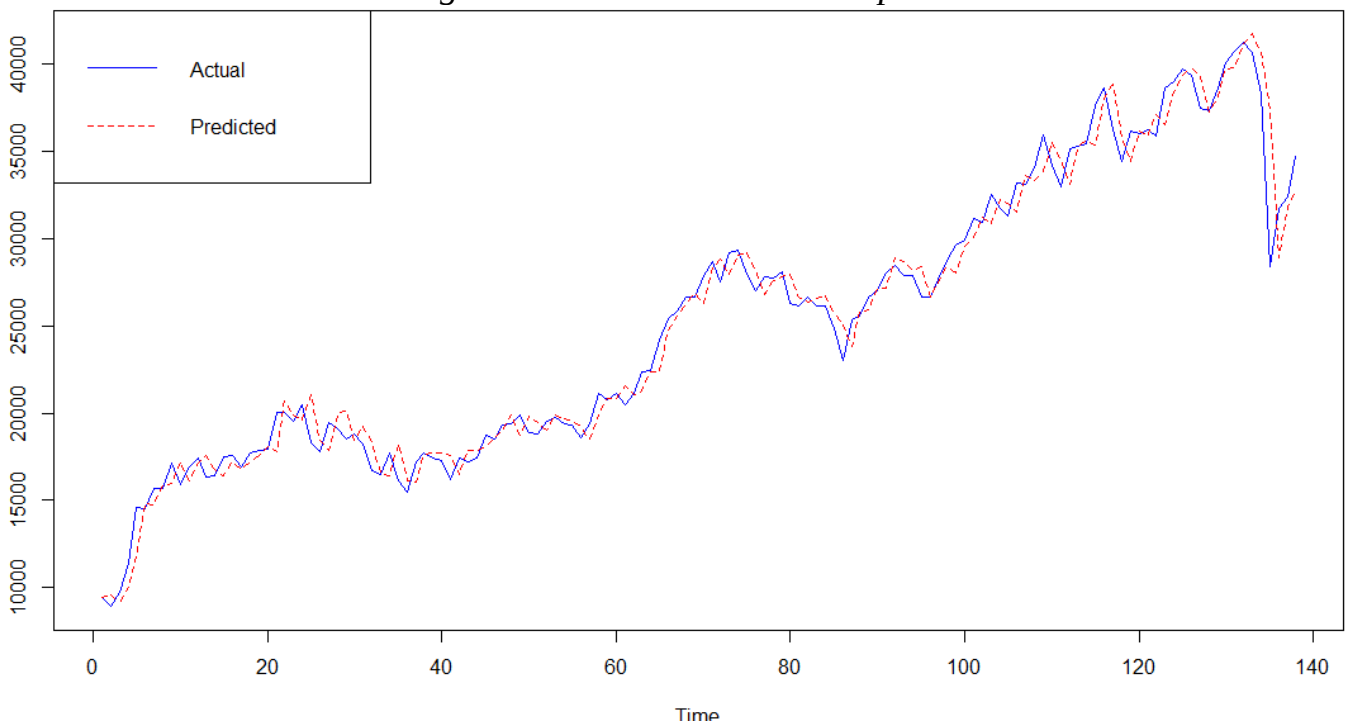


Figure 15: Abbreviations

Measure	Abbreviation	Definition
Mean error	ME	mean(e_t)
Root mean squared error	RMSE	sqrt(mean(e_t^2))
Mean absolute error	MAE	mean(| e_t |)
Mean percentage error	MPE	mean(100 * e_t / Y_t)
Mean absolute percentage error	MAPE	mean(| 100 * e_t / Y_t |)
Mean absolute scaled error	MASE	mean(| q_t |) where $q_t = e_t$ / (1/(T-1) * sum(| $y_t - y_{t-1}$ |)), T is the number of observations, and the sum goes from t=2 to t=T

The above table depicts the metrics used for measuring the accuracy of the ARIMA model.

By executing the accuracy () function for the ARIMA model, below table values are generated for the model.

ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
2.864798	1346.743	928.2619	-0.02910001	3.938919	0.2703027	-0.03740878

Mean Error (ME): It is the mean of the difference between the measured value and true/correct value.

Mean Absolute Error (MAE): MAE is the mean of the absolute errors. The absolute error is the absolute value of the difference between the forecasted value and the actual value

Mean Absolute Percentage Error (MAPE): It is a simple average of absolute percentage errors. The MAPE calculation is as follows:

$$\text{MAPE} = \frac{\sum \frac{|A-F|}{A} \times 100}{N}$$

Where A= Actual, F= Forecast, N= Number of observations

From the output of our ARIMA model we could see that the MAPE value is 3.9 % (Approx. 4%) which implies that our ARIMA model is **96%** accurate in predicting the next observations.

Mean Absolute Scaled Error (MASE): It is the scale-free error metric that gives each error as a ratio compared to a baseline's average error. If the value of MASE is less than 1 then the forecast is better than a naive one. In our model the MASE value is 0.27 which is less than one.

ACF1: It is a measure of how much is the current value influenced by the previous values in a time series. Specifically, the autocorrelation function tells you the correlation between points separated by various time lags. Typically we would expect the autocorrelation function to fall towards 0 as points become more separated (i.e. n becomes large in the above notation) because it's generally harder to forecast further into the future from a given set of data. This is not a rule, but is typical that ACF(0)=1 (all data are perfectly correlated with themselves), ACF(1)=.9 (the correlation between a point and the next point is 0.9), ACF(2)=.4 (the correlation between a point and a point two time steps ahead is 0.4)...etc.

