Great Learning

# TIME SERIES
# BUSINESS REPORT


## Submitted by:

**N. Aishwarya**
PGP-DSBA

July 2022

# Table of Contents



# List of Figures

# List of Tables

## Executive Summary – Time Series Forecasting:

Time series is defined as an ordered sequence of values of a variable at equally spaced time intervals. Based on the method, we obtain an understanding of the underlying factors that produced the observed data; this helps to fit a model and thereby forecast, monitor or even feedback and feedforward control.

Some of the applications of Time series include: Economic Forecasting, Sales Forecasting, Budgetary Analysis, Stock Market Analysis, Process and Quality Control etc.

## Business problem 2 – Time Series Forecasting- Rose dataset

## Problem Statement:

**For this particular assignment, the data of different types of wine sales in the 20th century is to be analysed. Both of these data are from the same company but of different wines. As an analyst in the ABC Estate Wines, you are tasked to analyse and forecast Wine Sales in the 20th century.**

## Solution Approach:

The purpose of the solutioning exercise is to explore the dataset using time series techniques to arrive at the customer segmentation, thus enabling business strategies customized to them.

### 1.1. Read the data as an appropriate Time Series data and plot the data

#### Dataset Background:

- Monthly Sales Data of 'Rose' Wine manufactured by ABC Estate Wines starting from Jan 1980 to July 1995 is provided.
- As an analyst in the ABC Estate Wines, the task is to analyse and forecast Wine Sales in the 20th century.

#### Data Dictionary of the Dataset:

- The dataset 'Rose' contains two columns of data:
- The monthly time stamp from Jan 1980 to July 1995 and the sales corresponding to the wines.

Loaded required packages and read Monthly Sales of Rose wine dataset without using panda's date-time format. Head of the data is as below:

| | YearMonth | Rose |
|---|---|---|
| 0 | 1980-01 | 112.0 |
| 1 | 1980-02 | 118.0 |
| 2 | 1980-03 | 129.0 |
| 3 | 1980-04 | 99.0 |
| 4 | 1980-05 | 116.0 |

### Method-1:

Created Time Stamps and adding it to the data frame to make it a Time Series Data, as below:

```
DatetimeIndex(['1980-01-31', '1980-02-29', '1980-03-31', '1980-04-30',
               '1980-05-31', '1980-06-30', '1980-07-31', '1980-08-31',
               '1980-09-30', '1980-10-31',
               ...
               '1994-10-31', '1994-11-30', '1994-12-31', '1995-01-31',
               '1995-02-28', '1995-03-31', '1995-04-30', '1995-05-31',
               '1995-06-30', '1995-07-31'],
              dtype='datetime64[ns]', length=187, freq='M')
```

Add the time stamp to the original data-frame and set the time stamp as an index, also drop the Year Month column from the dataset.

| | Rose |
|---|---|
| Time_Stamp | |
| 1980-01-31 | 112.0 |
| 1980-02-29 | 118.0 |
| 1980-03-31 | 129.0 |
| 1980-04-30 | 99.0 |
| 1980-05-31 | 116.0 |

Figure 1: Head of the Time Series data

### Method – 2:

Alternate way to read the original data-frame has a Time series data is by using panda's functions.

**Squeeze = True will return the index col as series.**

**View the top 5 rows of Rose dataset :**

```
YearMonth
1980-01-01    112.0
1980-02-01    118.0
1980-03-01    129.0
1980-04-01     99.0
1980-05-01    116.0
Name: Rose, dtype: float64
```

**View the bottom 5 rows of Rose dataset :**

```
YearMonth
1995-03-01    45.0
1995-04-01    52.0
1995-05-01    28.0
1995-06-01    40.0
1995-07-01    62.0
Name: Rose, dtype: float64
```

Figure 2: Head and Tail of Time- Series Data

- **Inferences:**
  - **All values are properly loaded for the dataset with the index as pandas datetime format.**
  - **The 'Rose' Time series has values in float64 datatype format.**

## Checking for Null values:

As it is a Time Series data, handling Null values is of utmost importance. The null values cannot be dropped as the Time series data need to contiguous, so they need to be properly imputed.

Rose time series contain 2 missing values, they are for the time stamp '1994-07-01' and '1994-08-01'.

## Plotting the Rose Time Series to understand the behavior of the data:

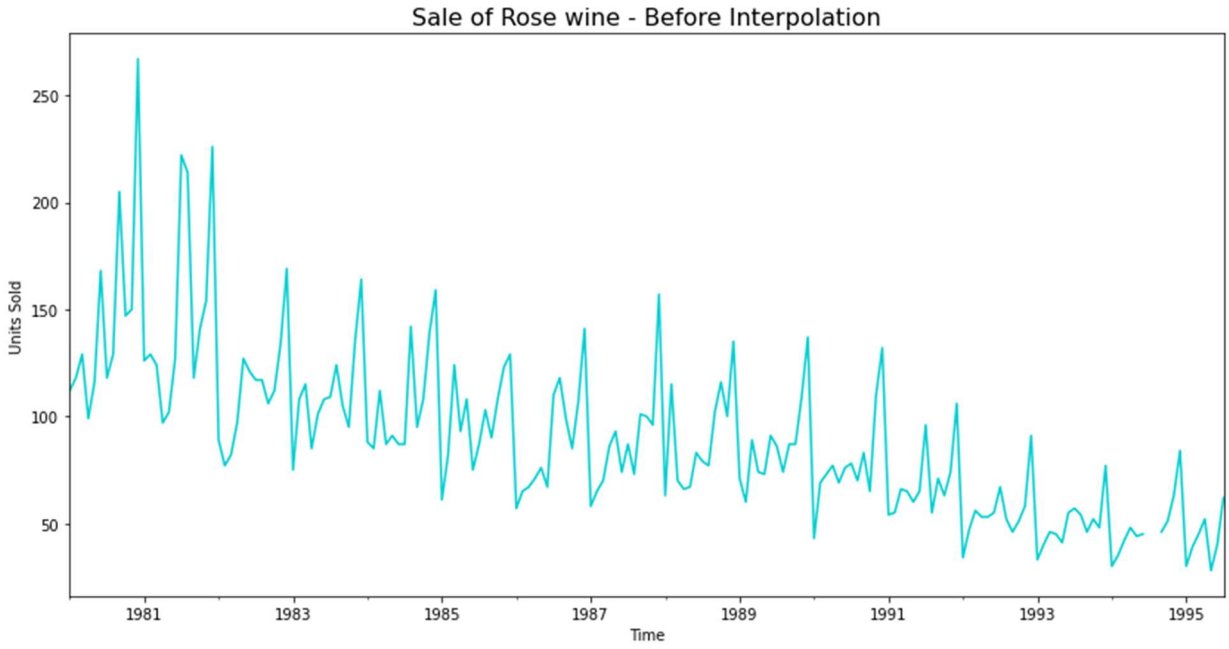



Figure 3: Plot of Rose Time series data – Before Interpolation

➢ **Inferences:**

- **A decreasing Trend could be observed with a multiplicative seasonality present.**
- **The Null values could be observed as a break in the plot for the observed timestamps.**

## Imputing the Null Values:

- We will Impute the null values by using interpolation [polynomial of order 2].

**The new interpolated values of the previously missing values:**

```
YearMonth
1994-07-01    45.364189
1994-08-01    44.279246
Name: Rose, dtype: float64
```

### Interpolated Values

The missing values are imputed using polynomial interpolation of order 2. The new values for the index '1994-07-01' are 45.36 and for index '1994-08-01' is 44.28 approximately. (Although the sales numbers should be whole numbers but here, we are getting float values due to interpolation. The values are kept as it is and not rounded off ) .

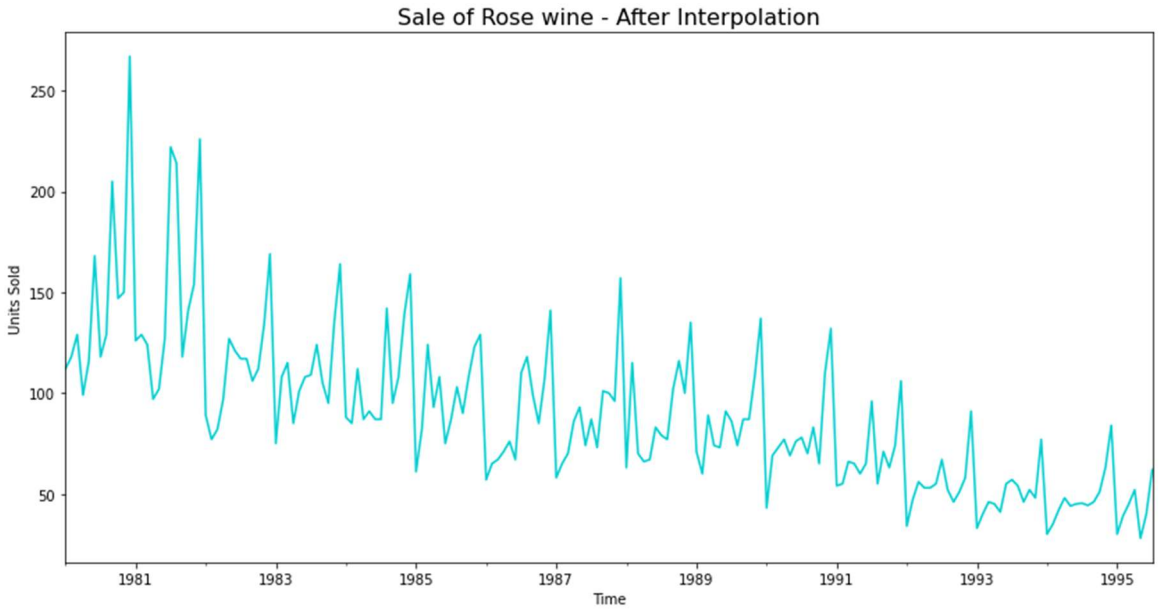



Figure 4: Plot of Rose Time series data – After Interpolation

- The plot now could be observed with no missing values.

## 1.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

**Solution:**

**Checking the basic measures of descriptive statistics:**

**Data Description for Rose Dataset is as below:**

```
count    187.000000
mean      89.907184
std       39.246679
min       28.000000
25%       62.500000
50%       85.000000
75%      111.000000
max      267.000000
Name: Rose, dtype: float64
```

- **Inferences:**
  - **The mean value of the Time Series is nearly same as the median values. As a time, series data, it may signify presence of decreasing trend and multiplicative seasonality.**
- **The descriptive summary of the data shows that on an average 90 units of Rose wines were sold each month on the given period of time. 50% of months sales varied from 63 units to 112 units. Maximum sale reported in a month is 267 units and minimum of 28 units.**
- **The basic measures of descriptive statistics tell us how the Sales have varied across years. But for this measure of descriptive statistics, we have averaged over the whole data without taking the time component into account.**

We will Plot a boxplot to understand the spread of sales across different years and within different months across years.

**Yearly Boxplot – Rose, shows that there is a yearly decreasing Trend.**

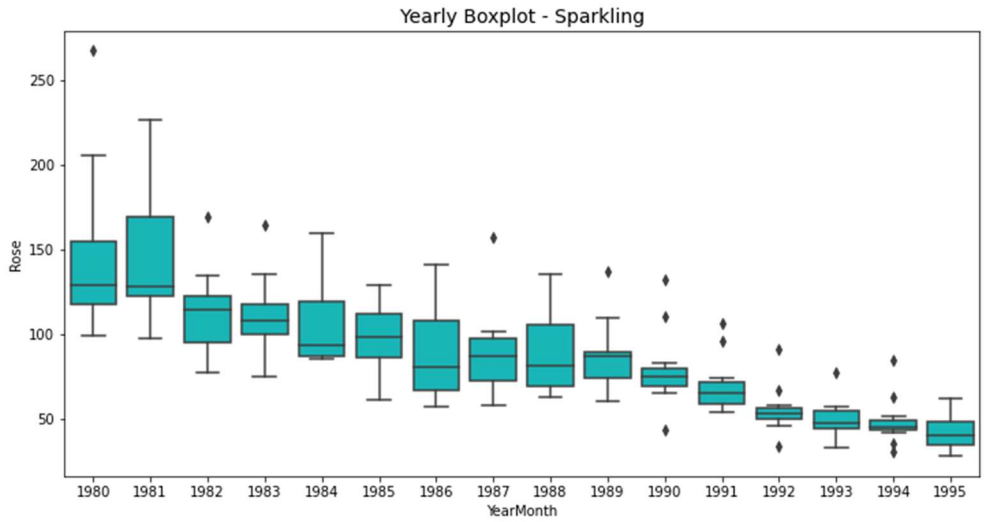



Figure 5: Yearly Boxplot for Rose Dataset

## Monthly Plot – Rose:

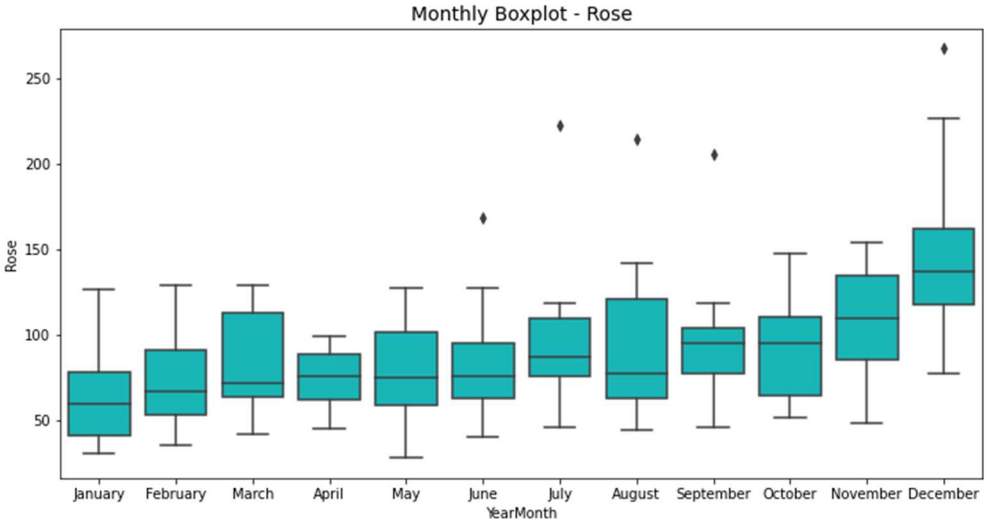
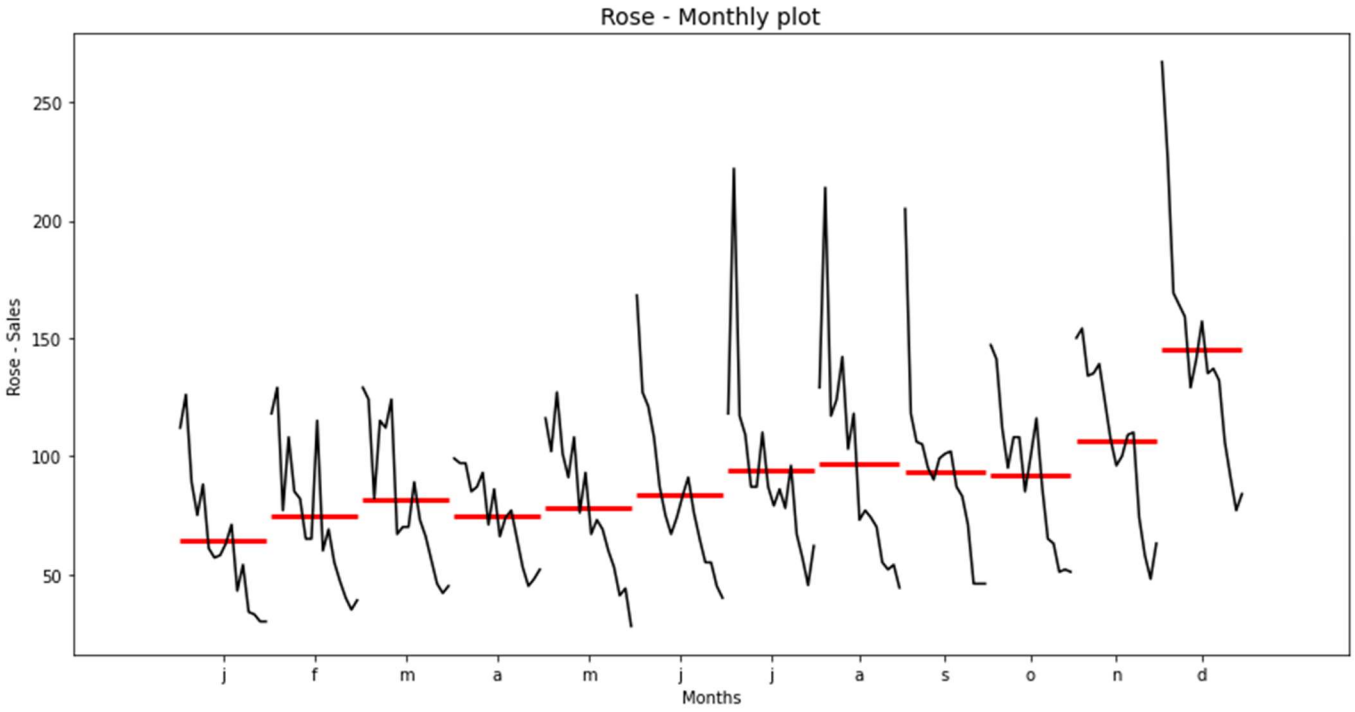



Figure 6: Monthly Boxplot for all the years for Rose Dataset

- **Inferences:**

- **The yearly-boxplot, shows that the average sale of Rose wine moving according to the downward trend in sales over the years. The outliers over upper bound in the yearly-boxplot most probably represent the seasonal sale during the seasonal months.**
- **The monthly-box-plot shows a clear seasonality during the seasonal months of November and December. Though the sale tanks in the month of January, it picks up in the due course of the year.**
- **Average sale in December is around 140 units, November is around 110 units and October is around 90 units.**

We will Plot a time series month plot to understand the spread of sales across different years and within different months across years.




**Monthly Plot**

This plot shows us the behavior of the Time Series across various months. The red line is the median value.

- **Inferences:**

- The monthly plot for Rose shows mean and variation of units sold each month over the years. Sale in months such as July, August, September and December show a higher variation than the rest.
- Sale in December with a mean- few point below 100, varies from 75 to 270 units over the years. Whereas the average sale is less than or closer to 100 units (above50) for the rest of the year.

