RAMAKRISHNA MISSION RESEDENTIAL COLLEGE (AUTONOMOUS)
Narendrapur, Kolkata – 700103

FORECASTING NIFTY 50 BENCHMARK INDEX

NAME : Spandan Ghosh
ROLL NO. : STUG/001/19
REG. NO. : A03-1142-0001-19
SEMESTER : VI
SUBJECT : Statistics
COURSE : DSE04
SESSION : 2019-22

EXAMINED,
Date: 24/7/22

ACKNOWLEDGEMENT

Presentation, Inspiration and motivation have always played a key role in the success of any venture.

I express my sincere thanks to principal Maharaj, Swami Ekachittananda, Ramakrishna Mission Residential College, Narendrapur.

I take this opportunity to express my profound gratitude and deep regards to my mentor Prof. Dilip Kumar Sahoo for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. I am also extremely grateful to all my other professors Dr. Parthasarathi Chakraborty, Sri Subhadeep Banerjee, Sri Palas Pal and Sri Tulsidas. The blessing, help and guidance given by them shall carry me a long way in the journey of life on which I am about to embark.

Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this assignment would not be possible.

ABSTRACT

The accuracy of several models used to forecast the Nifty 50 is investigated in this research. We used the decomposition method, the Simple Exponential Method (Holt's Method), the Double Exponential Method (Holt's Method), the Triple Exponential Method (Winter Holts' Method), and the ARIMA method. Before using the ARIMA approach, we used the Dickey Fuller Test to determine that the time series is non-stationary. As a consequence, we transformed the time series to stationary form using the differencing approach. We found that the ARIMA method is the best technique for forecasting.

Keywords:

Trend, Seasonality, Cyclical Fluctuations, Irregularity, Exponential Methods, Lag Plot, Auto Correlation Function, Augmented Dickey Fuller Test, Auto-ARIMA, Forecasting.

CONTENTS

1. INTRODUCTION

Time series forecasting is a method for predicting future events by looking at past trends, with the underlying premise being that future trends will be similar to past trends. Forecasting is the process of predicting future values using models fitted to historical data. Time series forecasting is necessary for prediction problems with a time component because it offers a data-driven method for effective and efficient planning.

Time series forecasting is one of the most applied data science techniques in business, finance, supply chain management, production and inventory planning. Many prediction problems involve a time component and thus require extrapolation of time series data, or time series forecasting. Time series forecasting is also an important area of machine learning (ML) and can be cast as a supervised learning problem. ML methods such as Regression, Neural Networks, Support Vector Machines, Random Forests and XGBoost — can be applied to it. Forecasting involves taking models fit on historical data and using them to predict future observations.

A nation's economy heavily depends on its equities market. For businesses and investors, it is one of the most important investment platforms. A weighted average of 50 of the largest companies in India that are listed on the National Stock Exchange makes up the NIFTY 50, a stock market index used in India. One of the two main stock indexes in India, the other being the BSE SENSEX, is this one. The Nifty 50 is owned and managed by NSE Indices, a fully owned subsidiary of the NSE Strategic Investment Corporation Limited (formerly known as India Index Services & Products Limited).

The NIFTY 50 index gives investment managers access to the Indian market through a single portfolio and encompasses 13 sectors of the Indian economy (as of 30 April 2021). Due to the growth of sectoral indexes including NIFTY Bank, NIFTY IT, NIFTY Pharma, NIFTY SERV SECTOR, NIFTY Next 50, etc., the NIFTY 50 index's market capitalization share of the NSE decreased from 65 percent to 29 percent between 2008 and 2012. Financial services receive a weighting of 39.47 percent in the NIFTY 50 Index, followed by the energy sector at 15.31%, 13.01 percent for IT, 12.38 percent for consumer goods, and 6.11 percent for automobiles.

2. PURPOSE OF THE STUDY

NIFTY 50 can be used to benchmark fund portfolios, establish index funds, exchange traded funds (ETFs), and structured products, among other things. The index follows the performance of a portfolio of blue-chip businesses, India's largest and most liquid stocks, and can be considered a real reflection of the Indian stock market. As a result, risk management has become vital in the Indian stock market, and volatility plays a critical role in analysing the risks of various stock market features such as portfolio risk management, derivative pricing, and hedging approaches. As a result, some academics have recently expressed an interest in anticipating stock market volatility.

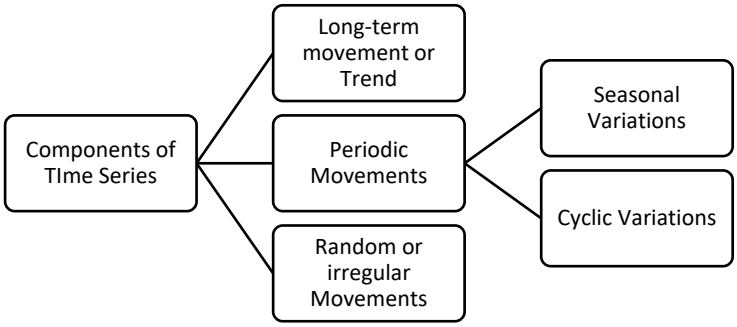
Forecasting stock market returns has recently gained popularity, possibly due to the fact that if the market's direction is properly forecasted, investors may be better advised. The profitability of stock market investing and trading is heavily reliant on predictability. If a system can reliably predict the tendencies of the volatile stock market, the owner of the system will become wealthy. Furthermore, forecasted market patterns will assist market regulators in implementing corrective measures.

3. REVIEW OF LITERATURE

3.1. **Time series analysis**: It is a method of studying a collection of data points over a period of time. Instead of capturing data points sporadically or arbitrarily, time series captures data points at constant intervals over a predetermined length of time. To maintain consistency and dependability, time series analysis often requires a high number of data points. A large data collection ensures that your sample size is representative and the analysis can cut through noisy data. It also guarantees that any trends or patterns found aren't outliers and that seasonal variations are also considered.

3.2. **Components for Time Series Analysis:** The various reasons or the forces which affect the values of an observation in a time series are the components of a time series. The four categories of the components of time series are:

- Trend
- Seasonal Variations
- Cyclic Variations
- Random or Irregular movements



3.2.1. **Trend:** The trend shows the general tendency of the data to increase or decrease during a long period of time. A trend is a smooth, general, long-term, average tendency. It is not always necessary that the increase or decrease is in the same direction throughout the given period of time. It is observable that the tendencies may increase, decrease or are stable in different sections of time. But the overall trend must be upward, downward or stable. The population, agricultural production, items manufactured, number of births and deaths, number of industry or any factory, number of schools or colleges are some of its examples showing some kind of tendencies of movement.

3.2.2. **Seasonal Variations:** These are the rhythmic forces which operate in a regular and periodic manner over a span of less than a year. They have the same or almost the same pattern during a period of 12 months. This variation will be present in a time series if the data are recorded hourly, daily, weekly, quarterly, or monthly. These variations come into play either because of the natural forces or man-made conventions. The various seasons or climatic conditions play an important role in seasonal variations. Such as production of crops depends on seasons, the sale of umbrella and raincoats in the rainy season, and the sale of electric fans and A.C. shoot up in summer seasons. The effect of man-made conventions such as some festivals, customs, habits, fashions, and some occasions like marriage is easily noticeable. They recur themselves year after year. An upswing in a season should not be taken as an indicator of better business conditions.

3.2.3. **Cyclic Variations:** The variations in a time series which operate themselves over a span of more than one year are the cyclic variations. This oscillatory movement has a period of oscillation of more than a year. One complete period is a cycle. This cyclic movement is sometimes called the ‘Business Cycle’. It is a four-phase cycle comprising of the phases of prosperity, recession, depression, and recovery. The cyclic variation may be regular are not periodic. The upswings and the downswings in business depend upon the joint nature of the economic forces and the interaction between them.

3.2.4. **Random or Irregular Movements:** There is another factor which causes the variation in the variable under study. They are not regular variations and are purely random or irregular. These fluctuations are unforeseen, uncontrollable, unpredictable, and are erratic. These forces are earthquakes, wars, flood, famines, and any other disasters.

3.3. Goals of time series analysis:

- Descriptive: Identify patterns in correlated data trends and seasonal variation.
- Explanation: Understanding and modelling the data.
- Forecasting: Prediction of short-term trends from previous patterns.
- Intervention analysis: How does a single event change the time series.
- Quality control: Deviations of a specified size indicate a problem.

3.4. Uses of Time Series:

- The most important use of studying time series is that it helps us to predict the future behavior of the variable based on past experience
- It is helpful for business planning as it helps in comparing the actual current performance with the expected one.
- From time series, we get to study the past behavior of the phenomenon or the variable under consideration.
- We can compare the changes in the values of different variables at different times or places, etc.

3.5. **Time Series Forecasting:** It means to forecast or to predict the future value over a period of time. It entails developing models based on previous data and applying them to make observations and guide future strategic decisions. The future is forecast or estimated based on what has already happened. Time series adds a time order dependence between observations. This dependence is both a constraint and a structure that provides a source of additional information. Before we discuss time series forecasting methods, let's define time series forecasting more closely. Time series forecasting is a technique for the prediction of events through a sequence of time. It predicts future events by analyzing the trends of the past, on the assumption that future trends will hold similar to historical trends. It is used across many fields of study in various applications including: Astronomy, Business planning, Control engineering, Earthquake prediction, Econometrics, Mathematical finance, Pattern recognition, Resources allocation, Signal processing, Statistics, Weather forecasting.