Root Mean Squared Error (RMSE): It measures accuracy for continuous variables. The RMSE will always be larger or equal to the MAE and greater the difference between RMSE and MAE, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude. Both the MAE and RMSE can range from 0 to ∞ and lower their values the better.

CHAPTER IV
LEARNINGS

The key learnings were how we build best effective ARIMA model using auto.arima function and evaluated the same by means of residual plots and other metrics such as MAPE and MASE. Short term forecasts like 1 year forecast seems to be accurate but as forecast period increases the accuracy of the ARIMA model decreased. Learnt about the significance of Augmented Dickey Fuller test as a measure of stationarity. The ACF and PACF plots and its interpretations were discussed. We learnt to decompose the time series to understand the seasonality and trend with individual visuals. Additionally plots like seasonal and monthly plots provided more insights on the pattern of the BSE Sensex index. We learned how R and R studio were best suited data science toolkits to perform the ARIMA model and analysis due to readily available packages and large open source community which developed various packages and documentations that served as reference points for our analysis.

LIMITATIONS

- ARIMA is used only for short term forecast
- ARIMA model requires a lot of training data or lot of time series observations compared to other forecasting models
- Prerequisite for ARIMA is that the data has to be stationary.
- White noise or outliers in the data can affect the model accuracy.
- ARIMA captures only linear relationships and not non – linear like quadratic or polynomial. In this case neural network model is appropriate
- Python is more popular compared to R since python has several visualization libraries like matplotlib, seaborn and bokeh. It also has modules like pandas, numpy, statsmodel and scikitlearn which offers wide variety of functions and methods for building ARIMA model.

CONCLUSION:

The purpose of the project is to forecast short term stock indices of BSE Sensex using ARIMA in R. We concluded that the data was non stationary and the same was proved statistically via augmented Dickey Fuller test. Further the time series was decomposed into components such as trends and seasonality to understand the patterns in the Sensex data. ACF and PACF plots were interpreted to gain insights on the significance of the time series. We learned how auto.arima() function in forecast package has been leveraged to get the best fitted ARIMA model with the best p, d, q values that corresponds to the lowest AIC values. Thus auto.arima enabled us to automate the task of choosing the values rather than manually getting the p, d, q values via complicated codes or calculations based on logarithms and differencing. The residuals of the ARIMA model were plotted and we found that the values were normally distributed thus strengthening the hypothesis that the ARIMA model built is appropriate for forecast. We then forecasted the BSE Sensex data for next one year and saw how accurate the forecast was along with 95% confidence intervals. We concluded that ARIMA is best for short term forecast by comparing the forecasts of longer periods like next ten years which turned out to be exponential smoothing based forecast. Box – Ljung test was performed to check whether the time series is independent and identically distributed and we found that the probability values for lag values were greater than 0.05. Then we evaluated the accuracy of the model by backtracking the predicted values from the model and the actual values from the data. Accuracy of the models were also evaluated by the metrics such as MAE, RMSE , MAPE, MPE and ME. Last we can conclude how R – studio can best be leveraged for ARIMA model building due to custom packages and strong community that cater to all statistical computing needs. Compared to tools like Tableau and SPSS, R and R studio are best suited for time series analysis due to easy steps and readily available computation functions, UI friendly plot interfaces and outputs, debugging and troubleshooting of modelling steps.

REFERENCES:

[1] Banerjee, D., "Forecasting of Indian stock market using timeseries ARIMA model", 2nd IEEE International Conference on Business and Information Management (ICBIM), January 2014, pp. 131-135.

[2] Ayodele A. Adebiyi, Aderemi O. Adewumi, Charles K. Ayo, "Stock price prediction using the ARIMA model", 16th IEEE International Conference on Computer Modelling and Simulation (UKSim), March 2014, pp. 106 -112.

[3] Qasem A. Al-radaideh, Adel Abu Asaf, Eman Alnagi, "Predicting stock prices using data mining techniques", The International Arab Conference on Information Technology 2013.

[4] Han, J., Kamber, M., Jian P., "Data mining concepts and techniques". San Francisco, CA: Morgan Kaufmann Publishers, 2011.

[5] Mohamed Ashik. A and Senthamarai Kannan. K (2017), "Forecasting National Stock Price Using ARIMA Model", Global and Stochastic Analysis, 4(1), 77-81

https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/

https://blogs.oracle.com/datascience/performing-a-time-series-analysis-on-the-sandp-500-stock-index

https://blogs.oracle.com/datascience/introduction-to-forecasting-with-arima-in-r

http://rstudio-pubs-static.s3.amazonaws.com/399202_e78dfd98a7434405893996f2e7cf4b37.html

https://towardsdatascience.com/time-series-analysis-with-auto-arima-in-r-2b220b20e8ab

https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/#:~:text=arima()%20function%3A&text=arima()%20function%20in%20R,chosen%20by%20minimizing%20the%20AICc.

http://ucanalytics.com/blogs/step-by-step-graphic-guide-to-forecasting-through-arima-modeling-in-r-manufacturing-case-study-example/

https://stackoverflow.com/questions/49099541/the-residuals-vs-fitted-values-plot-of-arima-model-in-r
https://stats.stackexchange.com/questions/194453/interpreting-accuracy-results-for-an-arima-model-fit