**Plot graph of monthly Wine sales across years:**

| YearMonth | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| YearMonth | | | | | | | | | | | | |
| 1980 | 112.0 | 118.0 | 129.0 | 99.0 | 116.0 | 168.0 | 118.000000 | 129.000000 | 205.0 | 147.0 | 150.0 | 267.0 |
| 1981 | 126.0 | 129.0 | 124.0 | 97.0 | 102.0 | 127.0 | 222.000000 | 214.000000 | 118.0 | 141.0 | 154.0 | 226.0 |
| 1982 | 89.0 | 77.0 | 82.0 | 97.0 | 127.0 | 121.0 | 117.000000 | 117.000000 | 106.0 | 112.0 | 134.0 | 169.0 |
| 1983 | 75.0 | 108.0 | 115.0 | 85.0 | 101.0 | 108.0 | 109.000000 | 124.000000 | 105.0 | 95.0 | 135.0 | 164.0 |
| 1984 | 88.0 | 85.0 | 112.0 | 87.0 | 91.0 | 87.0 | 87.000000 | 142.000000 | 95.0 | 108.0 | 139.0 | 159.0 |
| 1985 | 61.0 | 82.0 | 124.0 | 93.0 | 108.0 | 75.0 | 87.000000 | 103.000000 | 90.0 | 108.0 | 123.0 | 129.0 |
| 1986 | 57.0 | 65.0 | 67.0 | 71.0 | 76.0 | 67.0 | 110.000000 | 118.000000 | 99.0 | 85.0 | 107.0 | 141.0 |
| 1987 | 58.0 | 65.0 | 70.0 | 86.0 | 93.0 | 74.0 | 87.000000 | 73.000000 | 101.0 | 100.0 | 96.0 | 157.0 |
| 1988 | 63.0 | 115.0 | 70.0 | 66.0 | 67.0 | 83.0 | 79.000000 | 77.000000 | 102.0 | 116.0 | 100.0 | 135.0 |
| 1989 | 71.0 | 60.0 | 89.0 | 74.0 | 73.0 | 91.0 | 86.000000 | 74.000000 | 87.0 | 87.0 | 109.0 | 137.0 |
| 1990 | 43.0 | 69.0 | 73.0 | 77.0 | 69.0 | 76.0 | 78.000000 | 70.000000 | 83.0 | 65.0 | 110.0 | 132.0 |
| 1991 | 54.0 | 55.0 | 66.0 | 65.0 | 60.0 | 65.0 | 96.000000 | 55.000000 | 71.0 | 63.0 | 74.0 | 106.0 |
| 1992 | 34.0 | 47.0 | 56.0 | 53.0 | 53.0 | 55.0 | 67.000000 | 52.000000 | 46.0 | 51.0 | 58.0 | 91.0 |
| 1993 | 33.0 | 40.0 | 46.0 | 45.0 | 41.0 | 55.0 | 57.000000 | 54.000000 | 46.0 | 52.0 | 48.0 | 77.0 |
| 1994 | 30.0 | 35.0 | 42.0 | 48.0 | 44.0 | 45.0 | 45.364189 | 44.279246 | 46.0 | 51.0 | 63.0 | 84.0 |
| 1995 | 30.0 | 39.0 | 45.0 | 52.0 | 28.0 | 40.0 | 62.000000 | NaN | NaN | NaN | NaN | NaN |

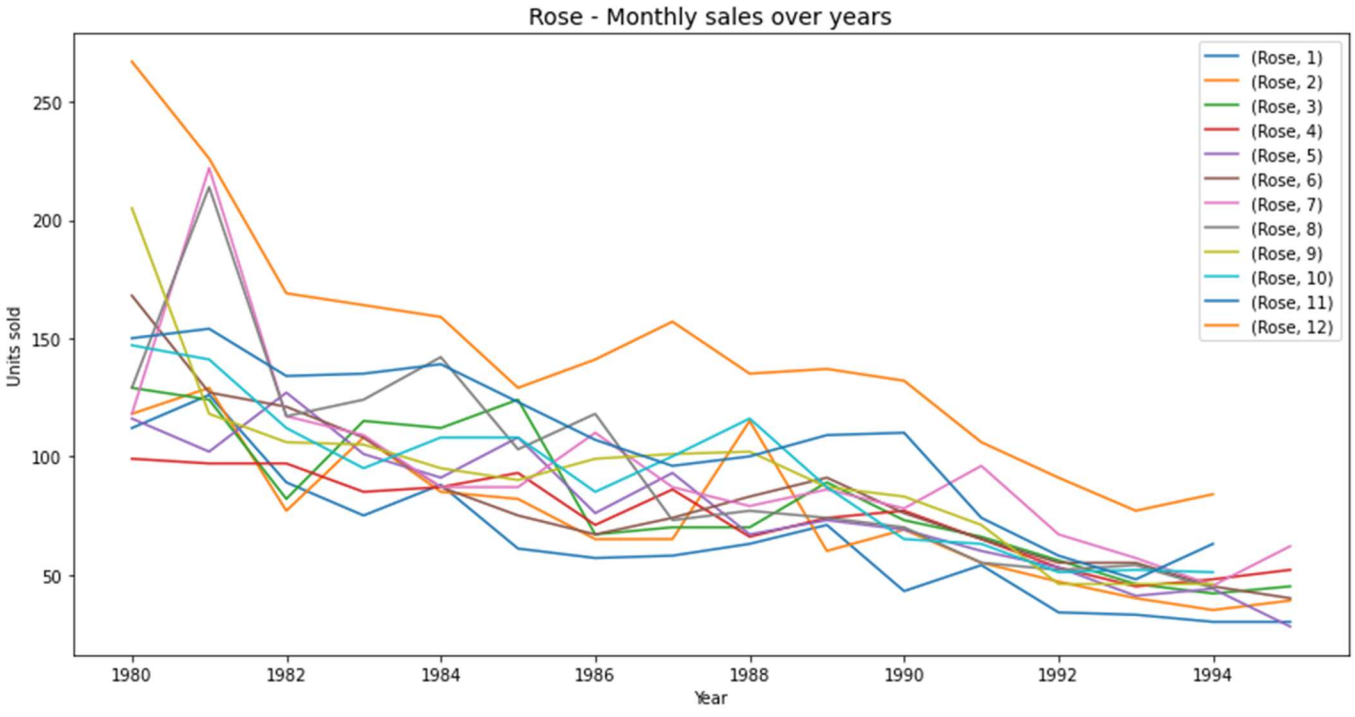



Figure 7: Monthly Wine sales across years for Rose

- **Inferences:**

- A decreasing Trend could be observed for the different months along the Years.
- Certain months have comparatively higher values throughout the years.
- The plot of monthly sale over the years also shows the seasonality component of the time-series, with November and December selling exponentially higher volumes than other months.
- The highest volume of Rose wines was sold in December, 1980 and the least of December sale was in 1993. Though December sale picked after 1983, it consistently dipped after 1987.

We will group the data by date and get average Rose sales and precent change.

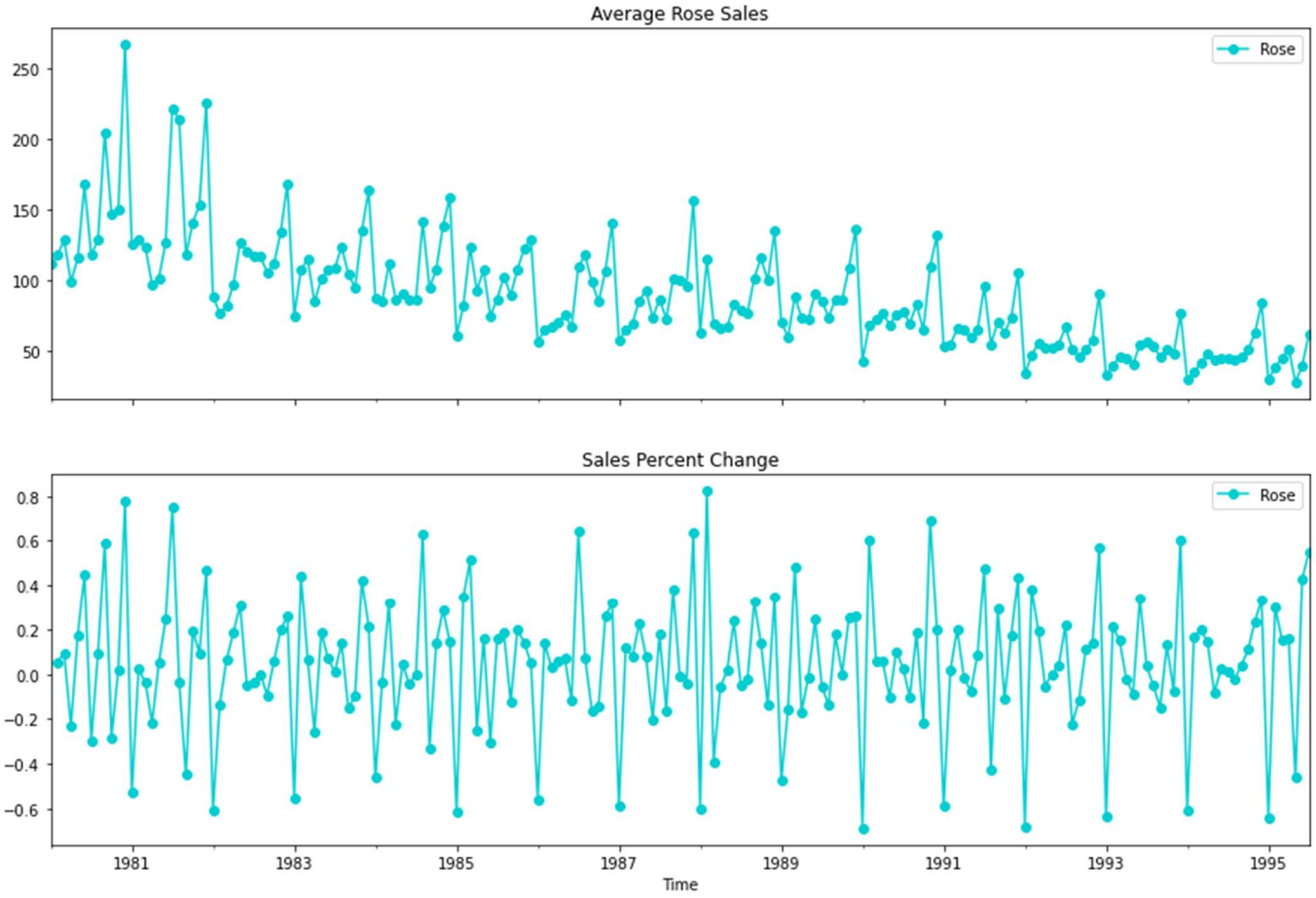



Figure 8: Average Sales and Percent change of Rose dataset

- There certain higher percentage changes between months present periodically suggesting presence of Seasonality.

## Decomposition of the Time Series and plot the different components:

By observing the plot above we have seen presence of multiplicative seasonality and thus decompose the rose time series accordingly.

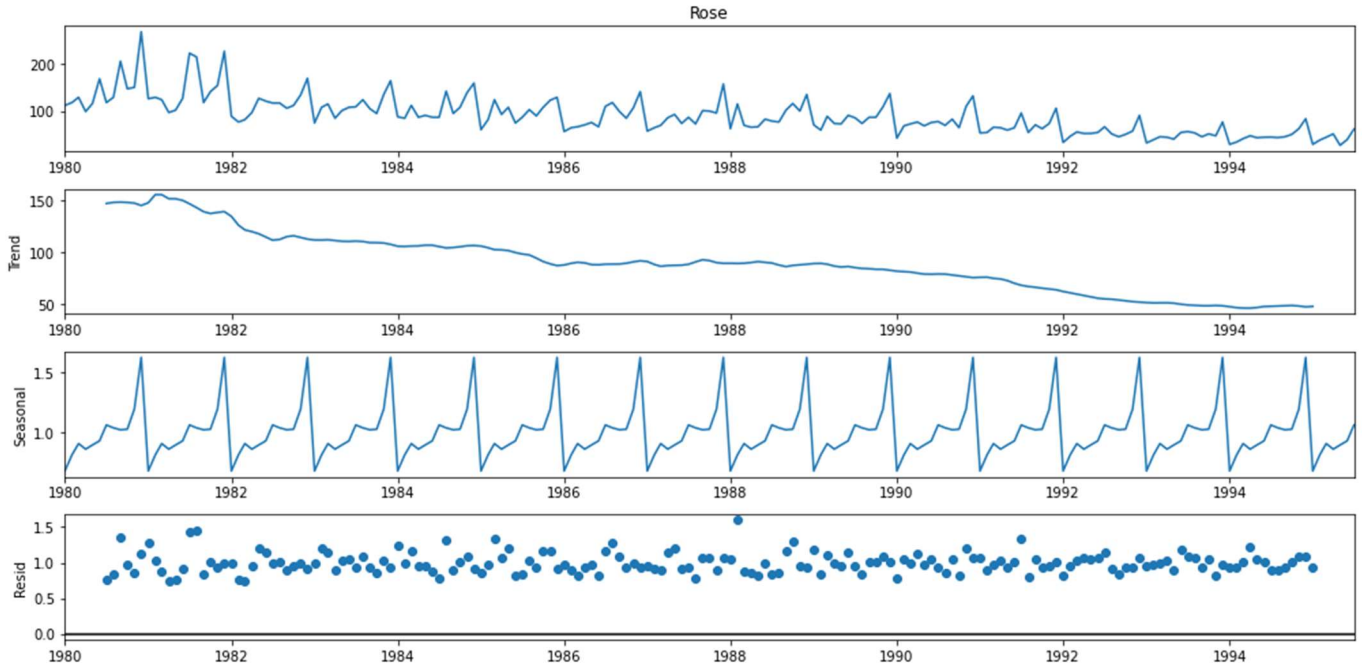



Figure 9: Decomposition of Rose Time Series with multiplicative Seasonality

- **Inferences:**
- **The observed plot of the decomposition diagram shows visible annual seasonality and a downward trend. The early period of the plot shows higher variation than in the later periods.**
- **The trend diagram shows a downward trend overall. Exponential dips can be seen between 1981 and 1983 and later from 1991 to 1993.**
- **Seasonal components are quite visible and consistent in both the observed and seasonal charts of the diagrams. The multiplicative model shows variance in seasonality of 16% .**
- **The residuals show a pattern of high variability across the period of time-series, which is more or less consistent.**
- **The variance in residuals shows higher variance in the early period of the series, which explains the higher variance in observed plot at same time period.**
- **As the seasonality peaks are consistently reducing its altitude in consistent with trend, the series can be treated as multiplicative in model building.**

## 1.3 Split the data into training and test. The test data should start in 1991.

**Solution:** The train and test datasets are created with year 1991 as starting year for test data.

**Length of Train Data: 132**
**Length of Test Data: 55**

First few rows of Training Data:

| YearMonth | Rose |
|---|---|
| 1980-01-01 | 112.0 |
| 1980-02-01 | 118.0 |
| 1980-03-01 | 129.0 |
| 1980-04-01 | 99.0 |
| 1980-05-01 | 116.0 |

Last few rows of Training Data:

| YearMonth | Rose |
|---|---|
| 1990-08-01 | 70.0 |
| 1990-09-01 | 83.0 |
| 1990-10-01 | 65.0 |
| 1990-11-01 | 110.0 |
| 1990-12-01 | 132.0 |

First few rows of Test Data:

| YearMonth | Rose |
|---|---|
| 1991-01-01 | 54.0 |
| 1991-02-01 | 55.0 |
| 1991-03-01 | 66.0 |
| 1991-04-01 | 65.0 |
| 1991-05-01 | 60.0 |

Last few rows of Test Data:

| YearMonth | Rose |
|---|---|
| 1995-03-01 | 45.0 |
| 1995-04-01 | 52.0 |
| 1995-05-01 | 28.0 |
| 1995-06-01 | 40.0 |
| 1995-07-01 | 62.0 |

**Train and Test Data**

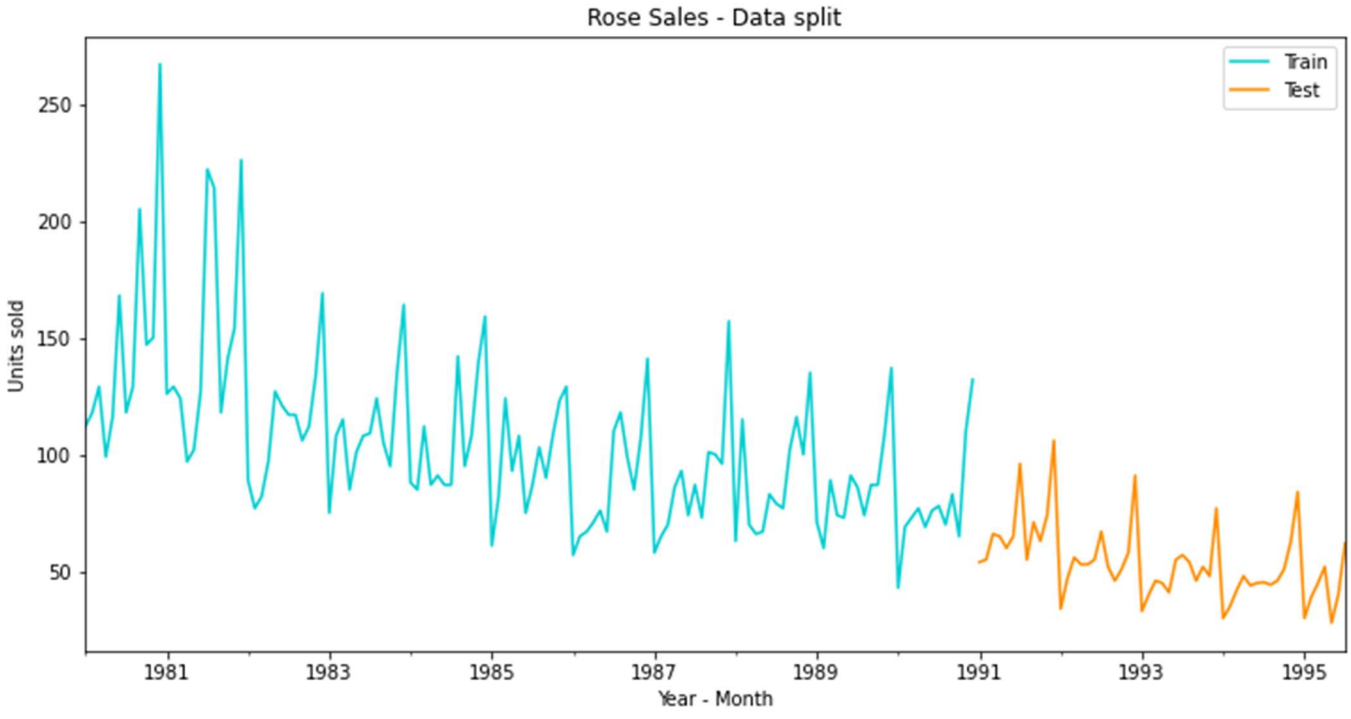



Figure 10: The Plot of Rose Time Series as train and test

- The train test split is done with the test data staring from the year 1991. There 132 values in the Train set and 55 values in the test set. The starting and ending values are also observed for the train and test set. The plot showing the train and test together is also observed.

## 1.4 Build various exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other models such as regression, naïve forecast models, simple average models etc. should also be built on the training data and check the performance on the test data using RMSE.

## Solution:

## Model 1: Linear Regression

To observe the sale of Rose wines, numerical time instance order for both training and test set were generated and the values added to the respective datasets.