3.6. **Methods of Forecasting:**

3.6.1. **Decompositional Models:** Time series data can exhibit a variety of patterns, so it is often helpful to split a time series into components, each representing an underlying pattern category. This is what decompositional models do.
The decomposition of time series is a statistical task that deconstructs a time series into several components, each representing one of the underlying categories of patterns. When we decompose a time series into components, we think of a time series as comprising three components: a trend component, a seasonal component, and residuals or ”noise” (containing anything else in the time series).

3.6.1.1. **Additive Decomposition:** If y_t is the time series value at time t. T_t, S_t, C_t, and R_t are the trend value, seasonal, cyclic and random fluctuations at time t respectively. According to the Additive Model, a time series can be expressed as

$$y_t = T_t + S_t + C_t + R_t.$$

3.6.1.2. **Multiplicative Decomposition:** The multiplicative model assumes that the various components in a time series operate proportionately to each other. According to this mode

$$y_t = T_t \times S_t \times C_t \times R_t$$

3.6.2. **Exponential Smoothing Models:** Exponential smoothing is a rule of thumb technique for smoothing time series data using the exponential window function. Exponential smoothing is an easily learned and easily applied procedure for making some determination based on prior assumptions by the user, such as seasonality.

3.6.2.1. **Simple Exponential Method:** **S**imple Exponential Smoothing(SES) is a time series forecasting method for univariate data without a trend or seasonality.
It requires a single parameter, called alpha (α), also called the smoothing factor or smoothing coefficient. This parameter controls the rate at which the influence of the observations at prior time steps decay exponentially. Alpha is often set to a value between 0 and 1. Large values mean that the model pays attention mainly to the most recent past observations, whereas smaller values mean more of the history is considered when making a prediction.

3.6.2.2. **Double Exponential (Holt's Method):** Double Exponential Smoothing is an extension to Exponential Smoothing that explicitly adds support for trends in the univariate time series.

In addition to the alpha parameter for controlling smoothing factor for the level, an additional smoothing factor is added to control the decay of the influence of the change in trend called beta (β). The method supports trends that change in different ways: an additive and a multiplicative, depending on whether the trend is linear or exponential respectively.

Double Exponential Smoothing with an additive trend is classically referred to as Holt's linear trend model, named for the developer of the method Charles Holt.

3.6.2.3. **Triple Exponential Method (Holt-winters' Method):** Triple Exponential Smoothing is an extension of Exponential Smoothing that explicitly adds support for seasonality to the univariate time series. This method is sometimes called Holt-Winters Exponential Smoothing, named for two contributors to the method: Charles Holt and Peter Winters.

In addition to the alpha and beta smoothing factors, a new parameter is added called gamma (γ) that controls the influence on the seasonal component. As with the trend, the seasonality may be modelled as either an additive or multiplicative process for a linear or exponential change in the seasonality. Triple exponential smoothing is the most advanced variation of exponential smoothing and through configuration, it can also develop double and single exponential smoothing models.

3.6.3. **ARIMA Model:** An ARIMA model can be understood by outlining each of its components as follows:

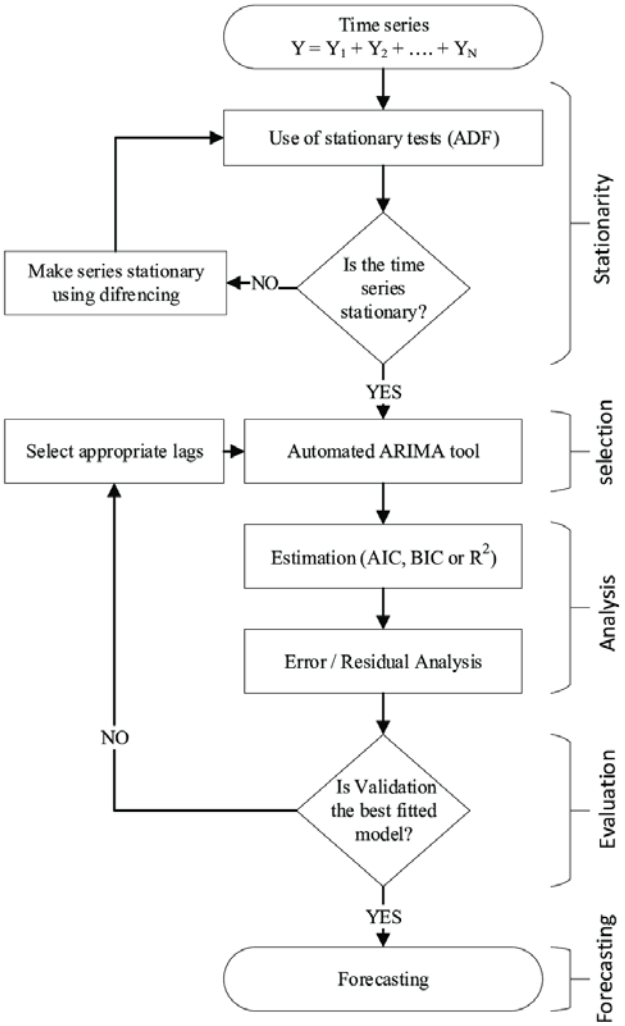
- Autoregression (AR): refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- Integrated (I): represents the differencing of raw observations to allow for the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values).
- Moving average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA Parameters: Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

- *p*: the number of lag observations in the model; also known as the lag order.
- *d*: the number of times that the raw observations are differenced; also known as the degree of differencing.
- q: the size of the moving average window; also known as the order of the moving average.

In a linear regression model, for example, the number and type of terms are included. A 0 value, which can be used as a parameter, would mean that particular component should not be used in the model. This way, the ARIMA model can be constructed to perform the function of an ARMA model, or even simple AR, I, or MA models.

ARIMA and Stationarity: In an autoregressive integrated moving average model, the data are differenced in order to make it stationary. A model that shows stationarity is one that shows there is constancy to the data over time. Most economic and market data show trends, so the purpose of differencing is to remove any trends or seasonal structures.



4. METHEDOLOGY:

The methodology adopted in this study of "FORECASTING NIFTY 50 BENCHMARK INDEX" are as follows:

i. Importing Libraries
ii. Importing Data
iii. Converting to Time Series Object
iv. Plotting Data and Trend Line
v. Analysing Seasonal Component
vi. Decomposition
vii. Simple Exponential Smoothing
viii. Double Exponential Smoothing
ix. Triple Exponential Smoothing
x. ARIMA Model
xi. Comparing Models

4.1. Importing Libraries

4.1.1. Code

```
#### Load the Libraries
library(forecast)
library(graphics)
library(datasets)
library(tseries)
library(ggplot2)
library(fpp2)
```

4.2. Importing Data

4.2.1. Code and Output

```
#### Import Data
nifty50 = scan("S:/Data.txt")
Read 312 items
> class(nifty50)
[1] "numeric"
```