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 12
1, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 15
7, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 1
82, 183, 184, 185, 186, 187]
```

We see that we have successfully the generated the numerical time instance order for both the training and test set. Now we will add these values in the training and test set.

First few rows of Training Data

| YearMonth | Rose | time |
|---|---|---|
| 1980-01-01 | 112.0 | 1 |
| 1980-02-01 | 118.0 | 2 |
| 1980-03-01 | 129.0 | 3 |
| 1980-04-01 | 99.0 | 4 |
| 1980-05-01 | 116.0 | 5 |

Last few rows of Training Data

| YearMonth | Rose | time |
|---|---|---|
| 1990-08-01 | 70.0 | 128 |
| 1990-09-01 | 83.0 | 129 |
| 1990-10-01 | 65.0 | 130 |
| 1990-11-01 | 110.0 | 131 |
| 1990-12-01 | 132.0 | 132 |

First few rows of Test Data

| YearMonth | Rose | time |
|---|---|---|
| 1991-01-01 | 54.0 | 133 |
| 1991-02-01 | 55.0 | 134 |
| 1991-03-01 | 66.0 | 135 |
| 1991-04-01 | 65.0 | 136 |
| 1991-05-01 | 60.0 | 137 |

Last few rows of Test Data

| YearMonth | Rose | time |
|---|---|---|
| 1995-03-01 | 45.0 | 183 |
| 1995-04-01 | 52.0 | 184 |
| 1995-05-01 | 28.0 | 185 |
| 1995-06-01 | 40.0 | 186 |
| 1995-07-01 | 62.0 | 187 |

Now that our training and test data has been modified, let us go ahead use LINEAR REGRESSION to build the model on the training data and test the model on the test data.

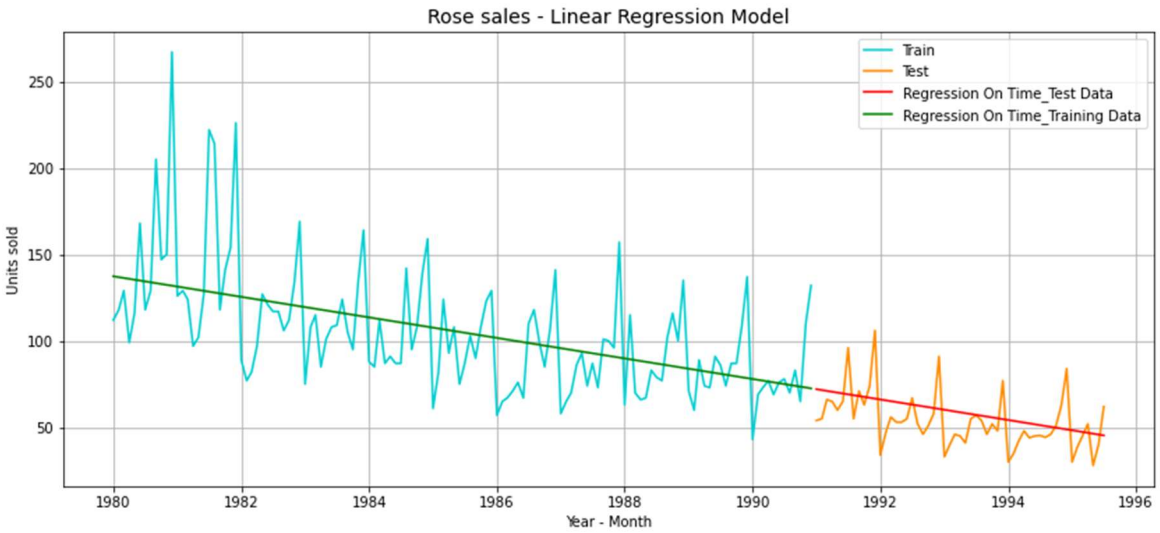



Figure 11: Linear Regression Model

### Inferences:

- The linear regression on the Rose dataset shows an apparent downward trend as consistent with the observed time-series.
- For Regression on Time forecast on the Test Data, RMSE is 15.278
- The model has successfully captured the trend of the series, but does not reflect the seasonality.

## Model 2: Naive forecast

In naive model, the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.

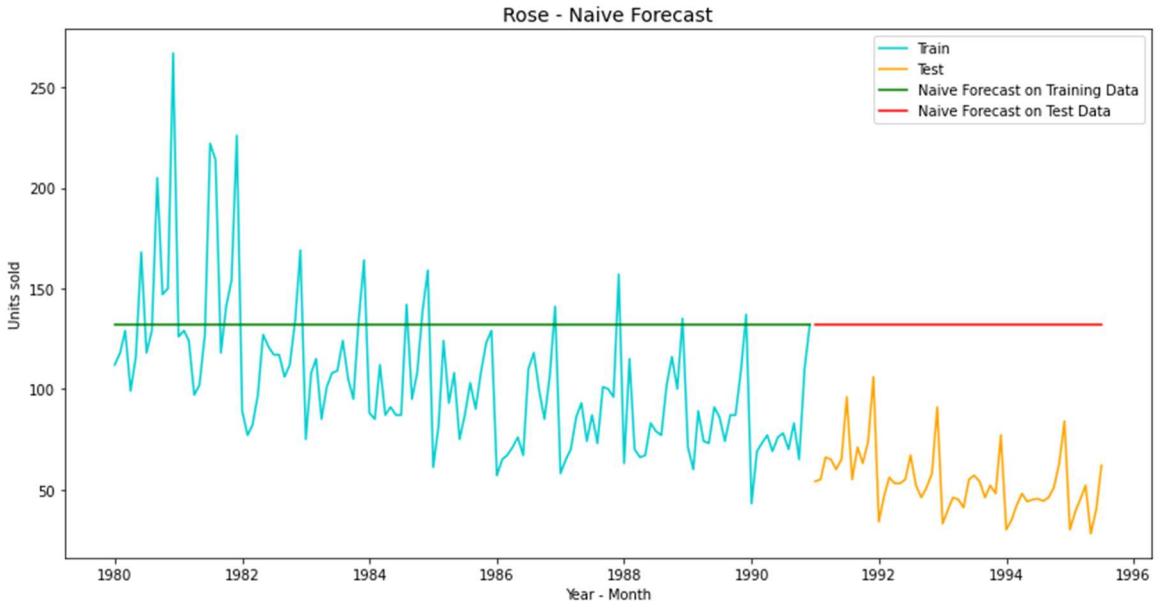



Figure 12: Naïve Forecast Model

- The model has taken the last value from the test set and fitted it on the rest of the train time period and used the same value to forecast the test set.
- For Naive forecast on the Test Data, RMSE is 79.75.
- The model does not capture the trend or seasonality for the given dataset.

## Model 3: Simple Average

In the Simple Average model, the forecast is done using the mean of the time-series variable from the training set.

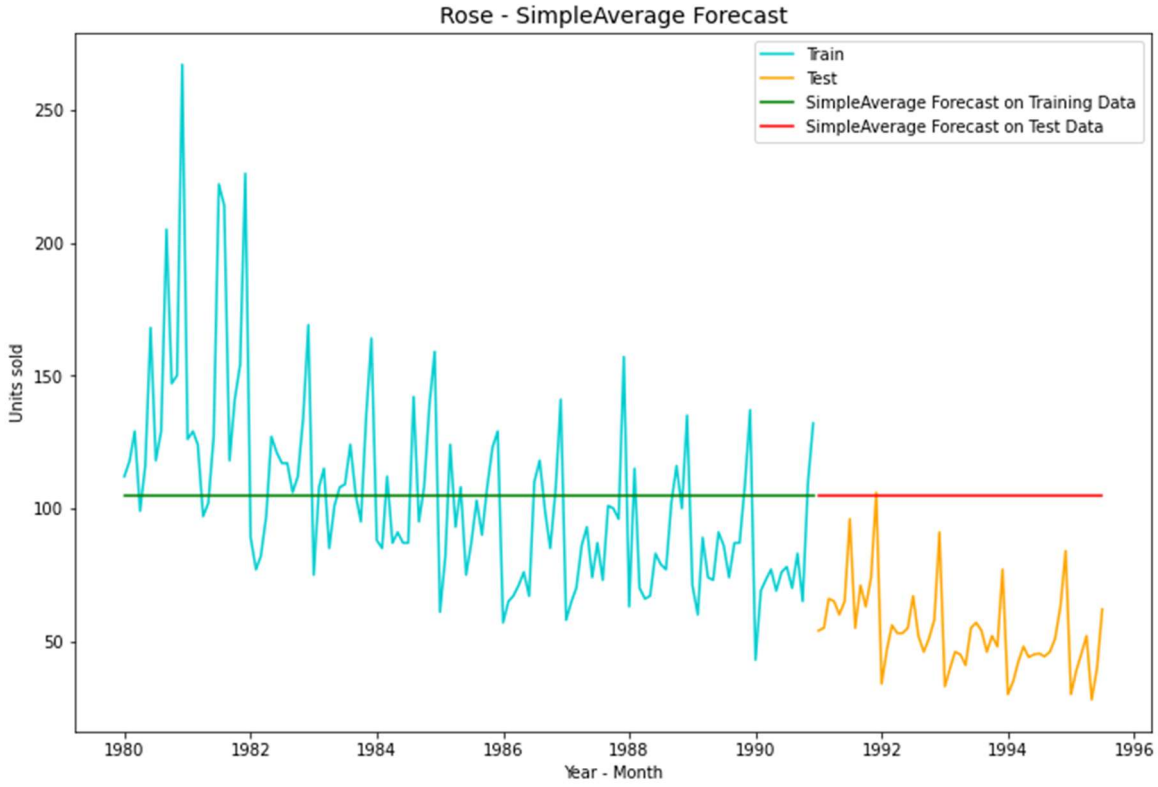



Figure 13: Simple Average Model

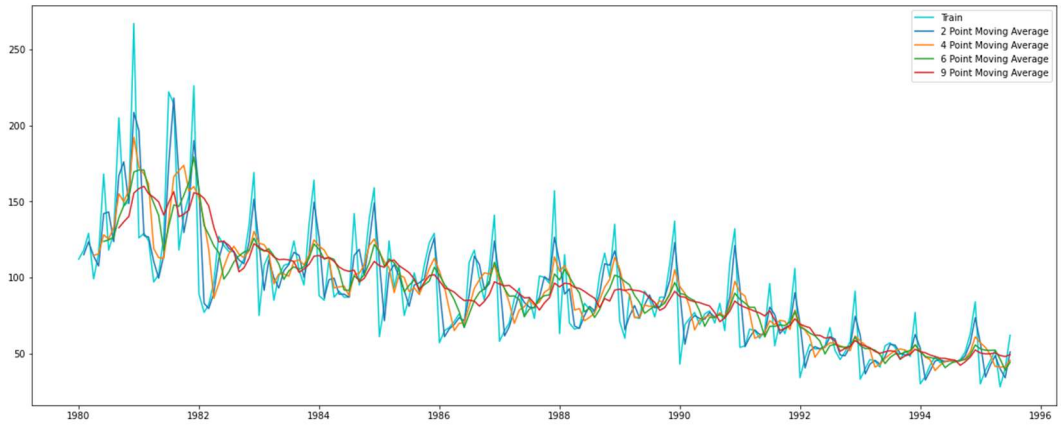
**No trend and seasonality present in the dataset. For Simple Average Model on the Test Data, RMSE value is 53.48**

## Model 4: Moving Average

- For the moving average model, we will calculate rolling means (or trailing moving averages) for different intervals. The best interval can be determined by the maximum accuracy.
- The moving average models are built for trailing 2 points, 4 points, 6 points and 9 points.
- For Rose dataset the accuracy is found to be higher with the lower rolling point averages.
- In moving average forecasts the values can be fitted with a delay of n number of points.
- The best interval of moving average from the model is 2 points.

| YearMonth | Rose | Rose_Trailing_2 | Rose_Trailing_4 | Rose_Trailing_6 | Rose_Trailing_9 |
|---|---|---|---|---|---|
| 1980-01-01 | 112.0 | NaN | NaN | NaN | NaN |
| 1980-02-01 | 118.0 | 115.0 | NaN | NaN | NaN |
| 1980-03-01 | 129.0 | 123.5 | NaN | NaN | NaN |
| 1980-04-01 | 99.0 | 114.0 | 114.5 | NaN | NaN |
| 1980-05-01 | 116.0 | 107.5 | 115.5 | NaN | NaN |

## Plotting on Training data:




## Plotting on Test data:

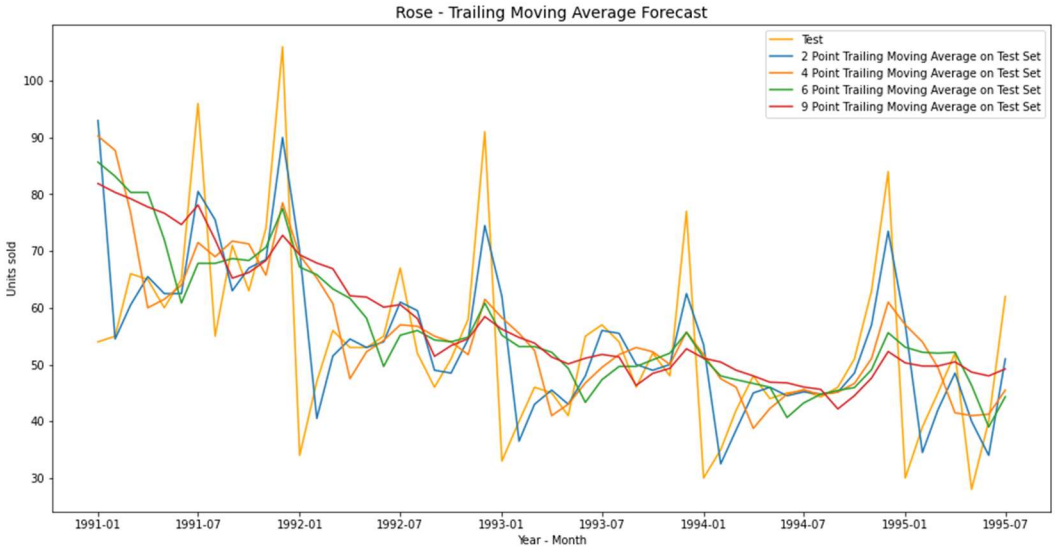



Figure 14: Moving Average on Train and Test data

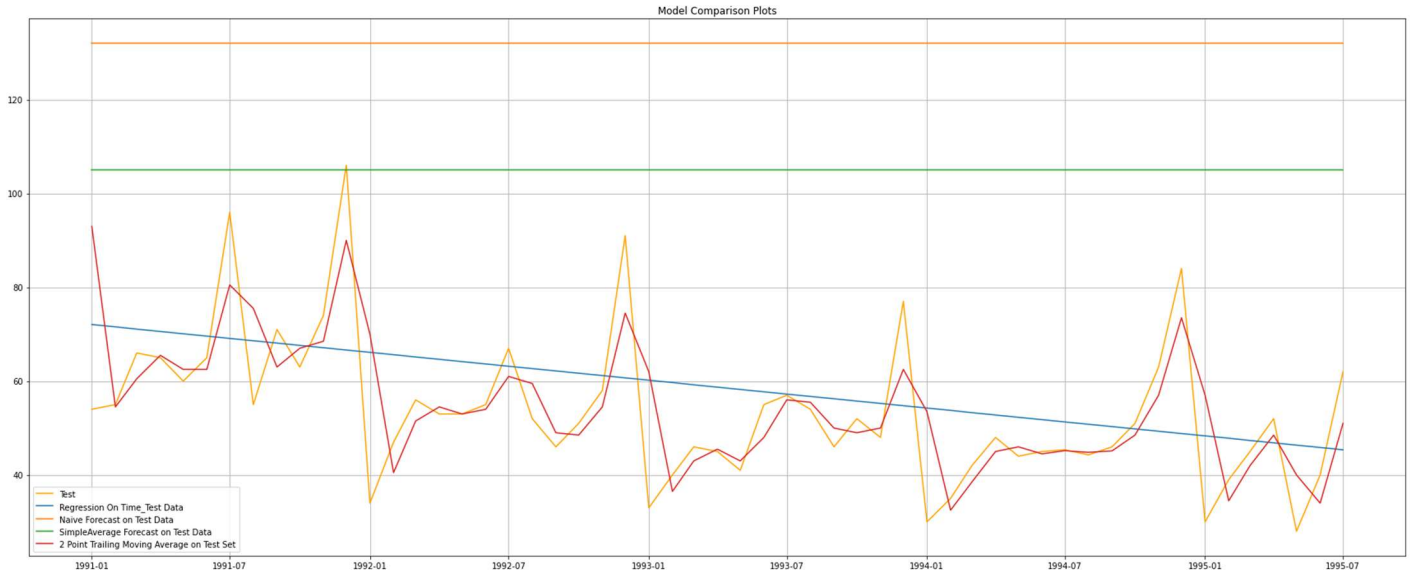
For 2 point Moving Average Model forecast on the Test Data, rmse_rose is 11.53
For 4 point Moving Average Model forecast on the Test Data, rmse_rose is 14.46
For 6 point Moving Average Model forecast on the Test Data, rmse_rose is 14.57
For 9 point Moving Average Model forecast on the Test Data, rmse_rose is 14.73

## Model Comparison and RMSE Value:

### RMSE on Test Data

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |

### Model Comparison



## Model 5: Simple Exponential Smoothing

### Auto fit Model:

The output parameters for optimized model are:

```
{'smoothing_level': 0.0987493111726833,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 134.38720226208358,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The auto-fit model picked up alpha = 0.0987 as the smoothing parameter. Simple Exponential Smoothing is applied if the time-series has neither a trend nor seasonality, which is not the case with the given data.