4.3. Converting to Time Series Object

4.3.1. Code and Output

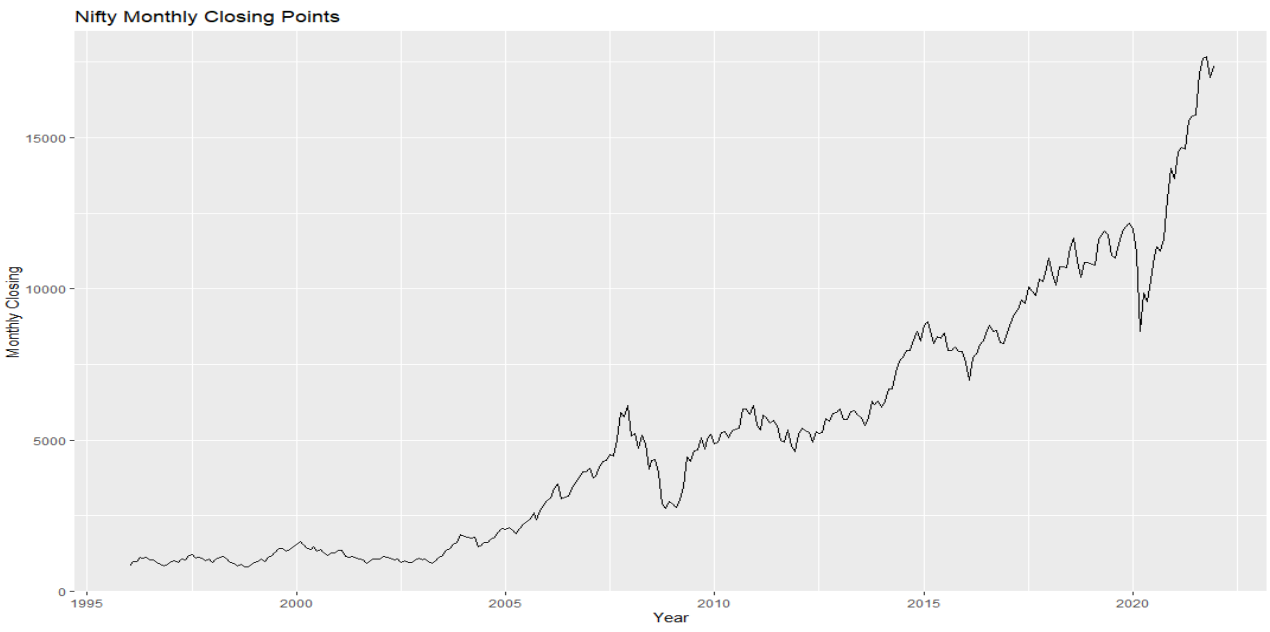
```
nifty50.ts <- ts(nifty50, frequency=12, start=(1996))
nifty50.ts
```

```
        Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
1996  848.42   992.51   985.30  1114.36  1089.92  1122.00  1042.81  1029.00   943.96   909.29   830.32   899.10
1997  972.65   998.65   968.30  1079.85  1050.90  1192.40  1221.50  1105.00  1123.80  1085.25  1023.95  1079.40
1998  963.45  1060.75  1116.90  1159.35  1063.15   941.65   931.40   852.80   904.95   824.55   817.75   884.25
1999  966.20   981.30  1078.05   978.20  1132.30  1187.70  1310.15  1412.00  1413.10  1325.45  1376.15  1480.45
2000 1546.20  1654.80  1528.45  1406.55  1380.45  1471.45  1332.85  1394.10  1271.65  1172.75  1268.15  1263.55
2001 1371.70  1351.40  1148.20  1125.25  1167.90  1107.90  1072.85  1053.75   913.85   971.90  1067.15  1059.05
2002 1075.40  1142.05  1129.55  1084.50  1028.80  1057.80   958.90  1010.60   963.15   951.40  1050.15  1093.50
2003 1041.85  1063.40   978.20   934.05  1006.80  1134.15  1185.85  1356.55  1417.10  1555.90  1615.25  1879.75
2004 1809.75  1800.30  1771.90  1796.10  1483.60  1505.60  1632.30  1631.75  1745.50  1786.90  1958.80  2080.50
2005 2057.60  2103.25  2035.65  1902.50  2087.55  2220.60  2312.30  2384.65  2601.40  2370.95  2652.25  2836.55
2006 3001.10  3074.70  3402.55  3557.60  3071.05  3128.20  3143.20  3413.90  3588.40  3744.10  3954.50  3966.40
2007 4082.70  3745.30  3821.55  4087.90  4295.80  4318.30  4528.85  4464.00  5021.35  5900.65  5762.75  6138.60
2008 5137.45  5223.50  4734.50  5165.90  4870.10  4040.55  4332.95  4360.00  3921.20  2885.60  2755.10  2959.15
2009 2874.80  2763.65  3020.95  3473.95  4448.95  4291.10  4636.45  4662.10  5083.95  4711.70  5032.70  5201.05
2010 4882.05  4922.30  5249.10  5278.00  5086.30  5312.50  5367.60  5402.40  6029.95  6017.70  5862.70  6134.50
2011 5505.90  5333.25  5833.75  5749.50  5560.15  5647.40  5482.00  5001.00  4943.25  5326.60  4832.05  4624.30
2012 5199.25  5385.20  5295.55  5248.15  4924.25  5278.90  5229.00  5258.50  5703.30  5619.70  5879.85  5905.10
2013 6034.75  5693.05  5682.55  5930.20  5985.95  5842.20  5742.00  5471.80  5735.30  6299.15  6176.10  6304.00
2014 6089.50  6276.95  6704.20  6696.40  7229.95  7611.35  7721.30  7954.35  7964.80  8322.20  8588.25  8282.70
2015 8808.90  8901.85  8491.00  8181.50  8433.65  8368.50  8532.85  7971.30  7948.90  8065.80  7935.25  7946.35
2016 7563.55  6987.05  7738.40  7849.80  8160.10  8287.75  8638.50  8786.20  8611.15  8625.70  8224.50  8185.80
2017 8561.30  8879.60  9173.75  9304.05  9621.25  9520.90 10077.10  9917.90  9788.60 10335.30 10226.55 10530.70
2018 11027.70 10492.85 10113.70 10739.35 10736.15 10714.30 11356.50 11680.50 10930.45 10386.60 10876.75 10862.55
2019 10830.95 10792.50 11623.90 11748.15 11922.80 11788.85 11118.00 11023.25 11474.45 11877.45 12056.05 12168.45
2020 11962.10 11201.75  8597.75  9859.90  9580.30 10302.10 11073.45 11387.50 11247.55 11642.40 12968.95 13981.75
2021 13634.60 14529.15 14690.70 14631.10 15582.80 15721.50 15763.05 17132.20 17618.15 17671.65 16983.20 17354.05
```

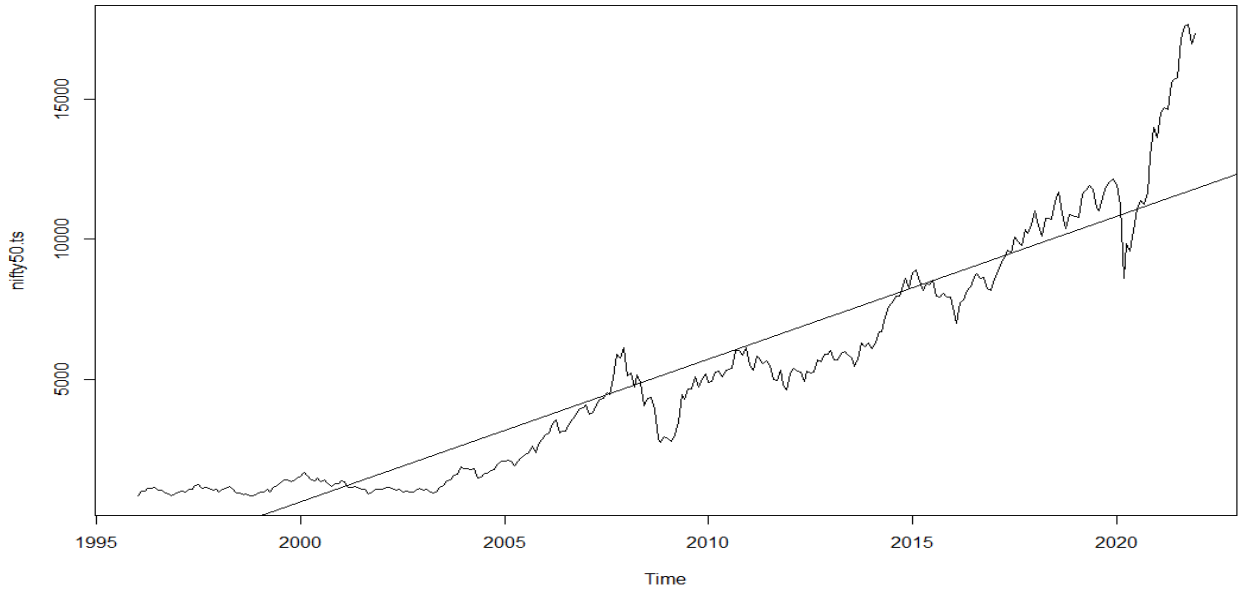
4.4. Plotting Data and Trend Line

4.4.1. Code and Output

```
autoplot(nifty50.ts) +
  ggtitle("Nifty Monthly Closing Points") +
  xlab("Year") +
  ylab("Monthly Closing")
```



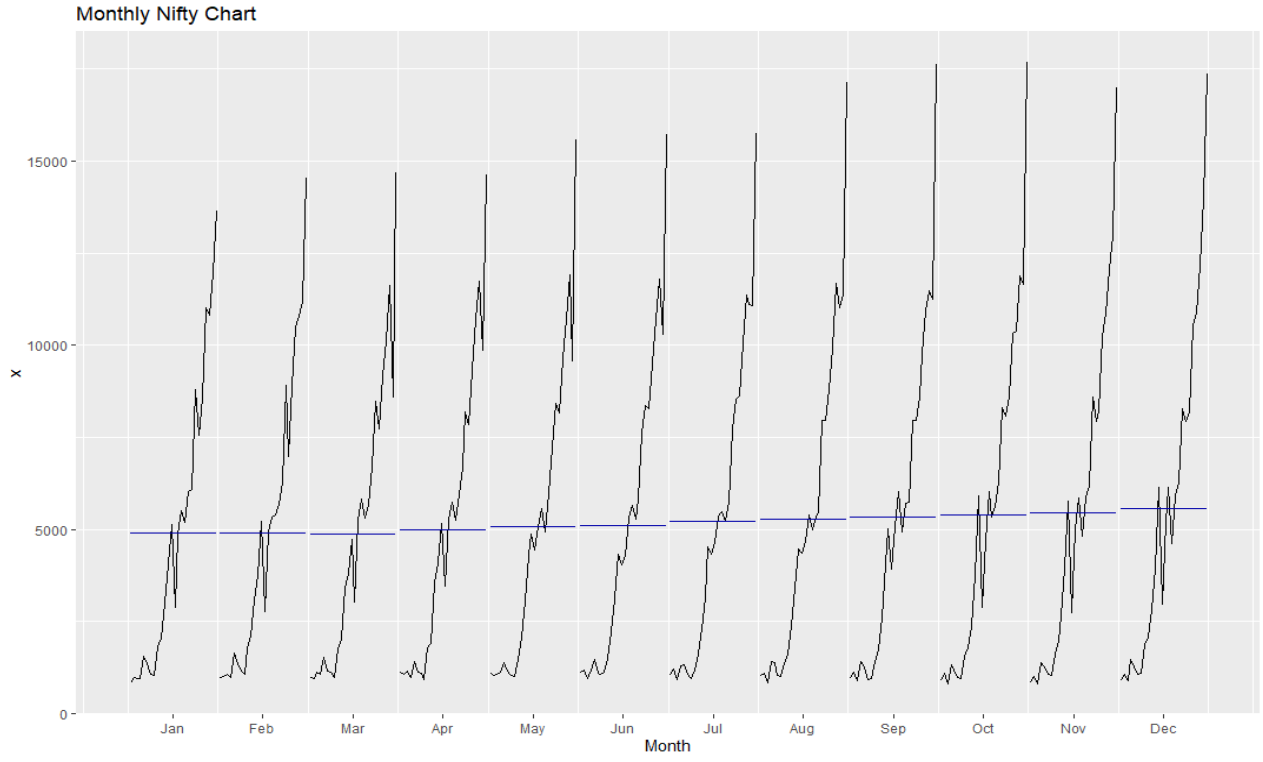
```
#### Plotting Trend line
plot(nifty50.ts)
abline(reg=lm(nifty50.ts ~ time(nifty50.ts)))
```



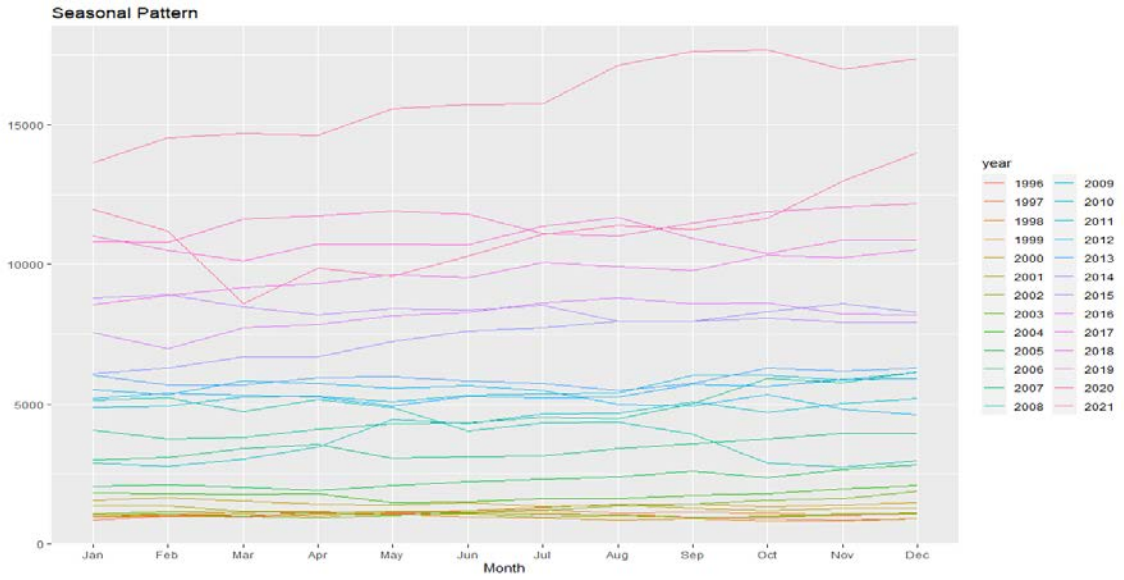
4.5. Analysing Seasonal Component

4.5.1. Code and Output

```
ggmonthplot(nifty50.ts) +
  ggtitle("Monthly Nifty Chart")
```



```
#### Checking Seasonal Pattern
ggseasonplot(nifty50.ts) +
  ggtitle("Seasonal Pattern")
```



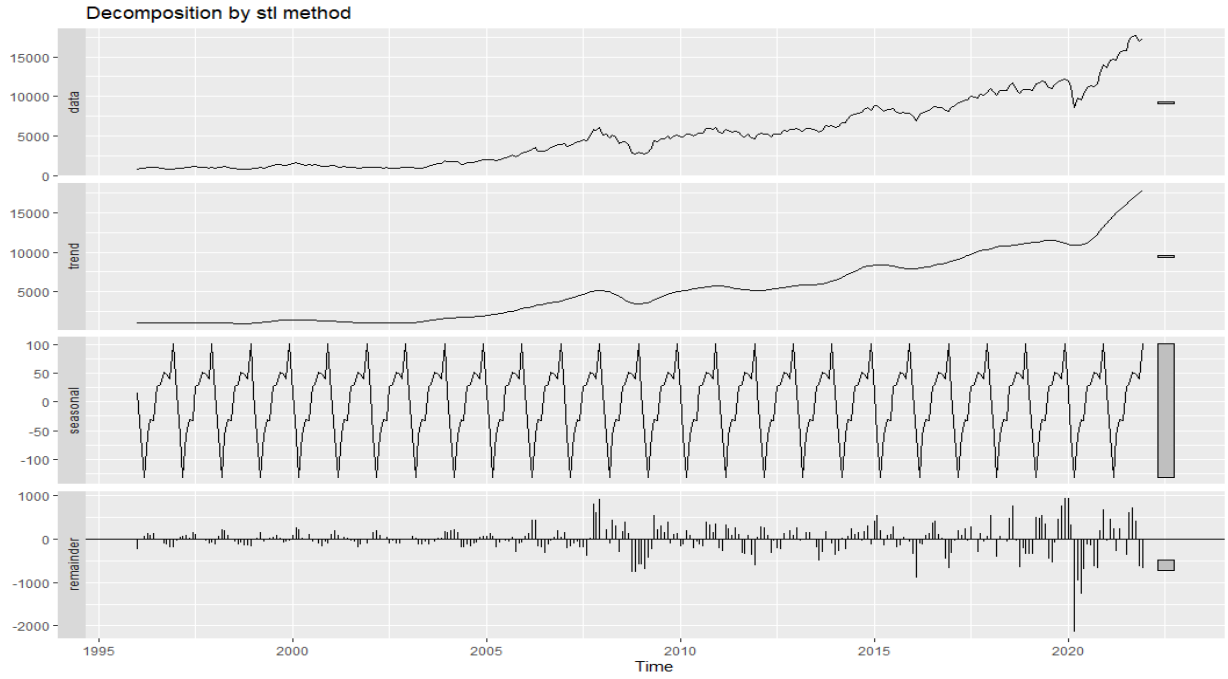
4.5.2. Interpretation

From the Nifty Monthly Chart and Seasonal Pattern chart, it is seen that there are no fixed seasonal pattern and even the means of every month are almost similar. It is because Nifty 50 comprises of 13 different sectors which counterbalances each other.

4.6. Decomposition

4.6.1. Code and Output

```
nifty50.ts.decomposition = stl(nifty50.ts,s.window="p")
autoplot(nifty50.ts.decomposition,main="Decomposition by stl method")
```

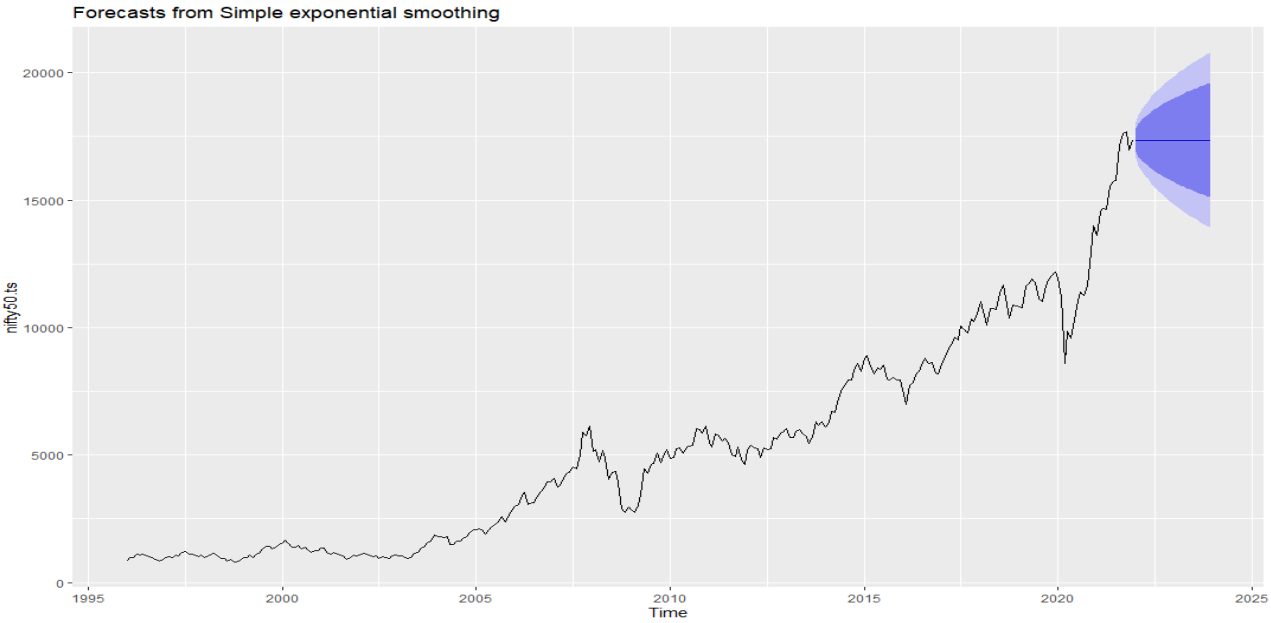


4.7. Simple Exponential Smoothing

4.7.1. Code and Output

```
nifty50ses = ses(nifty50.ts, h=24)
autoplot(nifty50ses)
nifty50ses$model
summary(nifty50ses)
```

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = nifty50.ts, h = 24)

  Smoothing parameters:
    alpha = 0.9999

  Initial states:
    l = 848.8677

  sigma:  356.9835

     AIC     AICc      BIC
5463.489 5463.567 5474.718

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 52.90642 355.8375 225.2684 0.7315109 5.207261 0.2412154 0.005275668