**Viewing the first five Predictions for Test Data:**

| YearMonth | Rose | predict |
|---|---|---|
| 1991-01-01 | 54.0 | 87.104983 |
| 1991-02-01 | 55.0 | 87.104983 |
| 1991-03-01 | 66.0 | 87.104983 |
| 1991-04-01 | 65.0 | 87.104983 |
| 1991-05-01 | 60.0 | 87.104983 |

### Plotting on both the Training and Test data:

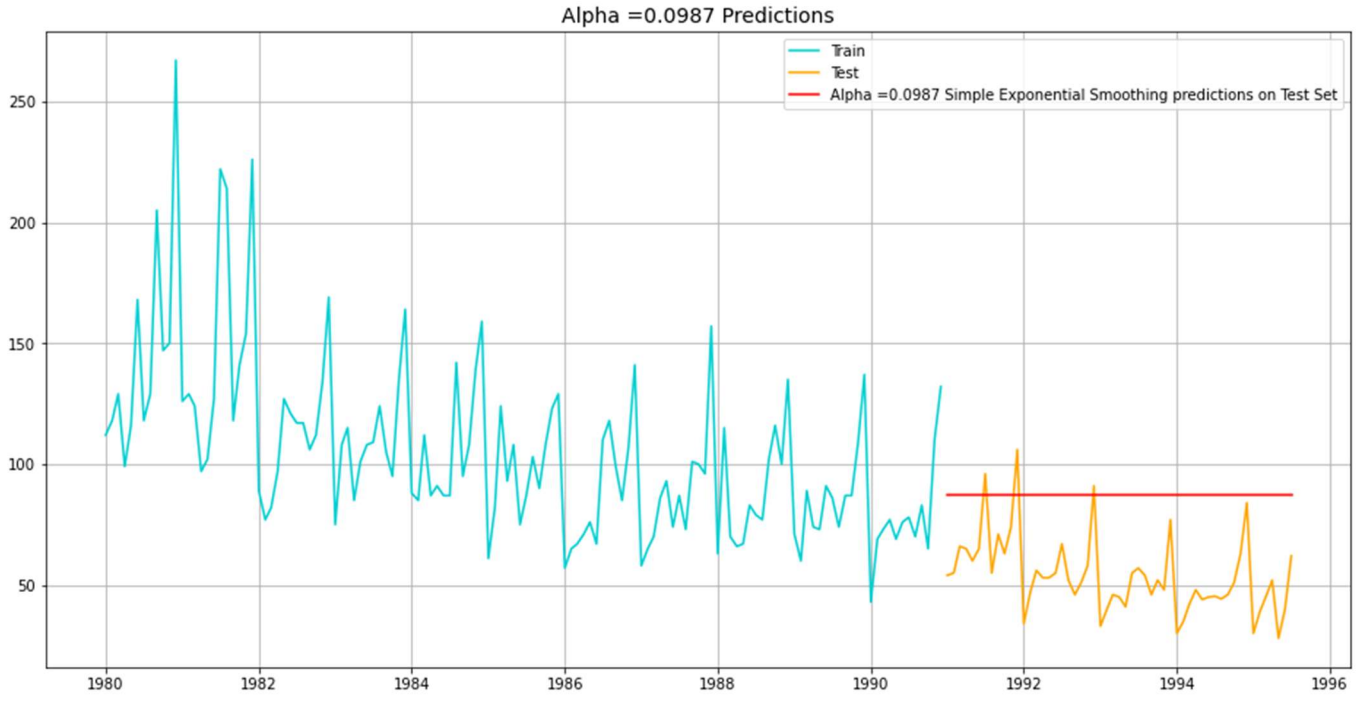



Figure 15: Simple Exponential Smoothing Model

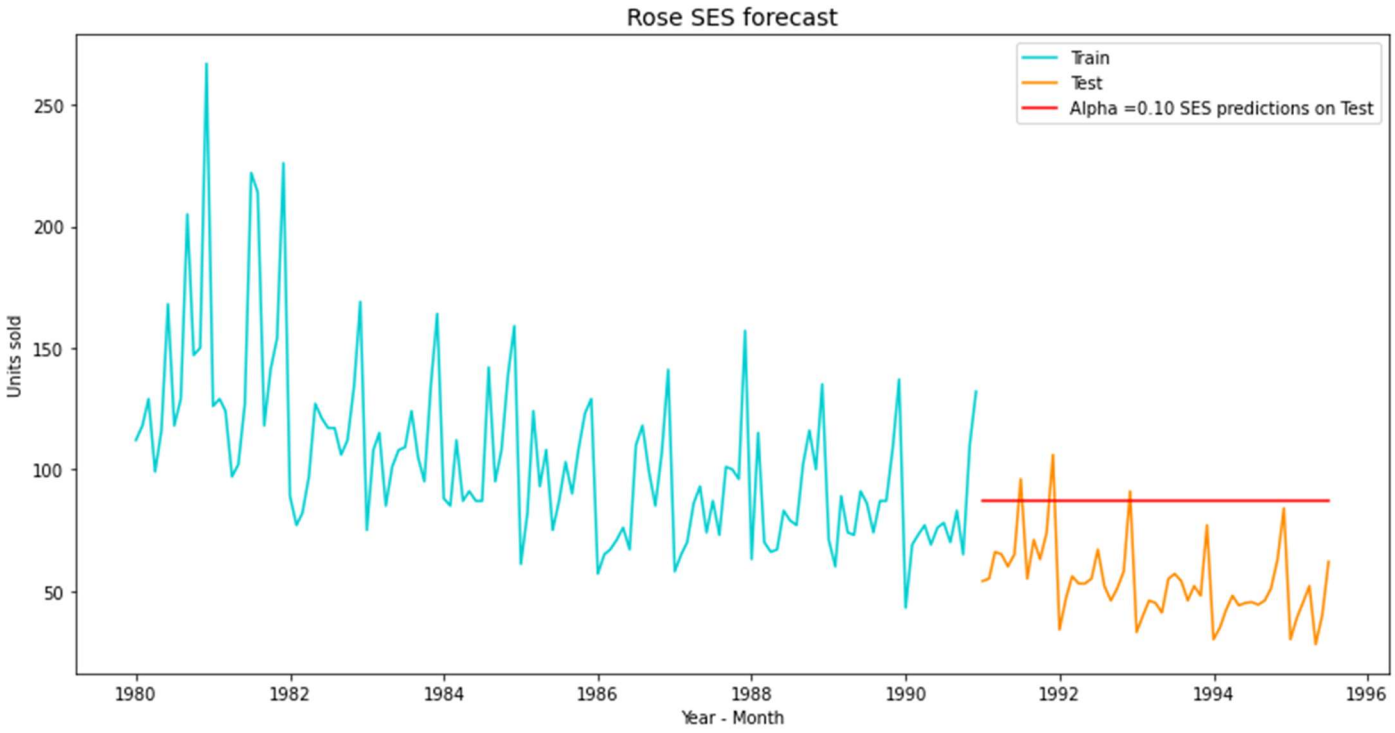
For Alpha =0.0987 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 36.824

## Iterative Method for Simple Exponential Smoothing:

**Model Evaluation based on Iterations:**

| | Alpha Values | Train RMSE | Test RMSE |
|---|---|---|---|
| 2 | 0.10 | 31.815610 | 36.856268 |
| 1 | 0.05 | 32.449102 | 37.039679 |
| 3 | 0.15 | 31.809845 | 38.750307 |
| 4 | 0.20 | 31.979391 | 41.389972 |
| 5 | 0.25 | 32.211871 | 44.388786 |

**Simple Exponential Smoothing Iterative Model is shown as below:**




- For alpha value closer to 1, forecasts follow the actual observation closely and closer to 0, forecasts are farther from actual and line gets smoothened.
- For Rose, test RMSE is found to be higher for values closer to zero, which is same as in Simple average forecast.
- Both manual alpha =0.10 and optimized alpha value are having similar RMSE value.

## Model 6: Double Exponential Smoothing (Holt's Model)

The Double Exponential Smoothing models is applicable when data has trend, but no seasonality. Rose data contain significant trend component and seasonality.

The output parameters for optimized model are:

```
{'smoothing_level': 0.017549790270679714,
 'smoothing_trend': 3.236153800377395e-05,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 138.82081494774005,
 'initial_trend': -0.492580228245491,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

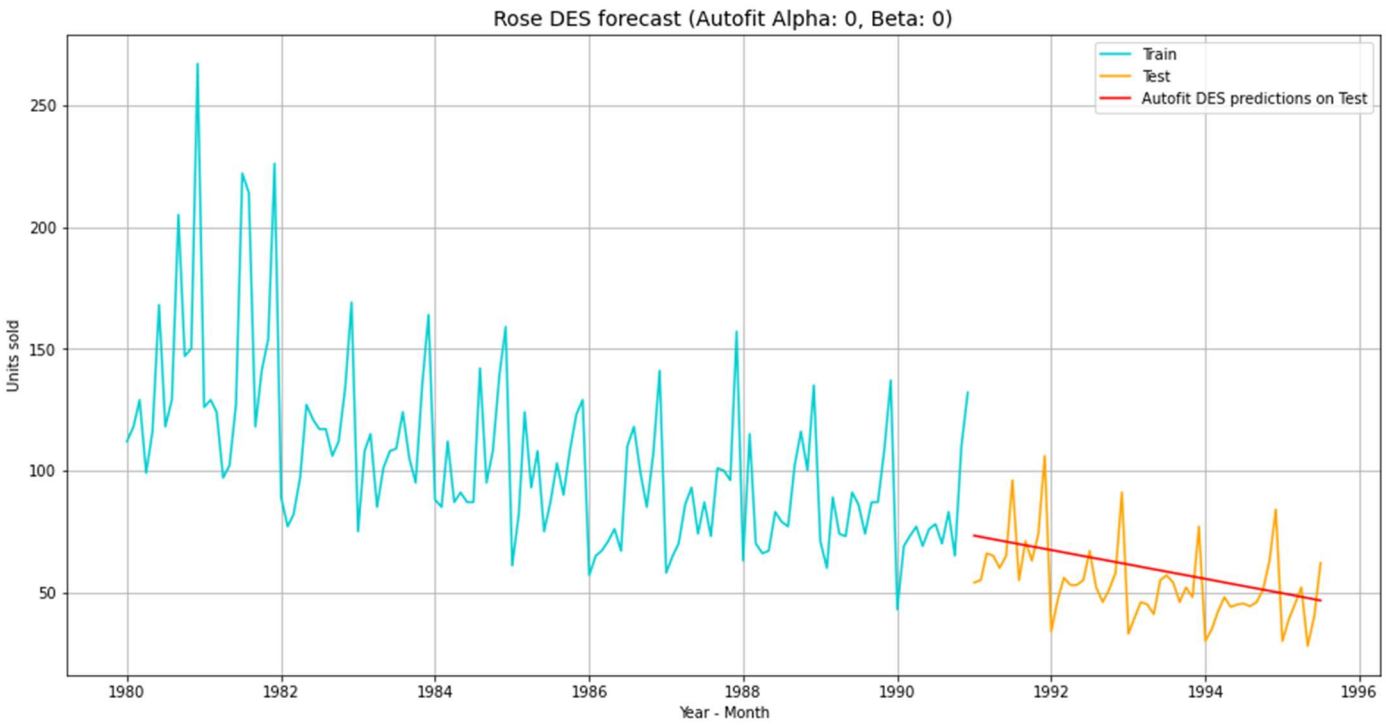
**Viewing the first five Predictions for Test Data:**

| | Rose | (predict, 0.017549790270679714, 3.236153800377395e-05) |
|---|---|---|
| YearMonth | | |
| 1991-01-01 | 54.0 | 73.259732 |
| 1991-02-01 | 55.0 | 72.767150 |
| 1991-03-01 | 66.0 | 72.274569 |
| 1991-04-01 | 65.0 | 71.781987 |
| 1991-05-01 | 60.0 | 71.289405 |

### Inferences:

- In first iteration, smoothing level (alpha) and trend (beta) are fitted to the model iteratively from values 0.1 to 1 and the best combination was chosen based on the RMSE values, which is as below with alpha 0.1 and beta 0.1.
- On the second iteration the model was allowed to choose the optimized values using parameters 'optimized=True, use brute="True".
- The auto-fit model has lower RMSE value compared to iterative alpha=0.1 and beta=0.1 RMSE value.

## Plotting on both the Training and Test data - Double Exponential Smoothing Model




## Iterative Method for Double Exponential Smoothing:

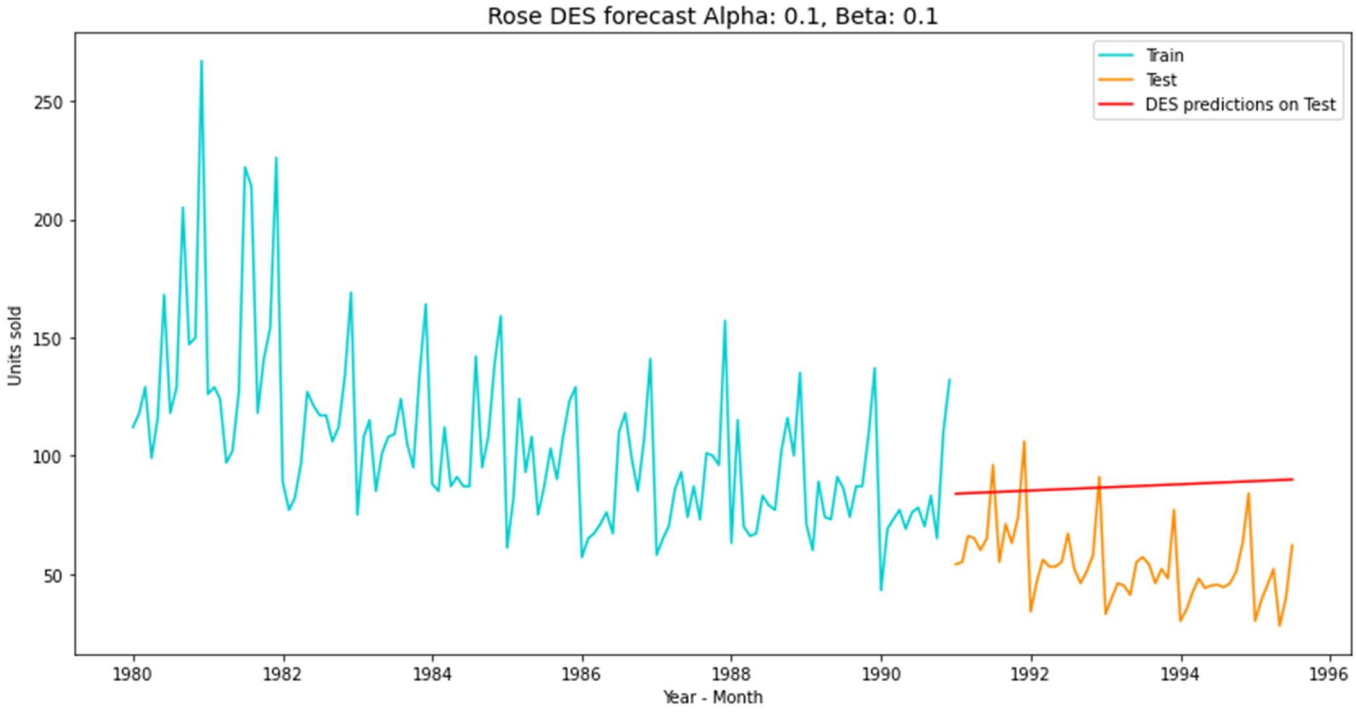



Figure 16: Double Exponential Smoothing Iterative Model

## RMSE VALUES:

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |

## Model 7: Triple Exponential Smoothing (Holt - Winter's Model)

The Triple Exponential Smoothing models (Holt-Winter's Model) is applicable when data has both trend and seasonality. Rose data contain significant trend and seasonality.

The output parameters for optimized model are:

```
{'smoothing_level': 0.06569374607191865,
 'smoothing_trend': 0.05192938504457338,
 'smoothing_seasonal': 3.879136202038614e-06,
 'damping_trend': nan,
 'initial_level': 54.10985491750761,
 'initial_trend': -0.33471965714896845,
 'initial_seasons': array([2.08282313, 2.36326666, 2.58210206, 2.25702695, 2.53757493,
        2.76639991, 3.04101803, 3.23434567, 3.06747277, 3.00164124,
        3.49893806, 4.82552476]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

## Plotting on both the Training and Test using autofit:

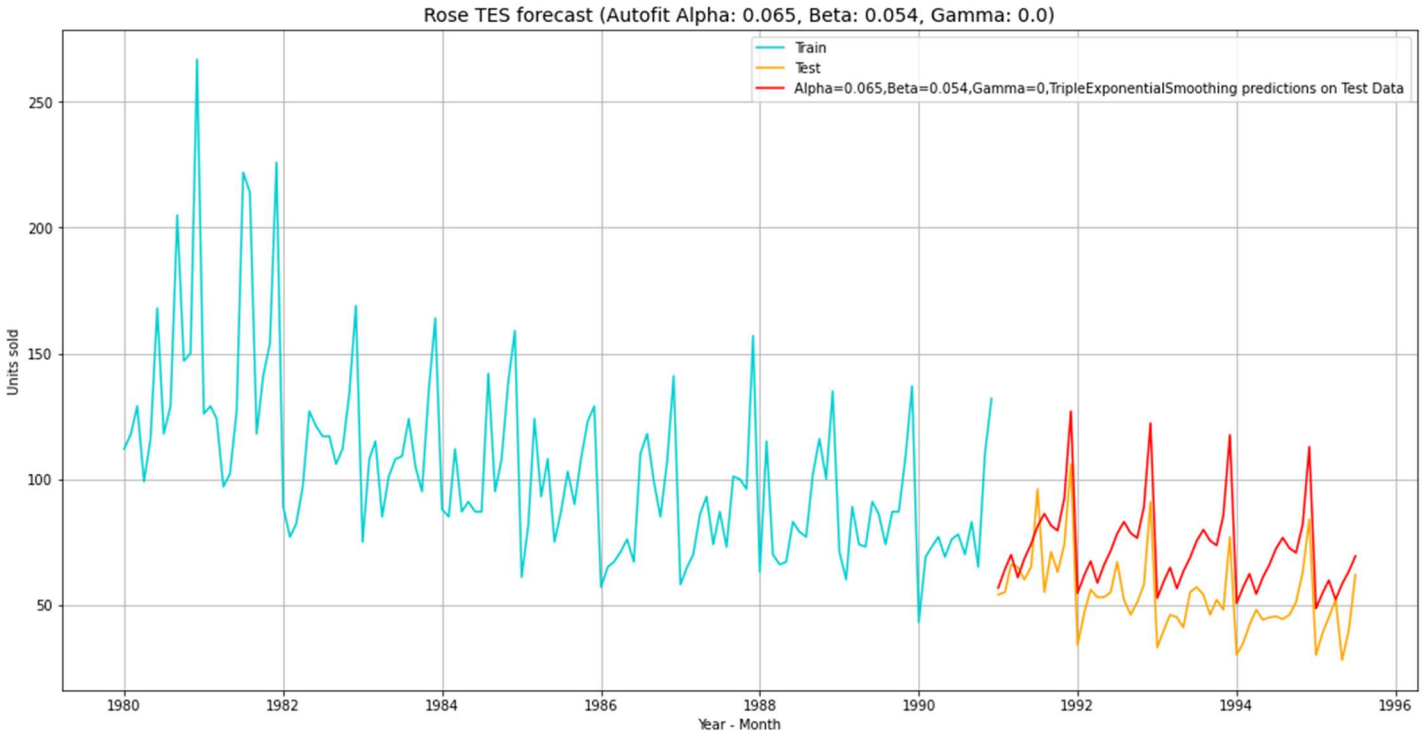



Figure 17: Triple Exponential Smoothing Optimized Model

- On first iteration, smoothing level (alpha), trend (beta) and seasonality (gamma) are fitted to the model iteratively from values 0.1 to 1 and the best combination was chosen based on the RMSE values, which is as below with alpha 0.4, beta 0.1 and gamma 0.3.
- On the second iteration the model was allowed to choose the optimized values using parameters 'optimized=True, use brute="True".
- The auto-fit model retuned higher RMSE value compared to iterative alpha=0.1, beta=0.2 and gamma=0.3 RMSE value.
- For Auto-fit Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 21.057.

## Model Comparison:

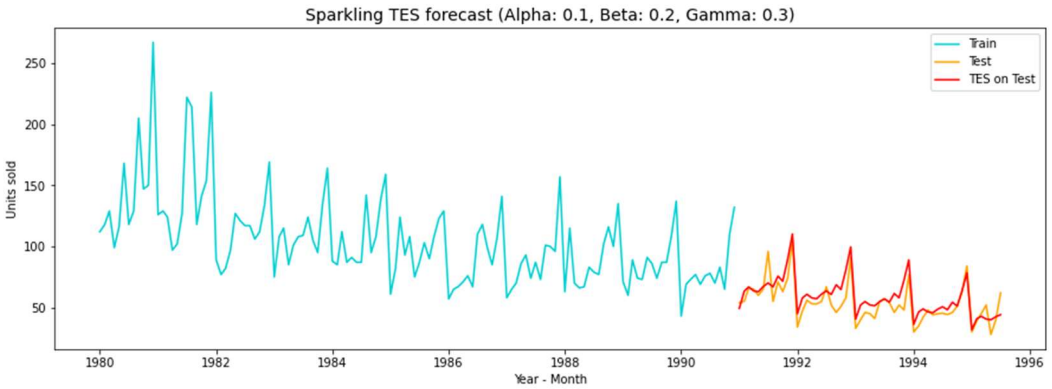
| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |
| Auto_SARIMA_log(0, 1, 1)*(1, 0, 1, 12) | 17.920074 |
| Manual_ARIMA(0,1,0) | 84.160493 |
| Manual_SARIMA(4, 1, 2)*(0, 1, 2, 12) | 15.388806 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |

Table 1: Model comparison based on various parameters

## Iterative Method for Triple Exponential Smoothing:

### Model Evaluation based on Iterations:

| | Alpha Values | Beta Values | Train RMSE | Test RMSE | Gamma Values |
|---|---|---|---|---|---|
| 8 | 0.1 | 0.2 | 23.969166 | 9.943563 | 0.3 |
| 112 | 0.2 | 0.5 | 27.631767 | 10.011658 | 0.3 |
| 170 | 0.3 | 0.2 | 26.806878 | 10.388879 | 0.5 |
| 9 | 0.1 | 0.2 | 23.919163 | 10.392020 | 0.4 |
| 184 | 0.3 | 0.4 | 28.111886 | 10.952348 | 0.3 |



## RMSE VALUES:

| | Test RMSE |
|---|---|
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| RegressionOnTime | 15.278369 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| SimpleAverage | 53.488233 |
| NaiveModel | 79.745697 |

Table 2: RMSE values of Triple Exponential Smoothing

## Rose Forecast v/s Actual Values


## Observations:

- From the comparison of accuracy values and the plot it can be inferred that Triple Exponential Smoothing is the best model, which has trend as well as seasonality components fitting well with the test data.
- 2-point trailing moving average model is also found to have fit well with a slight lag in test dataset.

**1.5 Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment.**

**Solution:**

Augmented Dickey Fuller test is the statistical test to check the stationarity of a time series. The test determines the presence of unit root in the series to understand if the series is stationary or not

**Null Hypothesis: The series has a unit root, that is series is non-stationary.**

**Alternate Hypothesis: The series has no unit root, that is series is stationary.**

- If we fail to reject the null hypothesis, it can say that the series is non-stationary and if we accept the null hypothesis, it can say that the series is stationary.
- The ADF test on the original Rose series retuned the below values, where p-value is greater than alpha .05 so we fail to reject the null hypothesis.

```
Results of Dickey-Fuller Test:
Test Statistic                  -1.872615
p-value                          0.345051
#Lags Used                      13.000000
Number of Observations Used    173.000000
Critical Value (1%)             -3.468726
Critical Value (5%)             -2.878396
Critical Value (10%)            -2.575756
dtype: float64
```

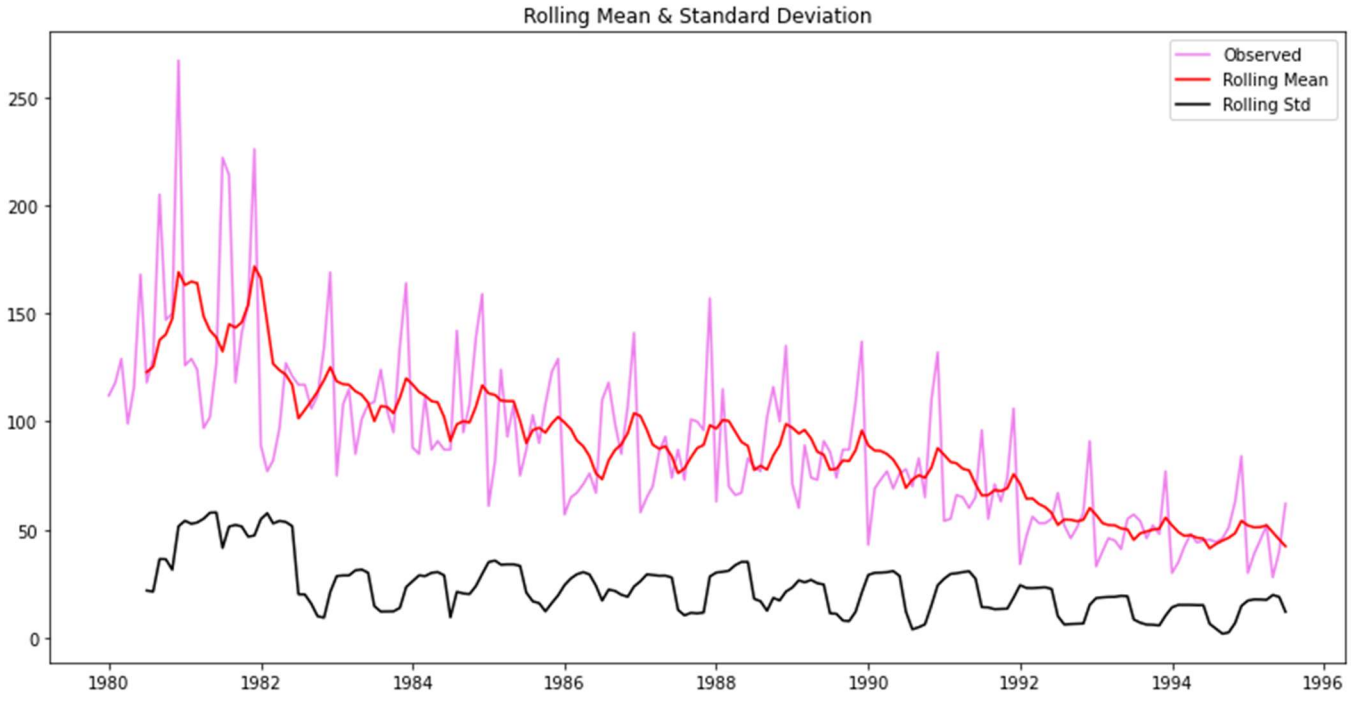
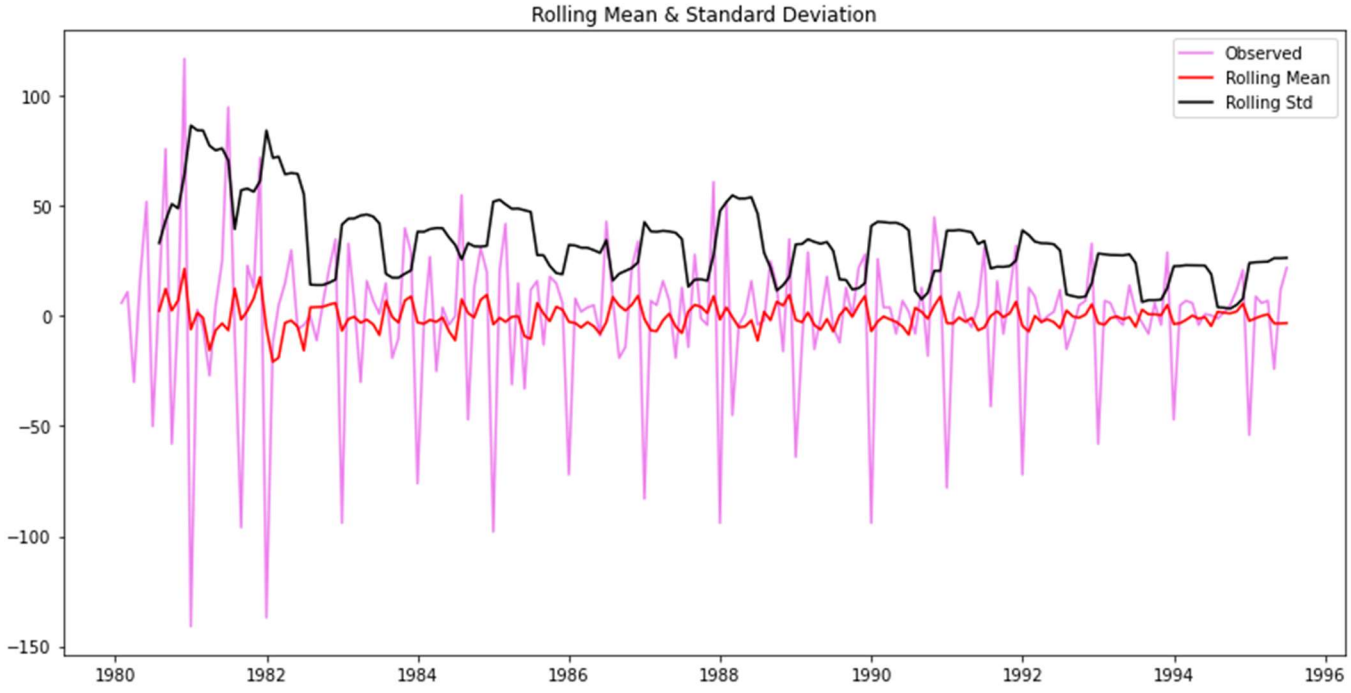



Figure 18: ADF Test on Original Series

## Inferences:

- Differencing of order one is applied on the Rose series as below and tested for stationarity. At an order of differencing 1, the series is found to be stationary as below.
- The rolling means and standard deviation is also plotted to understand the component of seasonality and to ascertain if it's multiplicative or additive in character.
- The altitude of rolling mean and std dev is seen changing according to change in slope, which indicates multiplicity.
- The ADF test is also done in this exercise with logarithmic transformation of the train data and differencing of seasonal order (12), to understand if removing the multiplicity of the seasonal component will have an impact on the accuracy of model.

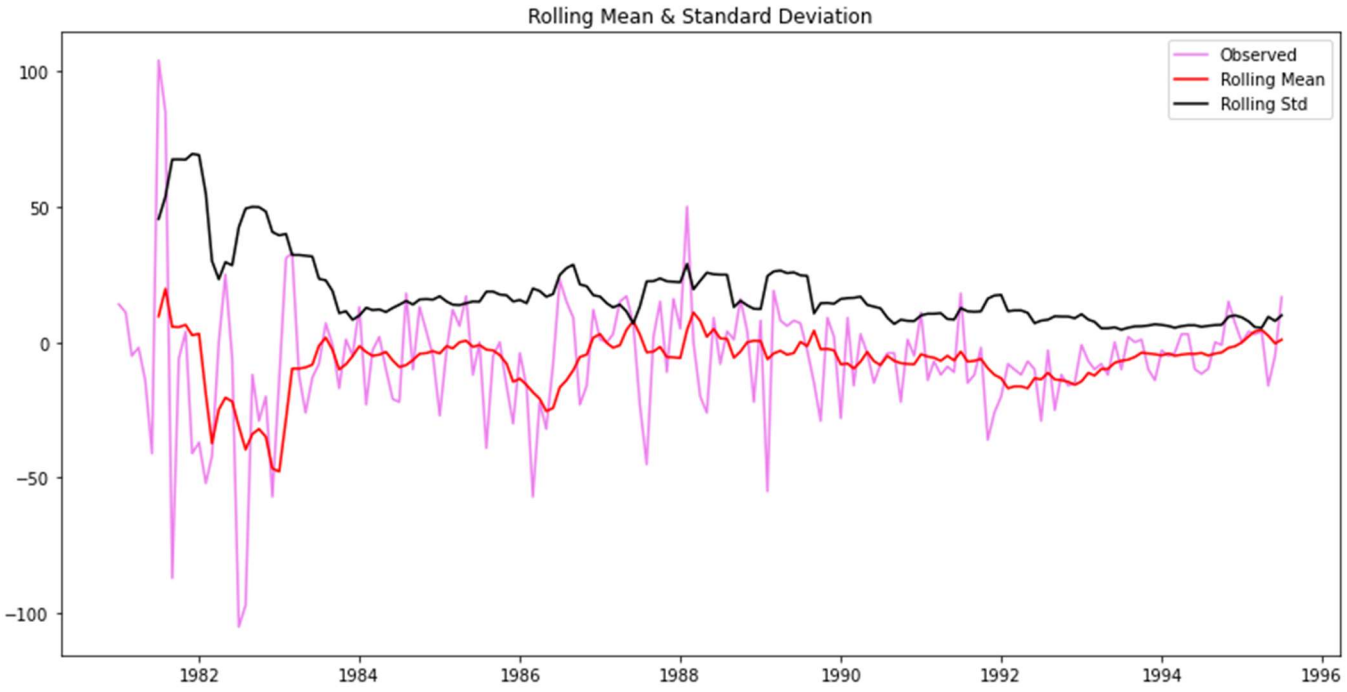
```
Results of Dickey-Fuller Test:
Test Statistic                -8.044081e+00
p-value                        1.814191e-12
#Lags Used                     1.200000e+01
Number of Observations Used    1.730000e+02
Critical Value (1%)           -3.468726e+00
Critical Value (5%)           -2.878396e+00
Critical Value (10%)          -2.575756e+00
dtype: float64
```



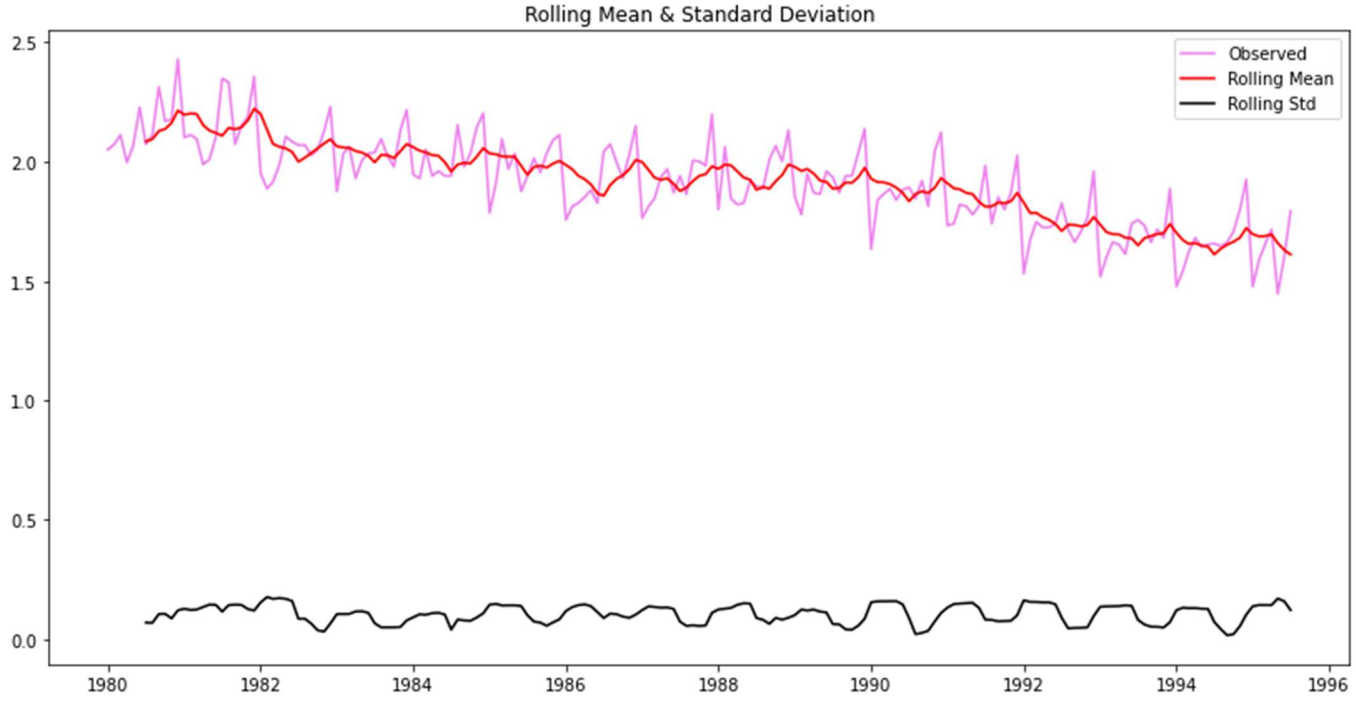

**ADF Test After differencing d=1**

## Check Seasonality Time Series:

```
Results of Dickey-Fuller Test:
Test Statistic                  -4.257265
p-value                          0.000526
#Lags Used                      11.000000
Number of Observations Used    163.000000
Critical Value (1%)             -3.471119
Critical Value (5%)             -2.879441
Critical Value (10%)            -2.576314
dtype: float64
```

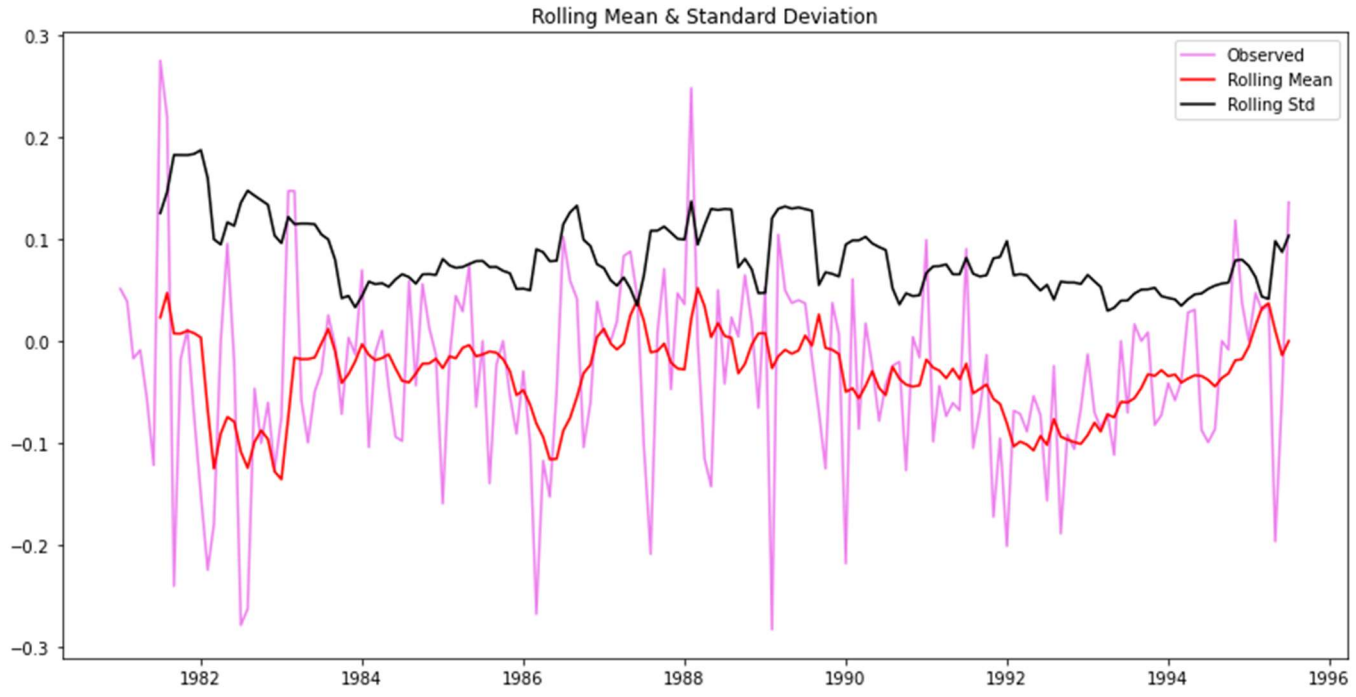



```
Results of Dickey-Fuller Test:
Test Statistic                  -0.412363
p-value                          0.908014
#Lags Used                      12.000000
Number of Observations Used    174.000000
Critical Value (1%)             -3.468502
Critical Value (5%)             -2.878298
Critical Value (10%)            -2.575704
dtype: float64
```