              Forecasts:
                       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
              Jan 2022       17354.01 16896.52 17811.51 16654.34 18053.69
              Feb 2022       17354.01 16707.05 18000.97 16364.57 18343.45
              Mar 2022       17354.01 16561.67 18146.36 16142.22 18565.80
              Apr 2022       17354.01 16439.10 18268.93 15954.77 18753.26
              May 2022       17354.01 16331.11 18376.92 15789.62 18918.41
              Jun 2022       17354.01 16233.48 18474.54 15640.31 19067.72
              Jul 2022       17354.01 16143.70 18564.32 15503.01 19205.02
              Aug 2022       17354.01 16060.14 18647.88 15375.21 19332.82
              Sep 2022       17354.01 15981.66 18726.37 15255.18 19452.85
              Oct 2022       17354.01 15907.42 18800.60 15141.65 19566.38
              Nov 2022       17354.01 15836.82 18871.21 15033.67 19674.36
              Dec 2022       17354.01 15769.36 18938.67 14930.49 19777.53
              Jan 2023       17354.01 15704.65 19003.37 14831.53 19876.49
              Feb 2023       17354.01 15642.39 19065.63 14736.31 19971.71
              Mar 2023       17354.01 15582.32 19125.71 14644.44 20063.59
              Apr 2023       17354.01 15524.21 19183.81 14555.58 20152.45
              May 2023       17354.01 15467.90 19240.13 14469.45 20238.57
              Jun 2023       17354.01 15413.22 19294.81 14385.83 20322.20
              Jul 2023       17354.01 15360.04 19347.99 14304.49 20403.53
              Aug 2023       17354.01 15308.24 19399.79 14225.27 20482.76
              Sep 2023       17354.01 15257.72 19450.31 14148.01 20560.02
              Oct 2023       17354.01 15208.39 19499.64 14072.56 20635.46
              Nov 2023       17354.01 15160.17 19547.86 13998.81 20709.21
              Dec 2023       17354.01 15112.98 19595.05 13926.65 20781.38
```



4.8. Double Exponential Smoothing

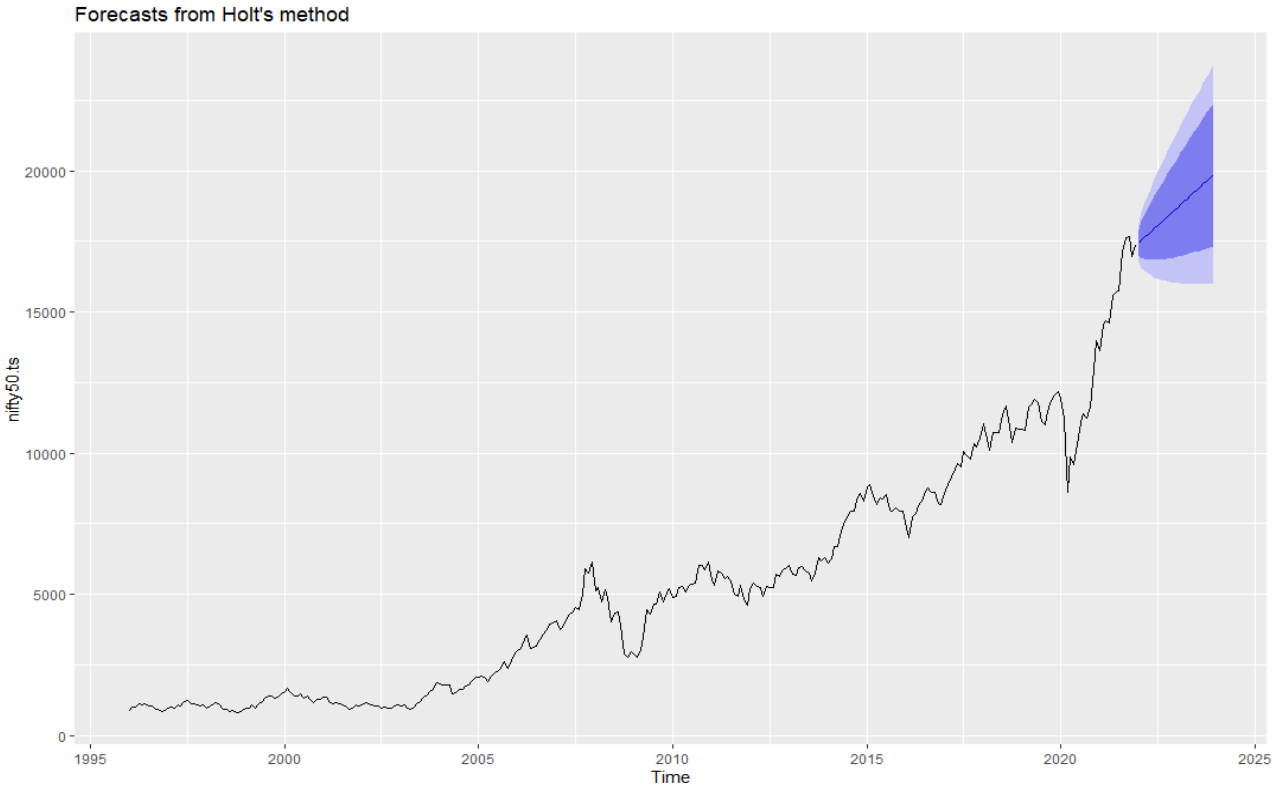
4.8.1. Code and Output

```
nifty50holt = holt(nifty50.ts, h=24)
autoplot(nifty50holt)
nifty50holt$model
summary(nifty50holt)
```

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = nifty50.ts, h = 24)

  Smoothing parameters:
    alpha = 0.995
    beta  = 0.0123

  Initial states:
    l = 998.1663
    b = 0.7852

  sigma:  354.7005

     AIC     AICc      BIC
5461.466 5461.662 5480.181

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 27.05172 352.4194 222.6622 0.2811965 5.227764 0.2384247 -0.005741335
```

Forecasts:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2022	17457.11	17002.54	17911.68	16761.91	18152.31
Feb 2022	17561.51	16916.32	18206.70	16574.78	18548.24
Mar 2022	17665.90	16871.52	18460.29	16451.00	18880.81
Apr 2022	17770.30	16847.79	18692.81	16359.44	19181.16
May 2022	17874.70	16837.26	18912.14	16288.07	19461.32
Jun 2022	17979.09	16835.91	19122.27	16230.75	19727.43
Jul 2022	18083.49	16841.38	19325.60	16183.85	19983.13
Aug 2022	18187.89	16852.13	19523.64	16145.02	20230.75
Sep 2022	18292.28	16867.09	19717.48	16112.64	20471.93
Oct 2022	18396.68	16885.49	19907.87	16085.52	20707.84
Nov 2022	18501.08	16906.76	20095.39	16062.78	20939.37
Dec 2022	18605.47	16930.45	20280.49	16043.75	21167.20
Jan 2023	18709.87	16956.21	20463.53	16027.88	21391.85
Feb 2023	18814.27	16983.76	20644.77	16014.75	21613.78
Mar 2023	18918.66	17012.86	20824.46	16003.99	21833.34
Apr 2023	19023.06	17043.32	21002.80	15995.31	22050.81
May 2023	19127.45	17074.98	21179.93	15988.46	22266.45
Jun 2023	19231.85	17107.70	21356.01	15983.24	22480.46
Jul 2023	19336.25	17141.36	21531.14	15979.45	22693.04
Aug 2023	19440.64	17175.86	21705.43	15976.96	22904.33
Sep 2023	19545.04	17211.11	21878.97	15975.61	23114.47
Oct 2023	19649.44	17247.04	22051.83	15975.29	23323.59
Nov 2023	19753.83	17283.57	22224.09	15975.89	23531.77
Dec 2023	19858.23	17320.65	22395.81	15977.33	23739.13