## ADF Test on Log Series

```
Results of Dickey-Fuller Test:
Test Statistic                  -3.934772
p-value                          0.001793
#Lags Used                      11.000000
Number of Observations Used    163.000000
Critical Value (1%)             -3.471119
Critical Value (5%)             -2.879441
Critical Value (10%)            -2.576314
dtype: float64
```




## ADF Test on Log Series after differencing

## Plotting the Autocorrelation and the Partial Autocorrelation function plots on the whole data:

### ACF Plots:

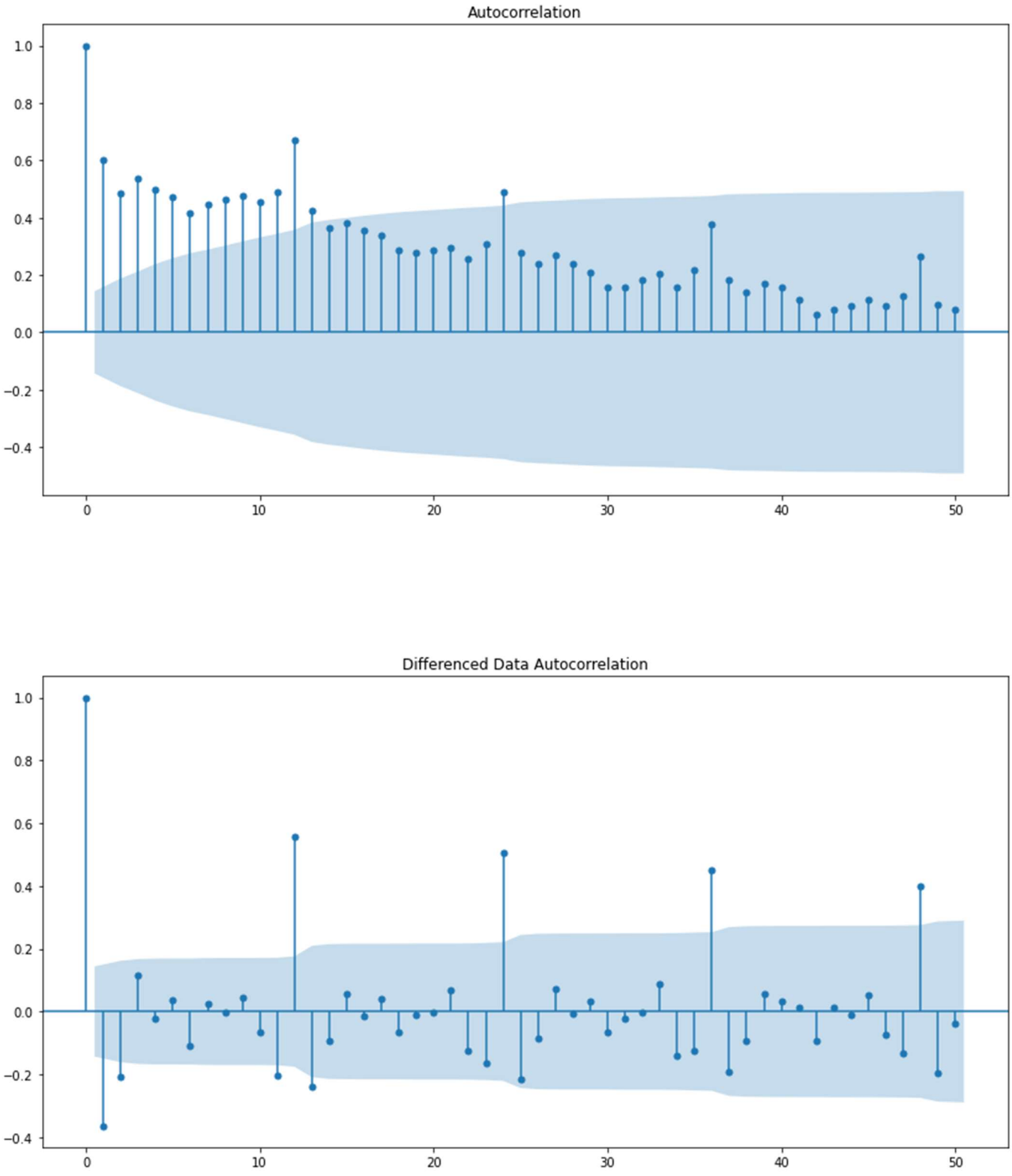



Figure 19: Autocorrelation and Differenced Data Autocorrelation of Rose dataset

## PACF Plots:

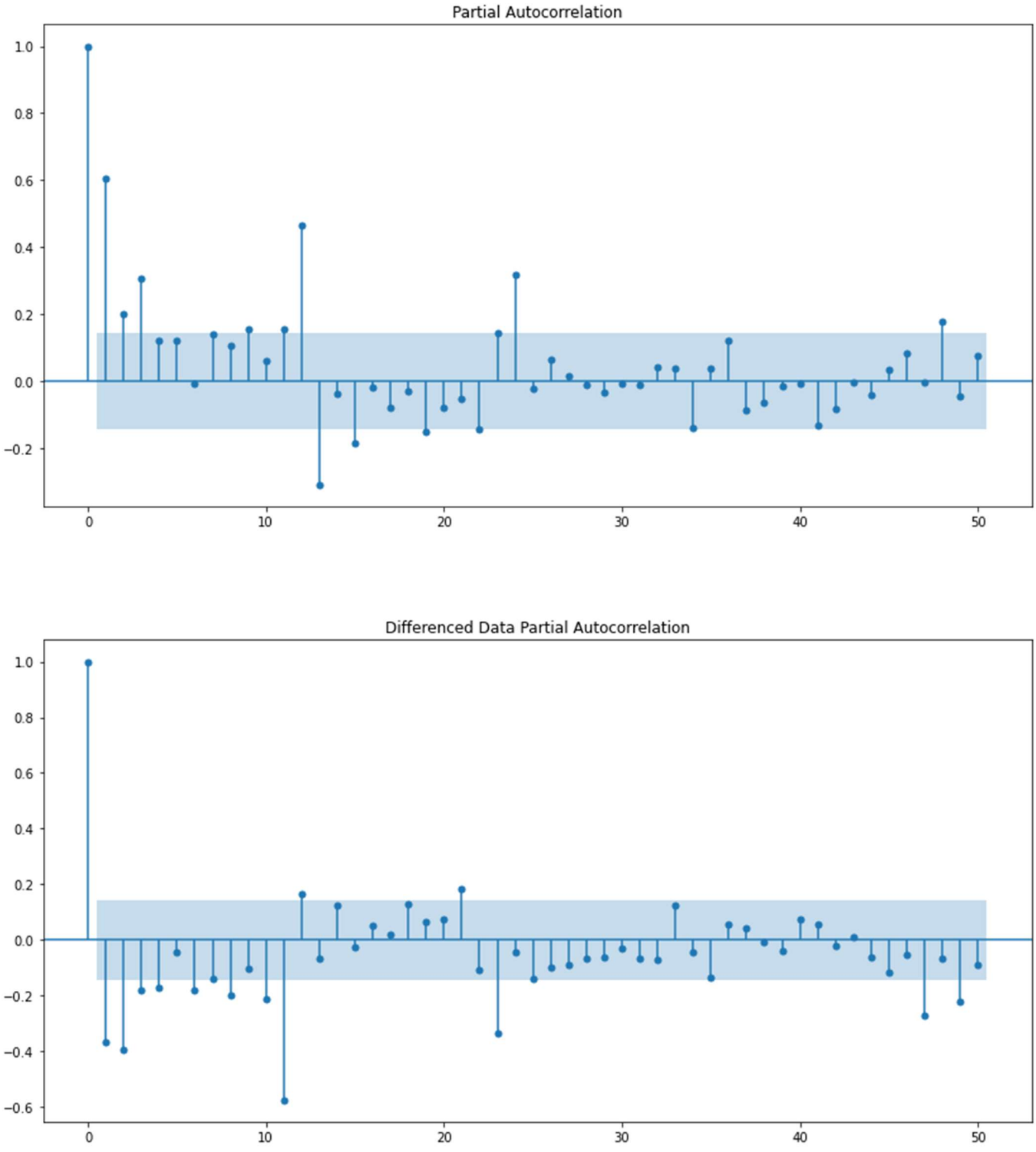
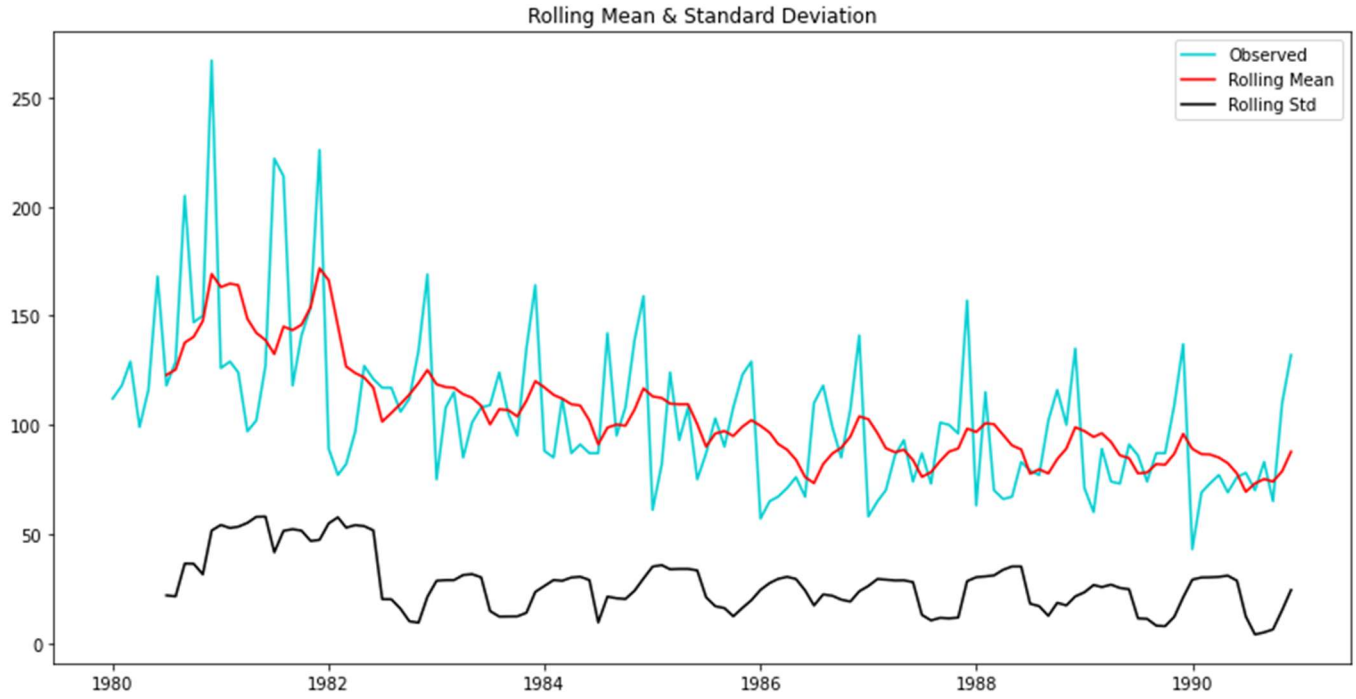



Figure 20: Partial Autocorrelation and Differenced Data Partial Autocorrelation of Rose dataset

From the above plots, we can say that there seems to be a seasonality in the data.

## Checking the stationarity of the Training Data Time Series:

```
Results of Dickey-Fuller Test:
Test Statistic                  -2.164250
p-value                          0.219476
#Lags Used                      13.000000
Number of Observations Used    118.000000
Critical Value (1%)             -3.487022
Critical Value (5%)             -2.886363
Critical Value (10%)            -2.580009
dtype: float64
```




**Stationarity of Training Data Time Series**

### Inferences:

We see that at 5% significant level the Time Series is non-stationary. Let us take a difference of order 1 and check whether the Time Series is stationary or not.

```
Results of Dickey-Fuller Test:
Test Statistic                -6.592372e+00
p-value                        7.061944e-09
#Lags Used                     1.200000e+01
Number of Observations Used    1.180000e+02
Critical Value (1%)           -3.487022e+00
Critical Value (5%)           -2.886363e+00
Critical Value (10%)          -2.580009e+00
dtype: float64
```

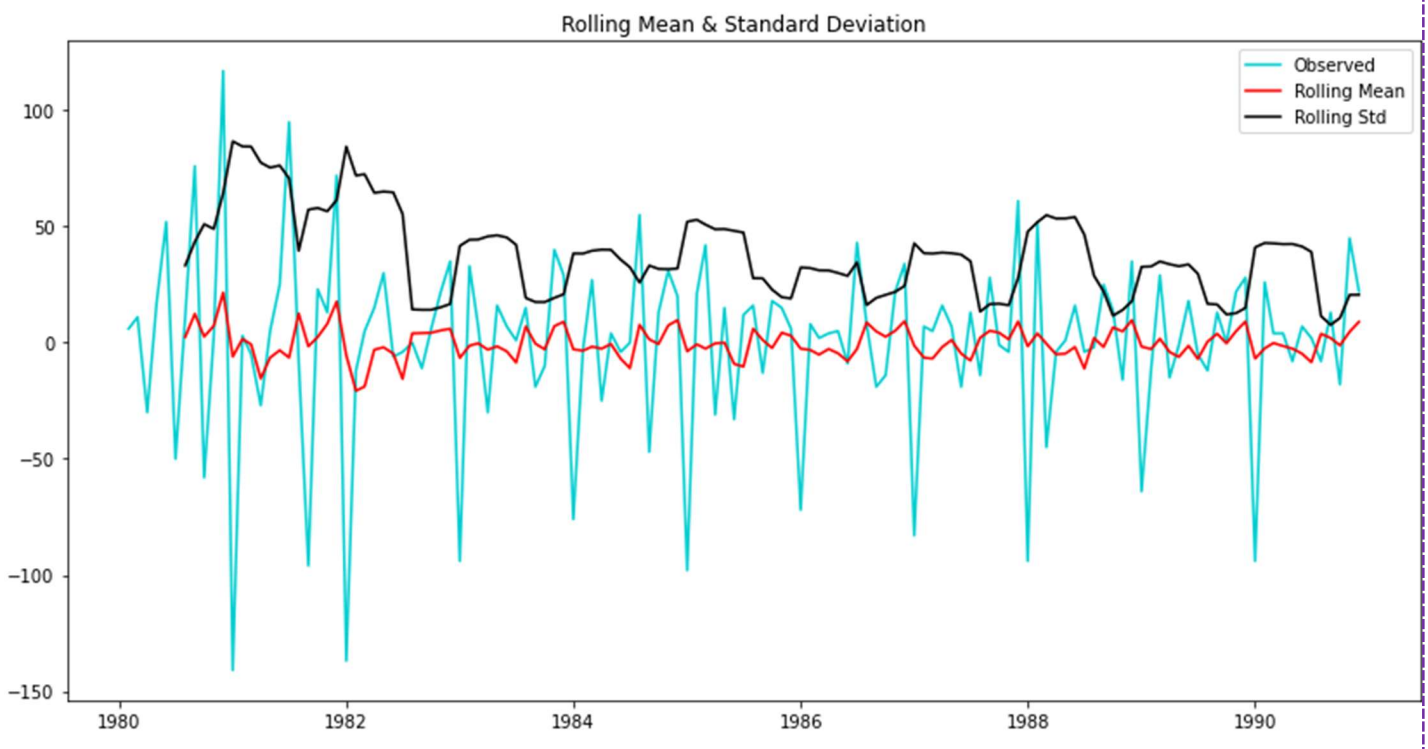



Figure 21: Stationarity of Training Data Time Series before and after differencing

**1.6 Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE.**

**Solution:**

## Auto-ARIMA Model:

Some parameter combinations for the Model...

```
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
```

## AIC Values:

| | param | AIC |
|---|---|---|
| 2 | (0, 1, 2) | 1276.835373 |
| 5 | (1, 1, 2) | 1277.359228 |
| 4 | (1, 1, 1) | 1277.775754 |
| 7 | (2, 1, 1) | 1279.045689 |
| 8 | (2, 1, 2) | 1279.298694 |
| 1 | (0, 1, 1) | 1280.726183 |
| 6 | (2, 1, 0) | 1300.609261 |
| 3 | (1, 1, 0) | 1319.348311 |
| 0 | (0, 1, 0) | 1335.152658 |

AIC values has been sorted in the ascending order to get the parameters for the minimum AIC value.

ARIMA Model Results

| | | | |
|---|---|---|---|
| Dep. Variable: | D.Rose | No. Observations: | 131 |
| Model: | ARIMA(0, 1, 2) | Log Likelihood | -634.418 |
| Method: | css-mle | S.D. of innovations | 30.167 |
| Date: | Fri, 15 Jul 2022 | AIC | 1276.835 |
| Time: | 18:51:48 | BIC | 1288.336 |
| Sample: | 02-01-1980 - 12-01-1990 | HQIC | 1281.509 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | -0.4886 | 0.085 | -5.742 | 0.000 | -0.655 | -0.322 |
| ma.L1.D.Rose | -0.7601 | 0.101 | -7.499 | 0.000 | -0.959 | -0.561 |
| ma.L2.D.Rose | -0.2398 | 0.095 | -2.518 | 0.012 | -0.427 | -0.053 |

Roots

| | Real | Imaginary | Modulus | Frequency |
|---|---|---|---|---|
| MA.1 | 1.0001 | +0.0000j | 1.0001 | 0.0000 |
| MA.2 | -4.1695 | +0.0000j | 4.1695 | 0.5000 |

Figure 22: Auto ARIMA Model Summary Report

ARIMA model was built with optimized model and found the least AIC value =1276 at (0, 1, 2). As the Rose series of data contain seasonality component, ARIMA model do not perform well. The RMSE value for this Auto- ARIMA model is 15.63.