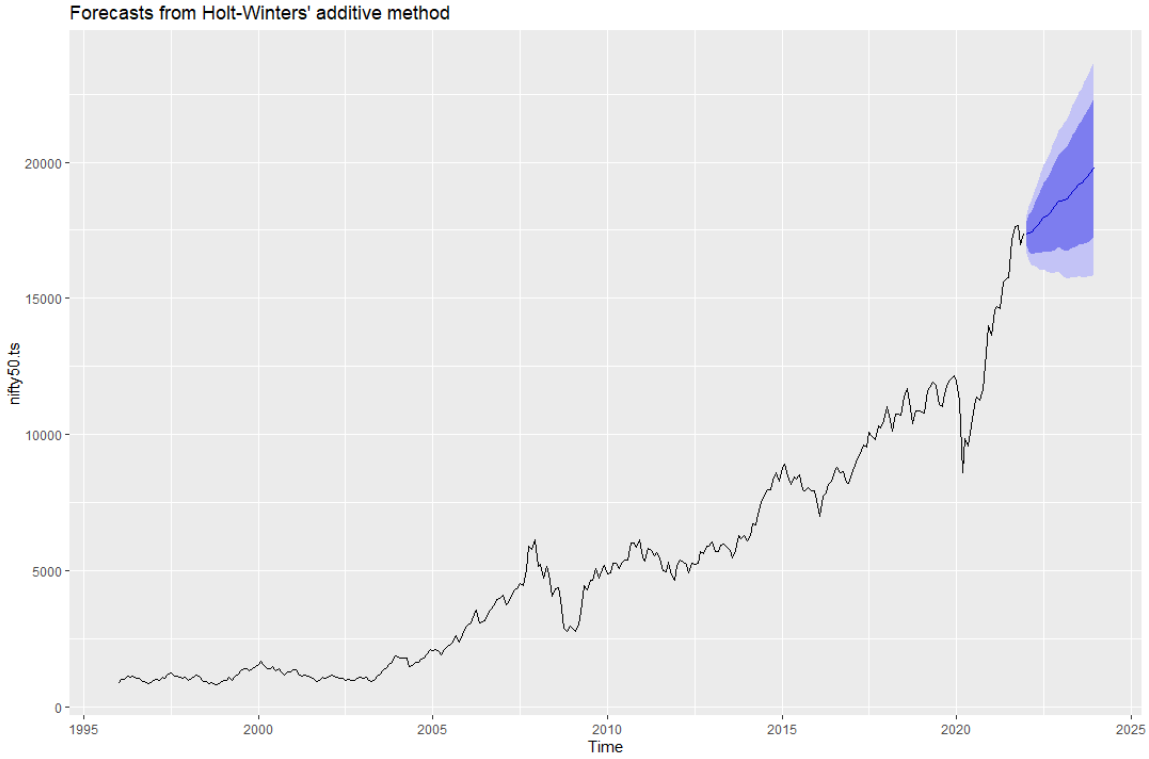


4.9. Triple Exponential Smoothing

4.9.1. Code and Output

```
nifty50hw = hw(nifty50.ts, h=24)
autoplot(nifty50hw)
nifty50hw$model
summary(nifty50hw)
```

```
Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = nifty50.ts, h = 24)

  Smoothing parameters:
    alpha = 0.9922
    beta  = 0.0119
    gamma = 1e-04

  Initial states:
    l = 1085.5495
    b = 3.392
    s = 117.6393 48.3192 11.5431 -0.4181 -7.0746 36.68
           -32.2779 -20.8866 -44.0266 -109.6259 -38.2073 38.3354

  sigma:  358.9168

     AIC     AICc      BIC
5480.441 5482.522 5544.072

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 26.39176 349.5927 226.1572 0.2070802 5.661505 0.2421671 -0.005442935

         Forecasts:
                  Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
         Jan 2022       17374.50 16914.53 17834.47 16671.04 18077.97
         Feb 2022       17399.32 16747.49 18051.16 16402.43 18396.22
         Mar 2022       17429.20 16627.15 18231.26 16202.57 18655.84
         Apr 2022       17596.12 16665.08 18527.15 16172.22 19020.01
         May 2022       17720.55 16673.85 18767.24 16119.77 19321.33
         Jun 2022       17810.47 16657.40 18963.55 16047.00 19573.95
         Jul 2022       17980.68 16728.13 19233.24 16065.06 19896.31
         Aug 2022       18038.40 16691.70 19385.11 15978.80 20098.01
         Sep 2022       18146.28 16709.70 19582.86 15949.22 20343.34
         Oct 2022       18259.50 16736.54 19782.46 15930.34 20588.67
         Nov 2022       18397.51 16791.08 20003.94 15940.69 20854.34
         Dec 2022       18568.25 16880.81 20255.69 15987.54 21148.97
         Jan 2023       18590.21 16823.85 20356.57 15888.80 21291.62
         Feb 2023       18615.03 16771.59 20458.47 15795.73 21434.33
         Mar 2023       18644.91 16725.97 20563.85 15710.15 21579.67
         Apr 2023       18811.82 16818.78 20804.87 15763.72 21859.93
         May 2023       18936.26 16870.32 21002.19 15776.69 22095.83
         Jun 2023       19026.18 16888.45 21163.91 15756.81 22295.56
         Jul 2023       19196.39 16987.83 21404.95 15818.69 22574.09
         Aug 2023       19254.11 16975.58 21532.64 15769.40 22738.82
         Sep 2023       19361.99 17014.26 21709.71 15771.45 22952.52
         Oct 2023       19475.21 17058.99 21891.44 15779.91 23170.51
         Nov 2023       19613.22 17129.12 22097.32 15814.12 23412.32
         Dec 2023       19783.96 17232.55 22335.37 15881.91 23686.01
```

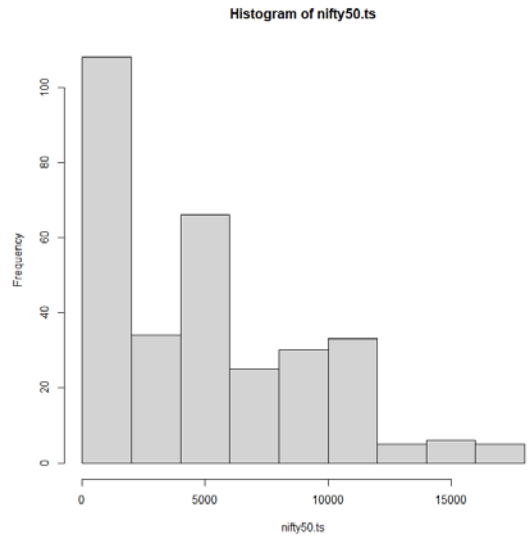


4.10. ARIMA Model

4.10.1. Visual Test for Stationarity

4.10.1.1. Code and Output

```
hist(nifty50.ts)
```



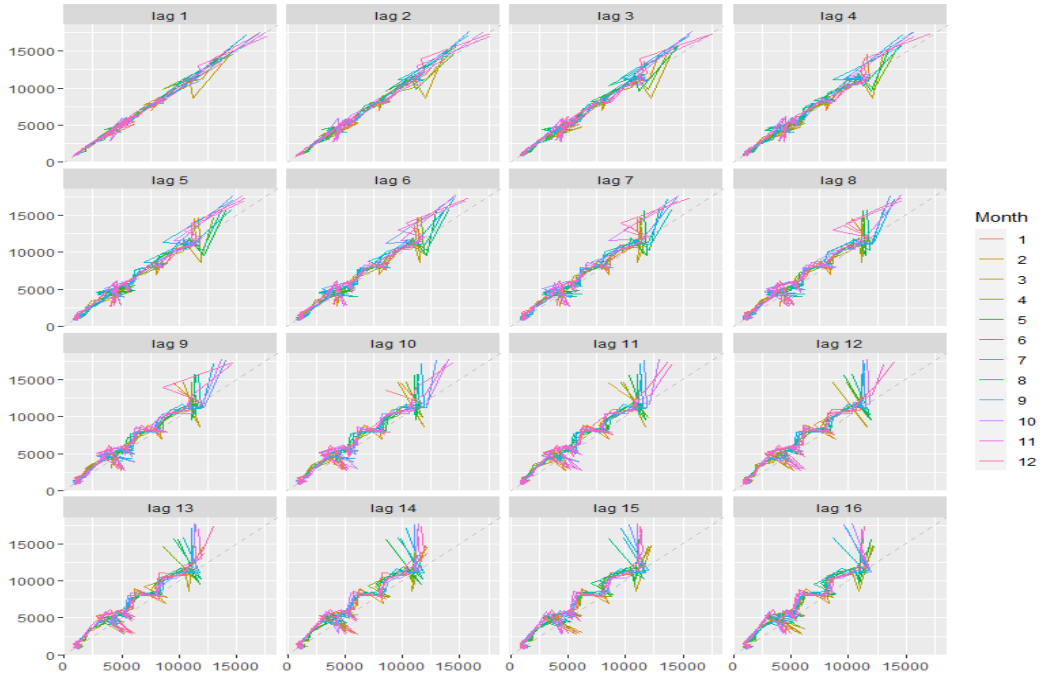
4.10.1.2. Interpretation

As the data is positively skewed, it can be concluded that the time series is non-stationary.

4.10.2. Lag Plot

4.10.2.1. Code and Output

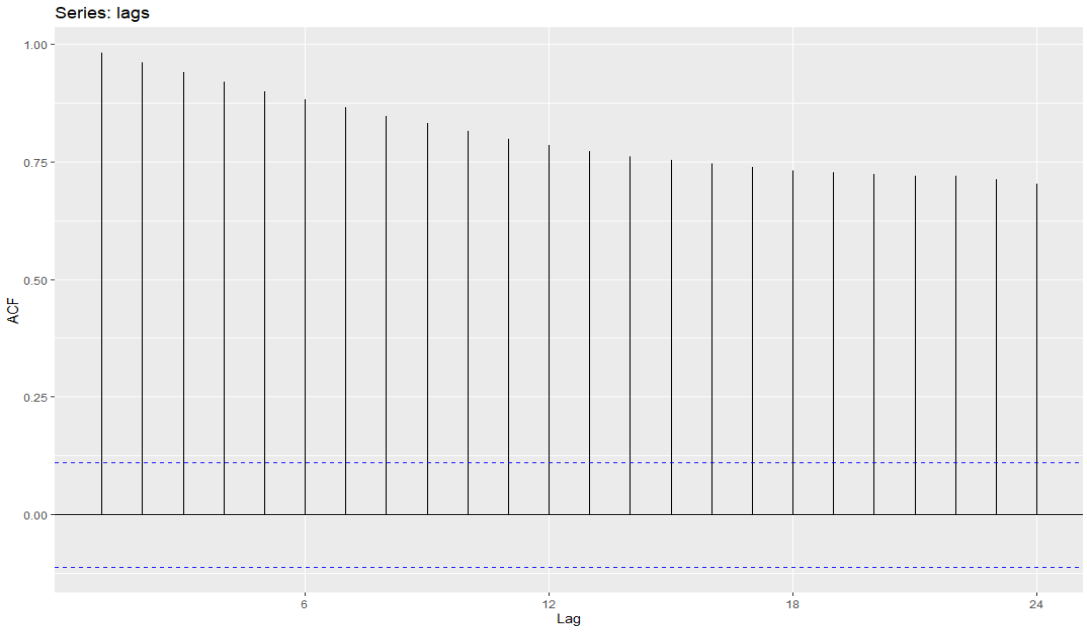
```
#### Lag Plots - each lag corresponds to the correlation coefficient
lags = window(nifty50.ts)
gglagplot(lags)
```



4.10.3. Auto Correlation Function

4.10.3.1. Code and Output

```
#### ACF of original time series
ggAcf(lags)
```



4.10.3.2. Interpretation

Here the lag lines are decaying but at a much slower rate. All the lag lines exceeded the statistical limits which infer that there is correlationship between the variables. Hence, the time series is non-stationary.

4.10.4. Augmented Dickey-Fuller Test (I)

4.10.4.1. Code and Output

```
# Null Hypothesis H0: Time series non-stationary
# Alternative hypothesis H1: Time series is stationary
adf.test(nifty50.ts)
```

```
        Augmented Dickey-Fuller Test

data:  nifty50.ts
Dickey-Fuller = -0.7987, Lag order = 6, p-value = 0.961
alternative hypothesis: stationary
```

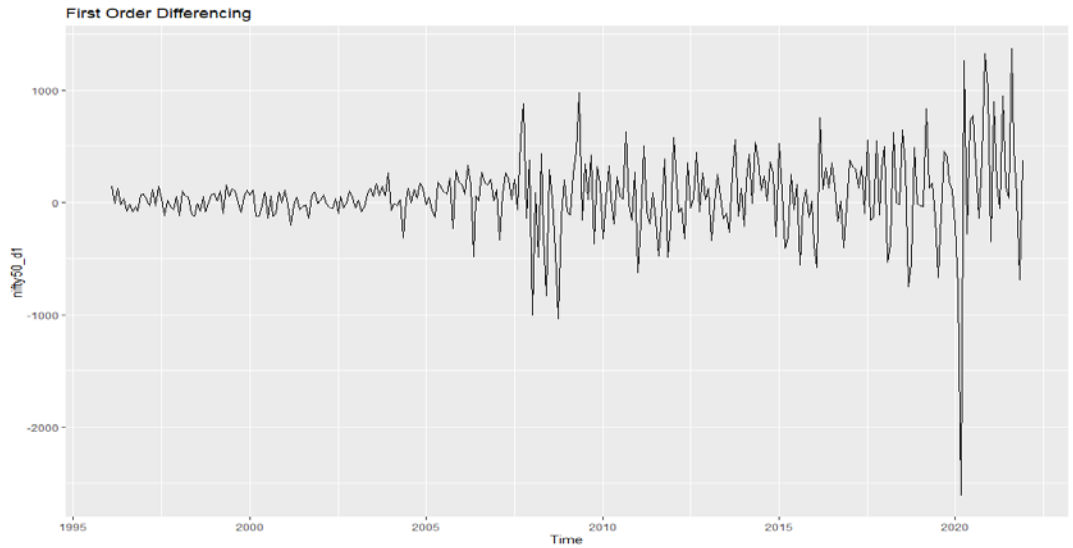
4.10.4.2. Interpretation:

The p-value is 0.961 so the null hypothesis can not be rejected, hence the time series is non-stationary.

4.10.5. Non-stationary to stationary

4.10.5.1. Code and Output

```
# Differencing
# Remove the unequal variance and remove the trending component
nifty50_d1 = diff(nifty50.ts, differences = 1)
autoplot(nifty50_d1,main="First Order Differencing")
```



4.10.6. Augmented Dickey-Fuller Test (II)

4.10.6.1. Code and Output

```
# Null Hypothesis H0: Time series non-stationary
# Alternative hypothesis H1: Time series is stationary
adf.test(nifty50_d1,alternative = "stationary", k=12)
```

```
        Augmented Dickey-Fuller Test

data:  nifty50_d1
Dickey-Fuller = -5.362, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary
```

4.10.6.2. Interpretation

The p-value is 0.01 so the null hypothesis can be rejected and the time series is stationary. So, now we can implement ARIMA method.

4.10.7. Creating Training and Test Dataset

4.10.7.1. Code and Output

```
tsdatatrain = window(nifty50.ts, end=c(2020,12), frequency = 12) #Train Data Set
tsdatatest = window(nifty50.ts, start=c(2021,1), frequncy = 12) #Test Data Set
```

4.10.8. Implementing Auto-Arima

4.10.8.1. Code and Output

```
#### Auto Arima Model Nifty 50
autoarima.fit.train = auto.arima(tsdatatrain, seasonal = TRUE, ic="bic")
autoarima.fit.train
```

```
Series: tsdatatrain
ARIMA(0,1,0)(0,0,2)[12]

Coefficients:
         sma1    sma2
      -0.1654  0.2444
s.e.   0.0640  0.0699

sigma^2 = 110636:  log likelihood = -2160.4
AIC=4326.8   AICc=4326.88   BIC=4337.9
```

4.10.9. Checking Accuracy of the Model

4.10.9.1. Code and Output

```
accuracy(autoarima.fit.train.forecast,tsdatatest)
```

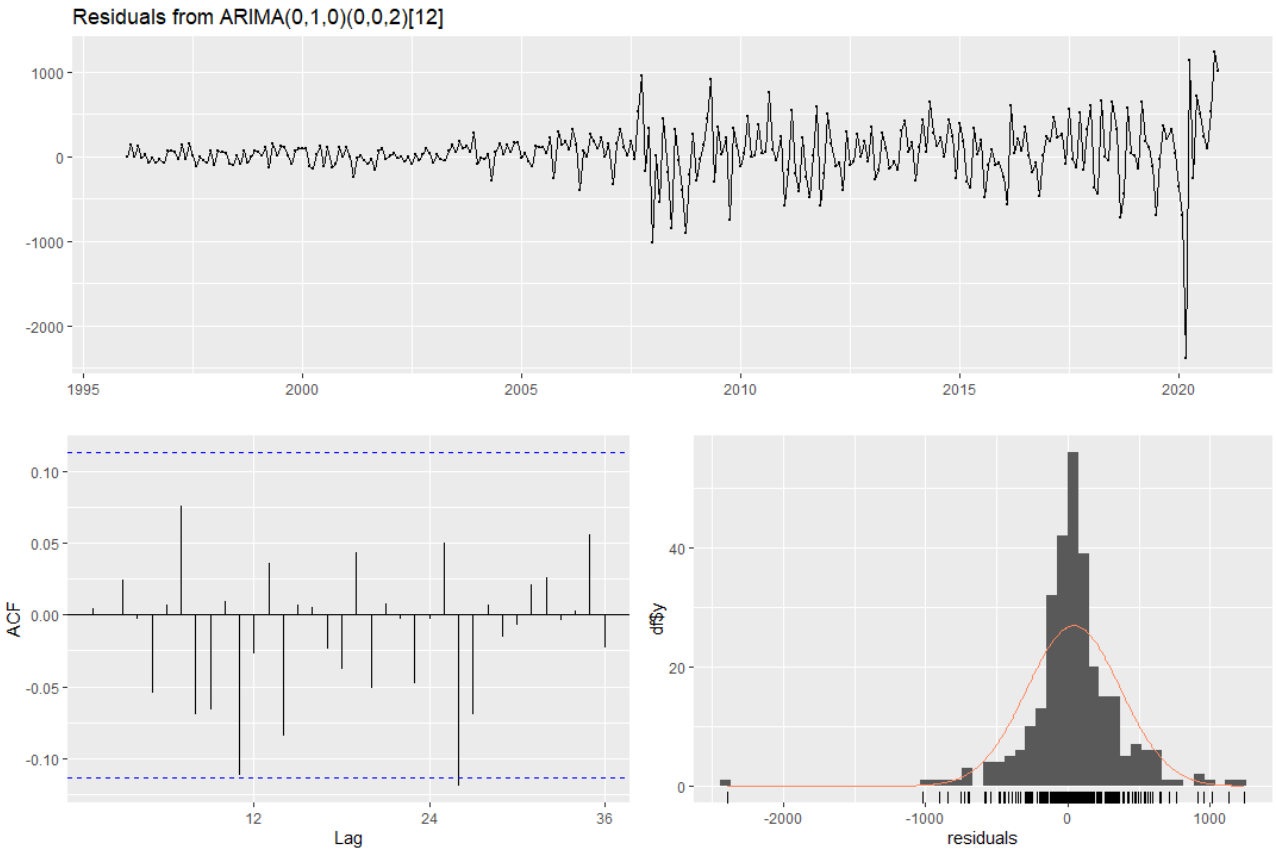
```
                    ME      RMSE       MAE       MPE     MAPE      MASE        ACF1 Theil's U
Training set   41.65266  330.9528  212.4121 0.6484633 5.284147 0.2747447 0.004039922        NA
Test set     1379.39550 1947.1749 1460.8688 8.1308874 8.728436 1.8895630 0.693385248  2.819717
```

4.10.10. Checking Residuals

4.10.10.1. Code and Output

```
checkresiduals(autoarima.fit.train.forecast)
```

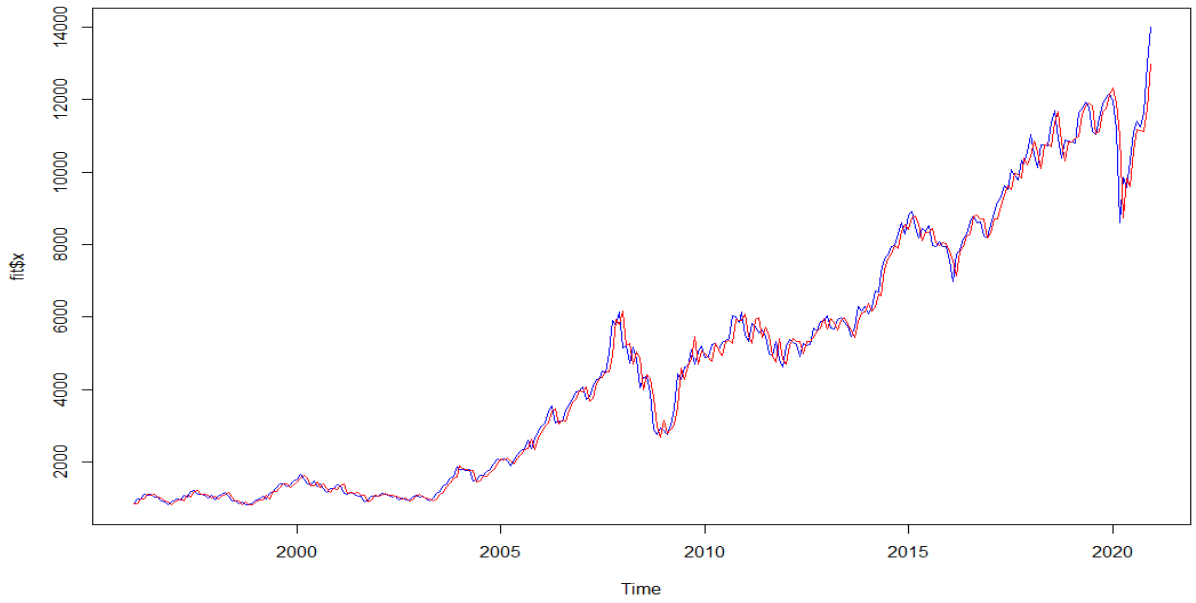
```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,0)(0,0,2)[12]
Q* = 15.473, df = 22, p-value = 0.8412

Model df: 2.   Total lags used: 24
```



4.10.11. Fitting Arima Model

4.10.11.1. Code and Output

```
fit = Arima(tsdatatrain, c(0,1,0), seasonal = list(order=c(0,0,2),period=12))
plot(fit$x, col="blue")
lines(fit$fitted, col="red", main = "NIfty50 Actual vs Forecast")
```

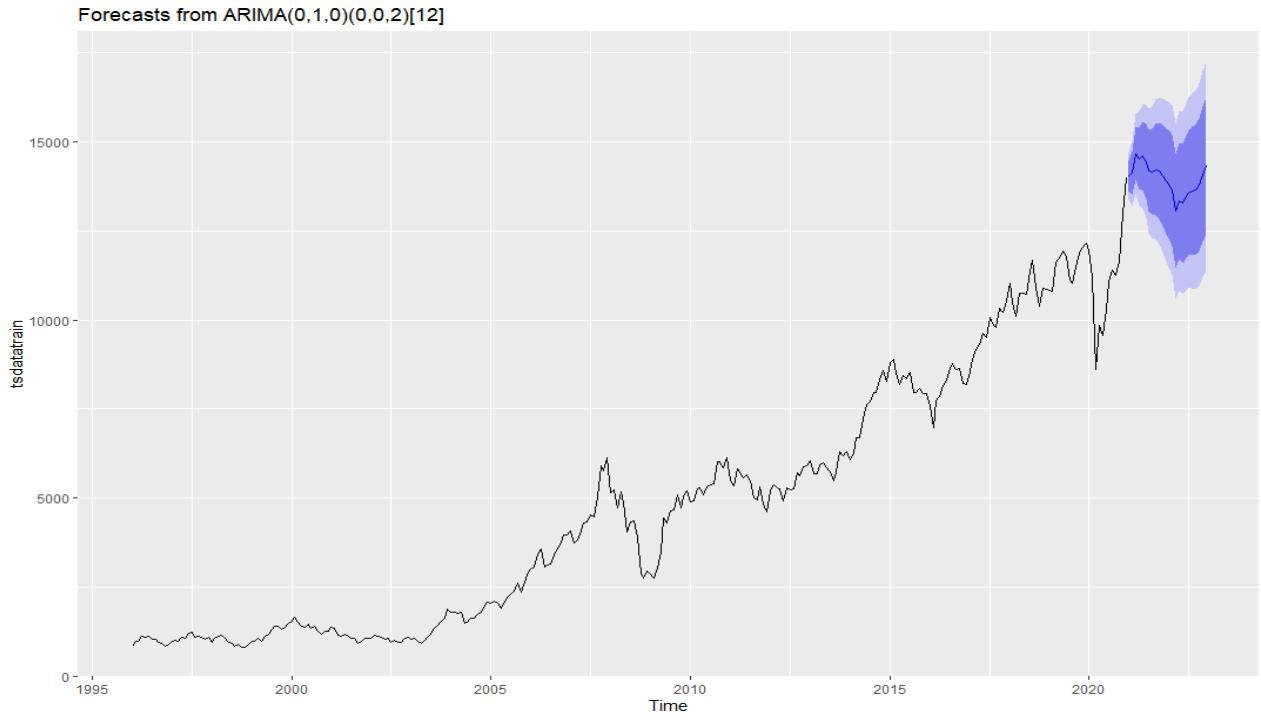


4.10.12. Forecasting using ARIMA

4.10.12.1. Code and Output

```
forecast(fit, h=12)
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2021	14041.97	13615.70	14468.24	13390.04	14693.89
Feb 2021	14121.96	13519.12	14724.79	13200.00	15043.92
Mar 2021	14674.14	13935.82	15412.46	13544.98	15803.30
Apr 2021	14530.28	13677.74	15382.82	13226.44	15834.13
May 2021	14599.41	13646.24	15552.58	13141.66	16057.15
Jun 2021	14451.98	13407.84	15496.12	12855.10	16048.86
Jul 2021	14198.36	13070.56	15326.17	12473.54	15923.19
Aug 2021	14151.89	12946.22	15357.56	12307.97	15995.81
Sep 2021	14224.81	12946.00	15503.62	12269.04	16180.58
Oct 2021	14186.15	12838.16	15534.13	12124.58	16247.71
Nov 2021	14058.32	12644.54	15472.10	11896.13	16220.50
Dec 2021	13900.78	12424.14	15377.42	11642.45	16159.11
Jan 2022	13815.04	12296.15	15333.94	11492.09	16137.99
Feb 2022	13645.26	12085.26	15205.27	11259.44	16031.09
Mar 2022	13061.66	11461.60	14661.72	10614.58	15508.74
Apr 2022	13338.19	11699.05	14977.32	10831.35	15845.03
May 2022	13274.76	11597.46	14952.07	10709.55	15839.98
Jun 2022	13449.46	11734.84	15164.08	10827.18	16071.74
Jul 2022	13570.81	11819.67	15321.95	10892.67	16248.95
Aug 2022	13627.43	11840.52	15414.35	10894.58	16360.28
Sep 2022	13651.34	11829.35	15473.32	10864.85	16437.83
Oct 2022	13782.40	11926.00	15638.80	10943.29	16621.52
Nov 2022	14084.66	12194.48	15974.85	11193.88	16975.45
Dec 2022	14331.59	12408.22	16254.96	11390.05	17273.14



4.11. Comparing Models

Methods/ Accuracy	RMSE	MASE	MAPE
Holt's Method	352.4194	0.2384247	5.227764
Holt's Winter Method	349.5927	0.2421671	5.661505
ARIMA	330.9528	0.2747447	5.284147

4.11.1. Interpretation

Root mean square deviation (RMSE): It is the square root value of the Mean squared error. It gives the output in terms of the metric of the dependent variable. Hence, it is considered more reliable than MSE.
Here, ARIMA Method minimises the RMSE most efficiently.

Mean Absolute Scaled Error (MASE): It is scale-free error metric that gives each error as ratio compared to a baseline's average error.
Here, Holt's Method, Holt's Winter Method minimises the MASE most efficiently.

Mean Absolute Percentage Error: It is the percentage of the average absolute difference between predicted values and true values, divided by the true value.
Here, Holt's Method, ARIMA minimises the MAPE most efficiently.

5. CONCLUSION

We forecasted the Nifty 50 using three distinct models in this article: Holt's Method, Winter Method, and ARIMA Method. ARIMA and Holt's winter approach anticipated the Nifty 50 more accurately. We can observe that the market has always followed an upward trend over a long period of time. We can see that there is no significant seasonality in the data, which supports the assumption that the sectors in the Nifty 50 balance each other. In this case, simple exponential smoothing is inefficient in projecting the Nifty 50. Winter Holts' approach is substantially more accurate than Holt's method since it incorporates seasonality. The Dickey Fuller Test results in a p-value of 0.961, indicating the time series is not stationary, hence we cannot use the ARIMA method directly. Using the differencing method, we transform non-stationary time series to stationary time series. Now that the test's p-value is 0.01, it is clear that the successful conversion of the non-stationary time series to stationary time series. The data set has now been divided into a train and test data set. By comparing the predicted value of the train data set with the test data set, we evaluate the model's accuracy. Using the Ljung-Box Test, we additionally verify that our assumption is correct about the normality of residual. Finally, we employ Auto-Arima to obtain the predicted value for the following 24 months. Using the ARIMA approach, we obtain the most accurate forecast. Due to its inability to predict highly volatile circumstances like pandemics and recessions, even this model is not entirely accurate.

6. REFERENCES

Fundamental of Statistics (Volume-2) by A.M. Gun, M.K. Gupta & B. Dasgupta (Published by D. Chakraborty for The World Press Private Limited, Kolkata)

Applied Multivariate Statistical Analysis 6th edition (Johnson and Wichern)

The Analysis of Time Series (C. Chatfield)

A Little Book of R for Time Series (Avril Coghlan)

Introduction to Linear Regression Analysis 5th edition (Douglas C. Montgomery)

Fundamental of Applied Statistics by S.C. Gupta & V.K. Kapoor (Publisher: Sultan Chand & Sons Educational Publishers, New Delhi)

Time Series Analysis: With Applications in R (Book by Jonathan Cryer and Kung-sik Chan)

https://www.sciencedirect.com/topics/engineering/moving-average

https://towardsdatascience.com/introduction-to-aic-akaike-information-criterion-9c9ba1c96ced

https://www.toppr.com/guides/business-mathematics-and-statistics/time-series-analysis/components-of-time-series/

https://rpubs.com/riazakhan94/arima_with_example

Appendix:
https://drive.google.com/file/d/1jteDZdbaMV7r17RQwqhoTqlj37y5a9Sc/view?usp=sharing

Data Source:

https://in.investing.com/indices/s-p-cnx-nifty-historical-data?end_date=1639506600&interval_sec=monthly&st_date=821730600