## RMSE Values:

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |

## Auto-SARIMA Model:

Examples of some parameter combinations for Model...

```
Model: (0, 1, 1)(0, 1, 1, 12)
Model: (0, 1, 2)(0, 1, 2, 12)
Model: (1, 1, 0)(1, 1, 0, 12)
Model: (1, 1, 1)(1, 1, 1, 12)
Model: (1, 1, 2)(1, 1, 2, 12)
Model: (2, 1, 0)(2, 1, 0, 12)
Model: (2, 1, 1)(2, 1, 1, 12)
Model: (2, 1, 2)(2, 1, 2, 12)
```

## AIC Values :

| | param | seasonal | AIC |
|---|---|---|---|
| 26 | (0, 1, 2) | (2, 1, 2, 12) | 774.969122 |
| 53 | (1, 1, 2) | (2, 1, 2, 12) | 776.940108 |
| 80 | (2, 1, 2) | (2, 1, 2, 12) | 776.996101 |
| 17 | (0, 1, 1) | (2, 1, 2, 12) | 782.153872 |
| 79 | (2, 1, 2) | (2, 1, 1, 12) | 783.703652 |

SARIMAX Results

| | | | |
|---|---|---|---|
| Dep. Variable: | y | No. Observations: | 132 |
| Model: | SARIMAX(0, 1, 2)x(2, 1, 2, 12) | Log Likelihood | -380.485 |
| Date: | Fri, 15 Jul 2022 | AIC | 774.969 |
| Time: | 18:53:45 | BIC | 792.622 |
| Sample: | 0 | HQIC | 782.094 |
| | - 132 | | |
| Covariance Type: | opg | | |

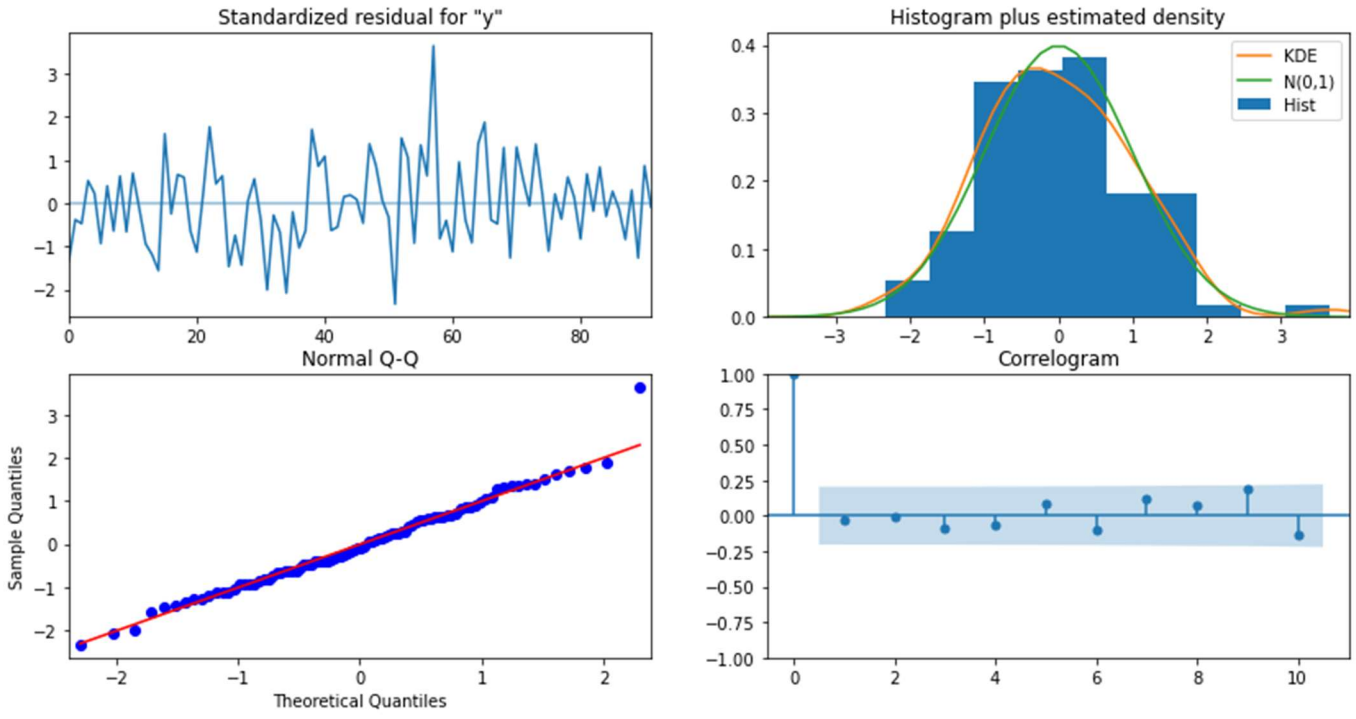
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ma.L1 | -0.9524 | 0.184 | -5.170 | 0.000 | -1.313 | -0.591 |
| ma.L2 | -0.0764 | 0.126 | -0.606 | 0.545 | -0.324 | 0.171 |
| ar.S.L12 | 0.0480 | 0.177 | 0.271 | 0.786 | -0.299 | 0.395 |
| ar.S.L24 | -0.0419 | 0.028 | -1.513 | 0.130 | -0.096 | 0.012 |
| ma.S.L12 | -0.7526 | 0.301 | -2.503 | 0.012 | -1.342 | -0.163 |
| ma.S.L24 | -0.0721 | 0.204 | -0.354 | 0.723 | -0.472 | 0.327 |
| sigma2 | 187.8364 | 45.261 | 4.150 | 0.000 | 99.127 | 276.546 |

| | | | |
|---|---|---|---|
| Ljung-Box (L1) (Q): | 0.06 | Jarque-Bera (JB): | 4.86 |
| Prob(Q): | 0.81 | Prob(JB): | 0.09 |
| Heteroskedasticity (H): | 0.91 | Skew: | 0.41 |
| Prob(H) (two-sided): | 0.79 | Kurtosis: | 3.77 |

Figure 23: SARIMA Model Result

The model was built on train data with seasonality 12 and with different optimal parameters (p, d, q)x(P, D, Q) parameters, the lowest AIC is 774.97 was obtained at (0, 1, 2)x(2, 1, 2, 12).The model was built with the above parameters.

## Diagnostic Plot:

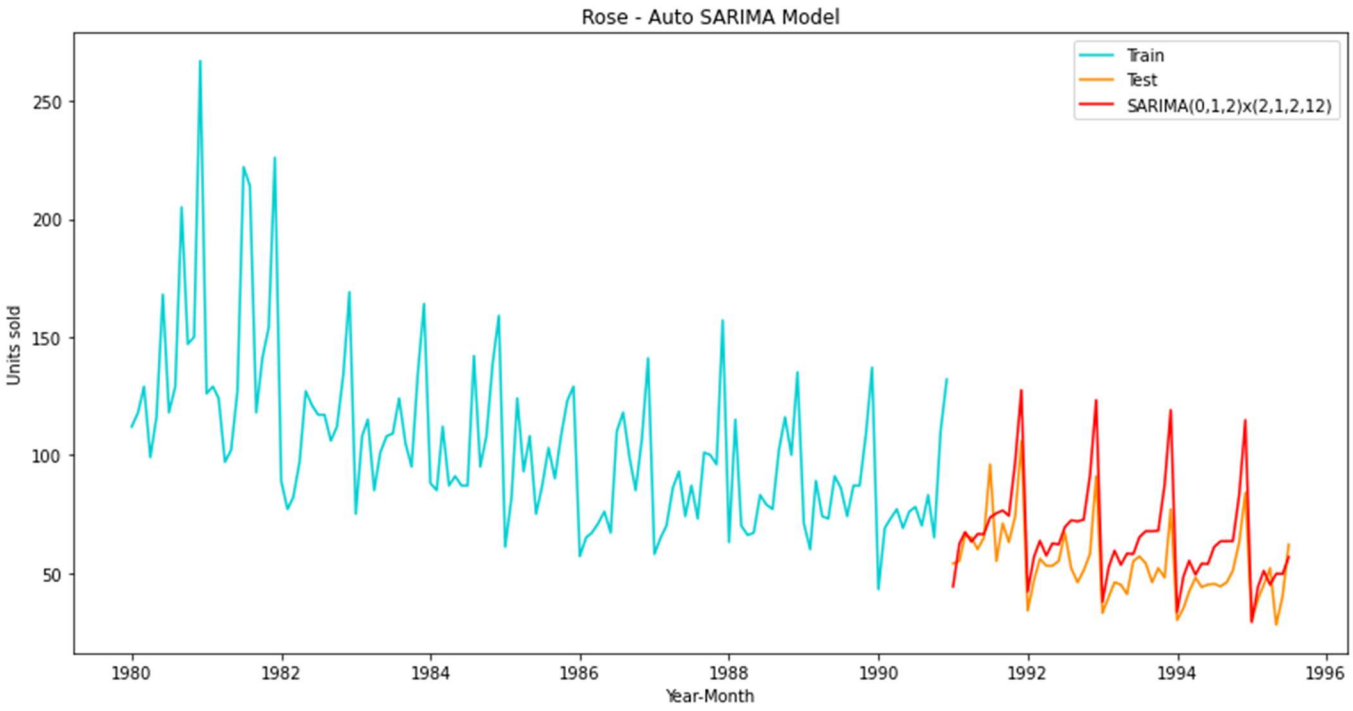


## Observations:

- The diagnostics plot of the model was derived and the standardized residuals are found to follow a mean of zero, and the histogram shows the residuals follow a normal distribution.
- The Normal Q-Q plot also shows that the quantiles come from a normal distribution as the point forms roughly a straight line.
- The correlogram shows the autocorrelation of the residuals and there are no significant lags above the confidence index.
- The RMSE values of the automated SARIMA model is 16.53.

**Extracting the predicted and true values of our time series:**

| YearMonth | Rose | rose_forecasted |
|---|---|---|
| 1991-01-01 | 54.0 | 44.214702 |
| 1991-02-01 | 55.0 | 62.327974 |
| 1991-03-01 | 66.0 | 67.315152 |
| 1991-04-01 | 65.0 | 63.162471 |
| 1991-05-01 | 60.0 | 66.476733 |




**Plot of Actual v/s Forecasted Result on test data**

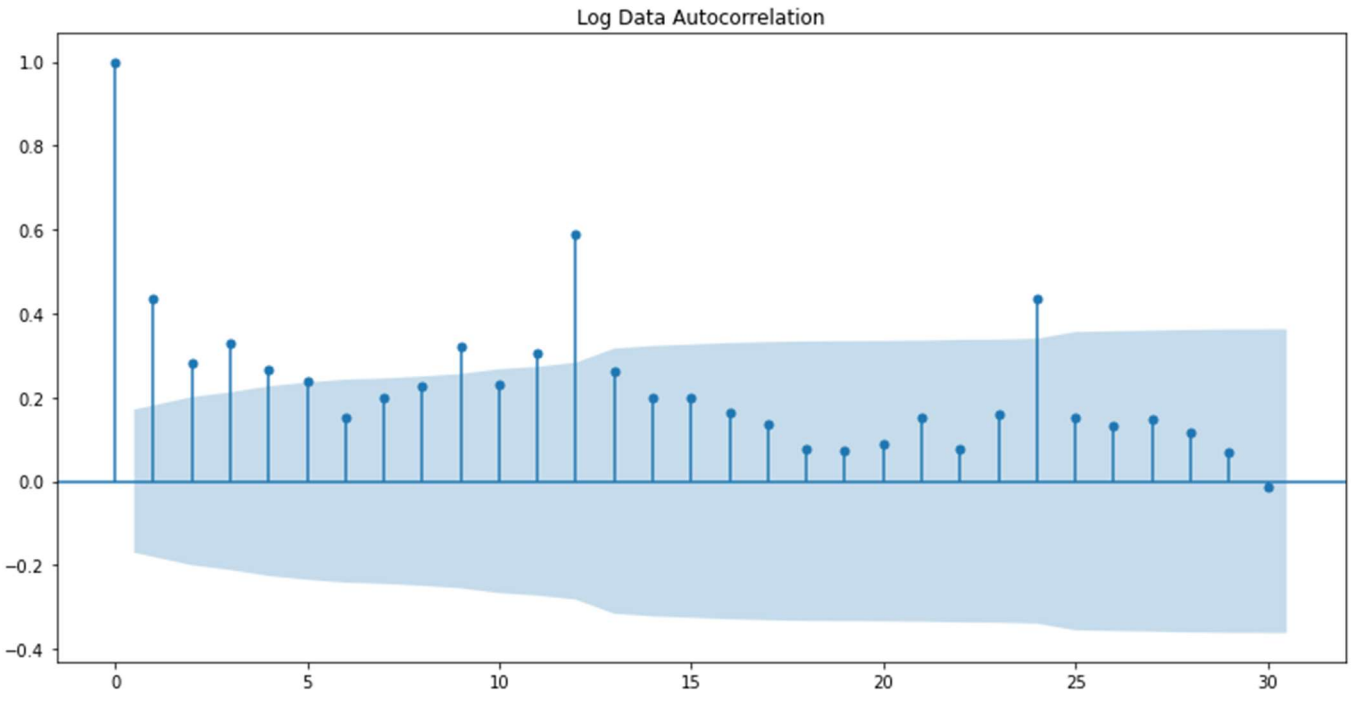
### RMSE Values:

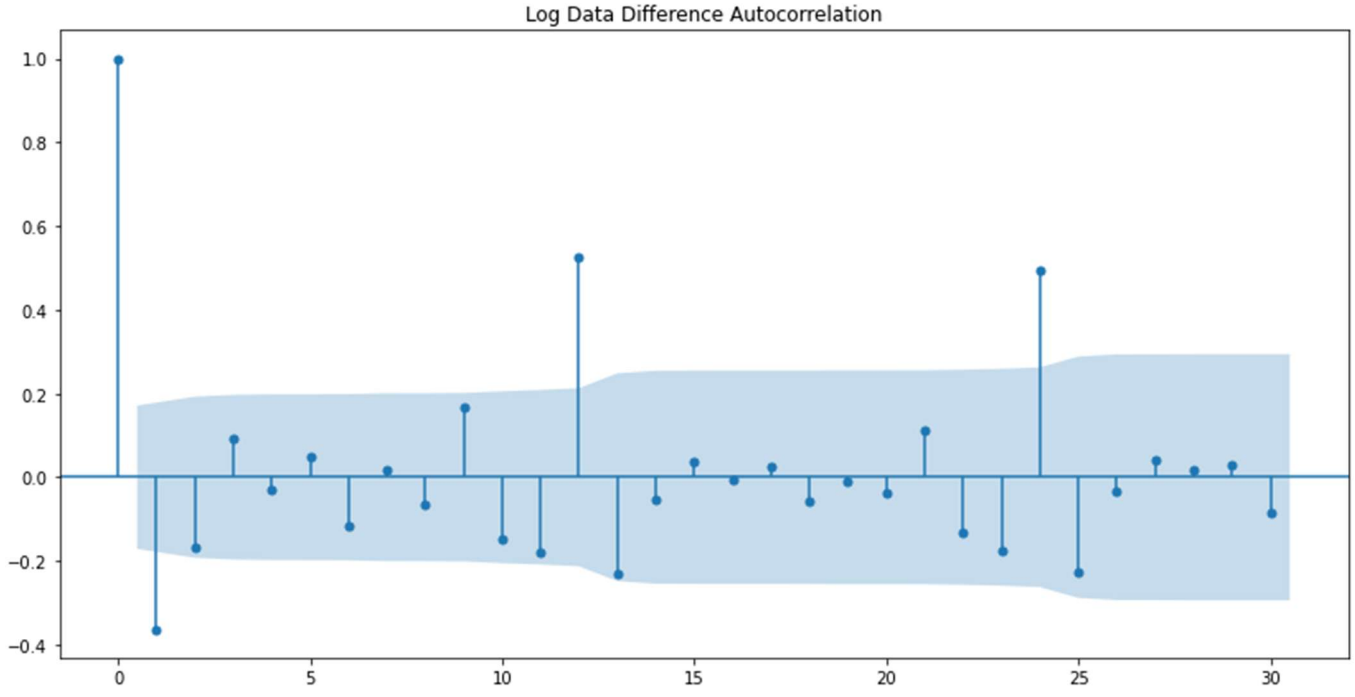
| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |

## AUTO SARIMA Model on Log Series data:

The model was built on log transformed train data and with seasonality 12 and with different optimal parameters (p, d, q)x(P, D, Q) parameters, the lowest AIC is -247.08 was obtained at (0, 1, 1)*(1, 0, 1, 12).

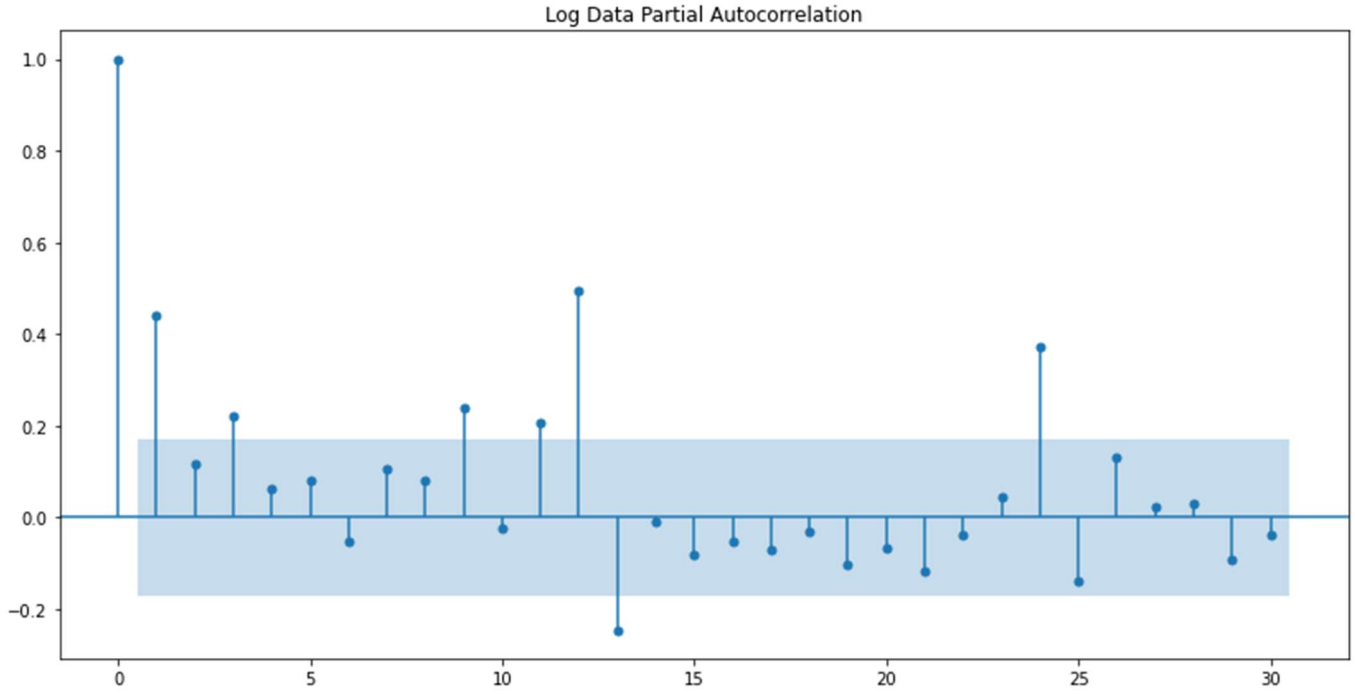
### ACF Plot:

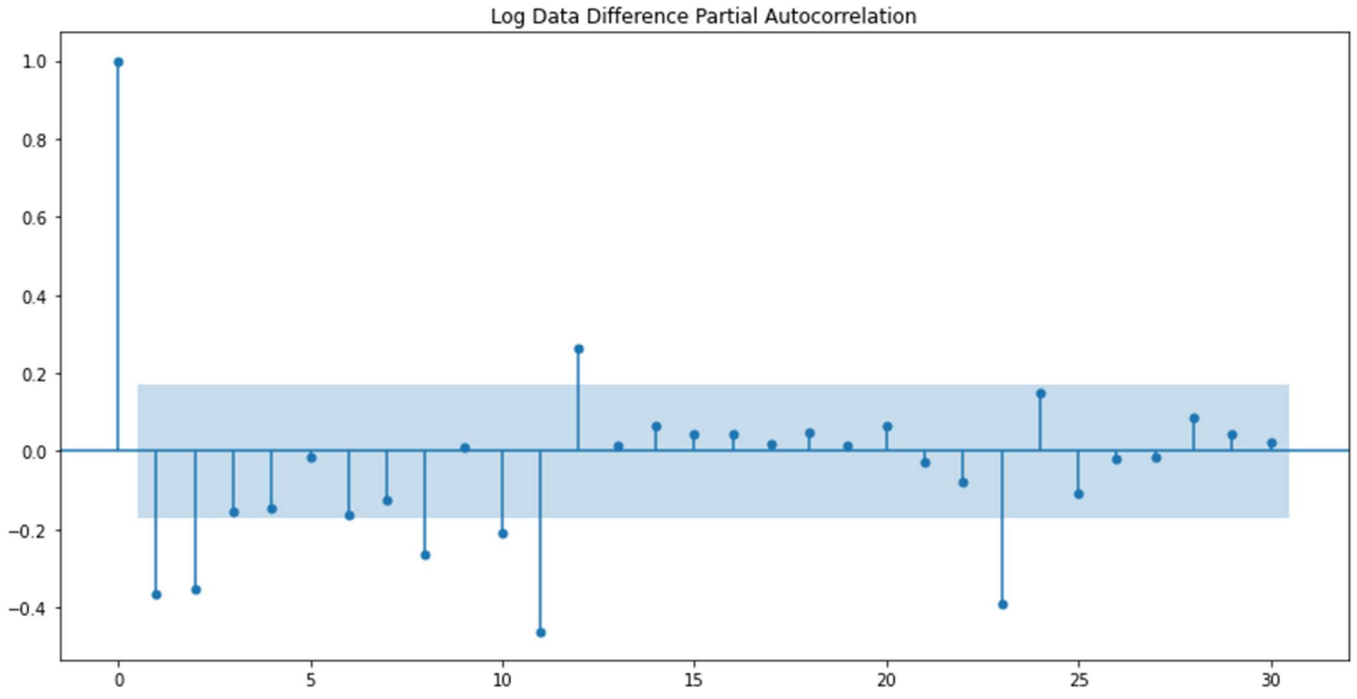





**ACF Plot for Log Series data**

## PACF Plot:






**PACF Plot for Log Series data**

**Examples of some parameter combinations for Model...**

```
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (1, 1, 0)(0, 1, 0, 12)
Model: (1, 1, 1)(0, 1, 1, 12)
Model: (1, 1, 2)(0, 1, 2, 12)
Model: (2, 1, 0)(1, 0, 0, 12)
Model: (2, 1, 1)(1, 0, 1, 12)
Model: (2, 1, 2)(1, 0, 2, 12)
```

**AIC Values for log series:**

| | param | seasonal | AIC |
|---|---|---|---|
| **25** | (0, 1, 1) | (1, 0, 1, 12) | -247.076408 |
| **79** | (1, 1, 1) | (1, 0, 1, 12) | -246.523803 |
| **133** | (2, 1, 1) | (1, 0, 1, 12) | -246.471941 |
| **97** | (1, 1, 2) | (1, 0, 1, 12) | -245.792457 |
| **22** | (0, 1, 1) | (0, 1, 1, 12) | -245.315623 |

SARIMAX Results

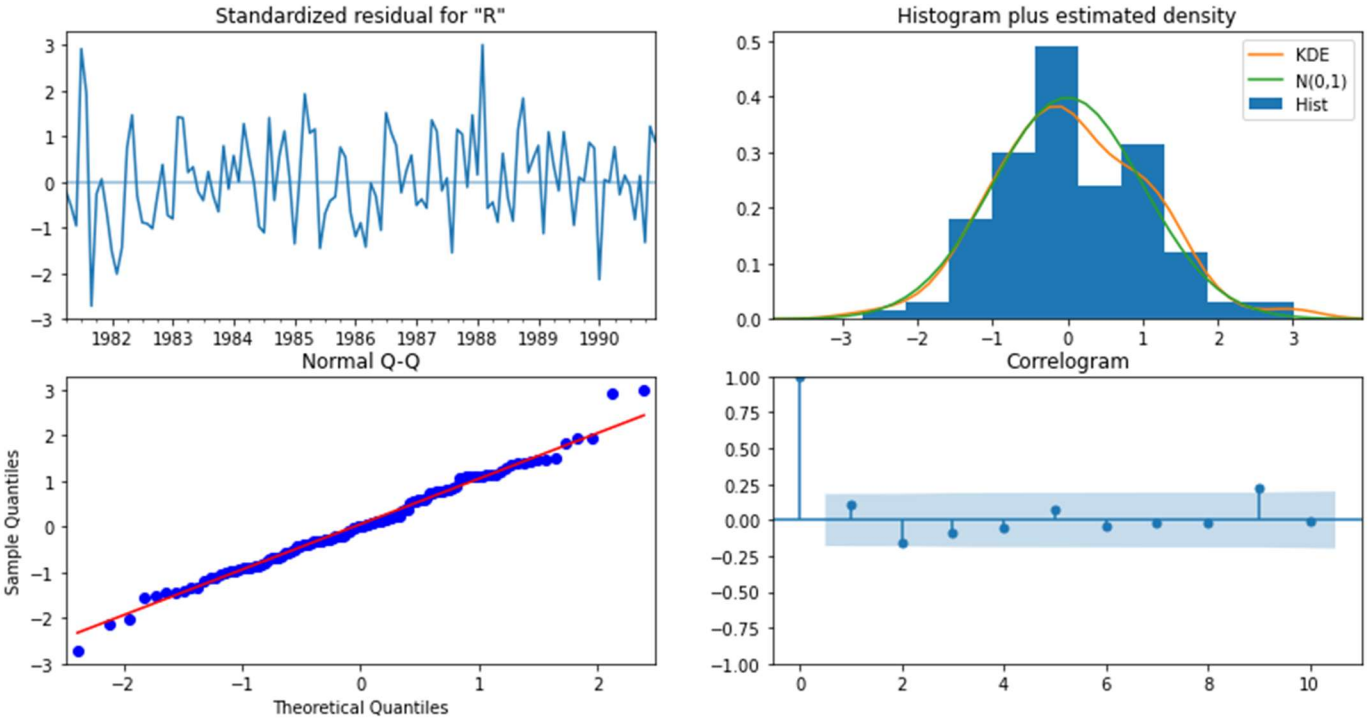
| | | | |
|---|---|---|---|
| Dep. Variable: | Rose | No. Observations: | 132 |
| Model: | SARIMAX(0, 1, 1)x(1, 0, 1, 12) | Log Likelihood | 127.538 |
| Date: | Sat, 16 Jul 2022 | AIC | -247.076 |
| Time: | 09:40:28 | BIC | -236.028 |
| Sample: | 01-01-1980 - 12-01-1990 | HQIC | -242.591 |
| Covariance Type: | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ma.L1 | -1.0652 | 0.058 | -18.389 | 0.000 | -1.179 | -0.952 |
| ar.S.L12 | 0.9555 | 0.028 | 33.775 | 0.000 | 0.900 | 1.011 |
| ma.S.L12 | -0.8303 | 0.151 | -5.498 | 0.000 | -1.126 | -0.534 |
| sigma2 | 0.0051 | 0.001 | 5.146 | 0.000 | 0.003 | 0.007 |

| | | | |
|---|---|---|---|
| Ljung-Box (L1) (Q): | 1.31 | Jarque-Bera (JB): | 0.98 |
| Prob(Q): | 0.25 | Prob(JB): | 0.61 |
| Heteroskedasticity (H): | 0.80 | Skew: | 0.18 |
| Prob(H) (two-sided): | 0.50 | Kurtosis: | 3.26 |

Figure 24: Log Series SARIMA Model Summary Result

## Diagnostic Plot:

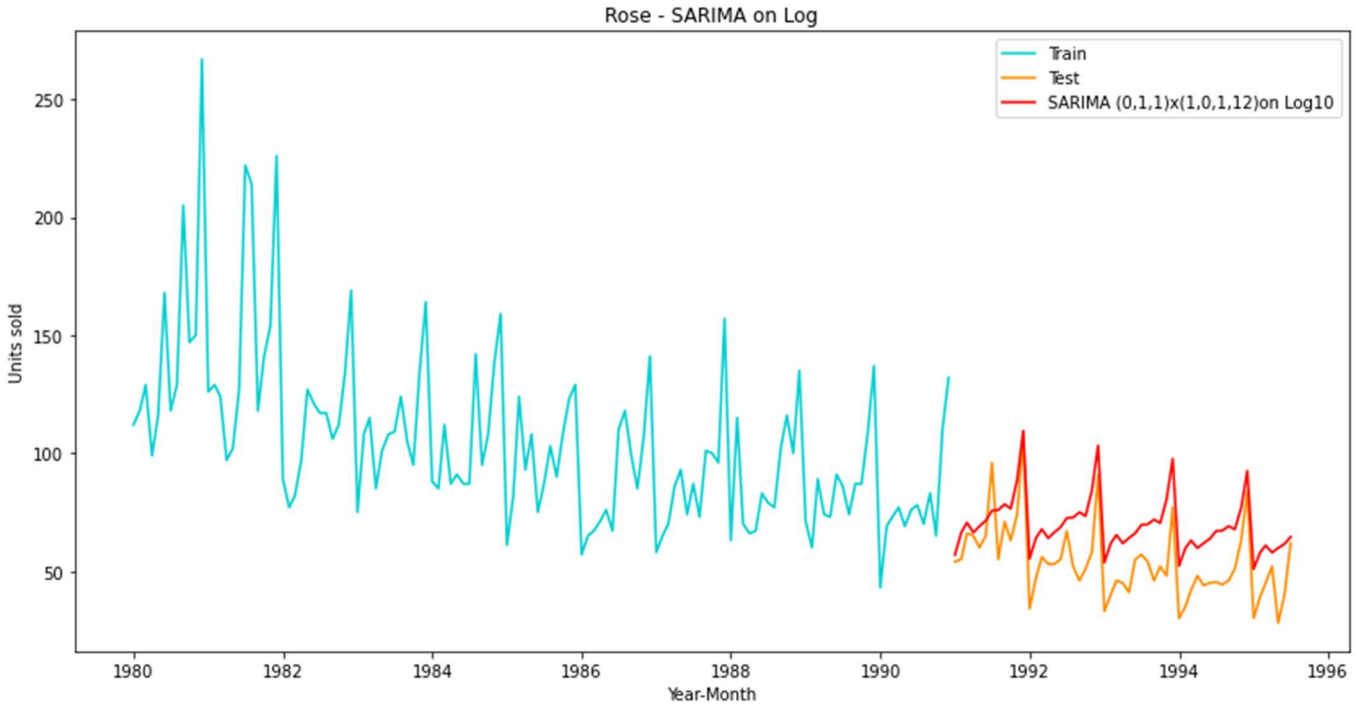



**Diagnostic Plot of Log Series Model**

## Extracting the predicted and true values of our time series:

| YearMonth | Rose | rose_forecasted | rose_forecasted_log |
|---|---|---|---|
| 1991-01-01 | 54.0 | 44.214702 | 56.850909 |
| 1991-02-01 | 55.0 | 62.327974 | 66.337363 |
| 1991-03-01 | 66.0 | 67.315152 | 70.531259 |
| 1991-04-01 | 65.0 | 63.162471 | 66.371329 |
| 1991-05-01 | 60.0 | 66.476733 | 69.031507 |

**Forecasted Value of Log series**




**Plot of Actual and Forecasted values of Auto SARIMA Model of Log Series**

### Inferences:

- The diagnostics plot of the model was derived and the standardized residuals are found to follow a mean of zero, and the histogram shows the residuals follow a normal distribution.
- The Normal Q-Q plot also shows that the quantiles come from a normal distribution as the point forms roughly a straight line.
- The correlogram shows the autocorrelation of the residuals and there are no significant lags above the confidence index.

- From the above model summary, it can be inferred that MA.L1, AR.L.S12, MA.L.S12 terms has the highest absolute weightage.
- From the p-values it can be inferred that terms MA.L1, AR.L.S12, MA.L.S12 are significant terms, as their values are below 0.05.
- The RMSE values of the automated SARIMA of log series model is 17.93.
- The model built with log series data has a higher RMSE value when compared to original train data.

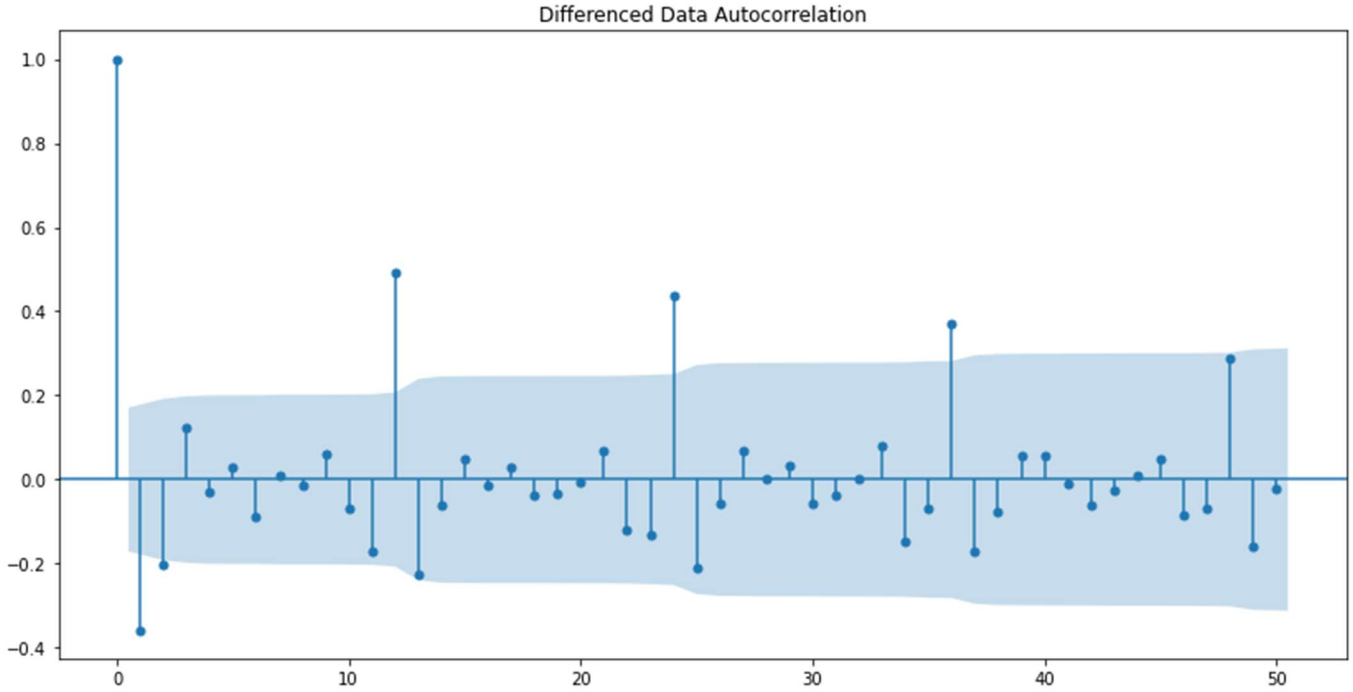
**RMSE Values:**

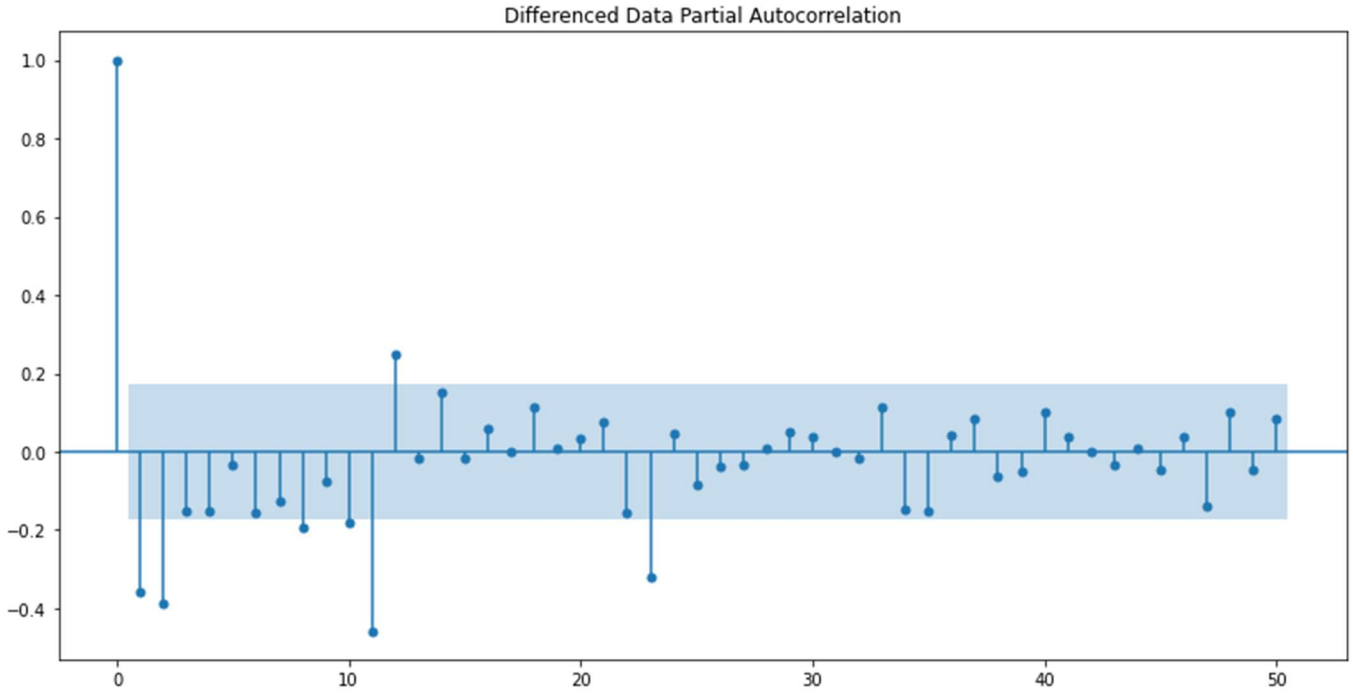
| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |
| Auto_SARIMA_log(0, 1, 1)*(1, 0, 1, 12) | 17.920074 |

**1.7 Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.**

**Solution:**

## Manual ARIMA Model:




- Here, we have taken alpha = 0.05.
- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 0.

- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 0.
- By looking at above plots, we can say that both the PACF and ACF plot cuts-off at lag 0.

ARIMA Model Results

| | | | |
|---|---|---|---|
| Dep. Variable: | D.Rose | No. Observations: | 131 |
| Model: | ARIMA(0, 1, 0) | Log Likelihood | -665.576 |
| Method: | css | S.D. of innovations | 38.931 |
| Date: | Sat, 16 Jul 2022 | AIC | 1335.153 |
| Time: | 09:40:30 | BIC | 1340.903 |
| Sample: | 02-01-1980 - 12-01-1990 | HQIC | 1337.489 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 0.1527 | 3.401 | 0.045 | 0.964 | -6.514 | 6.819 |

**Manual ARIMA Summary Result**

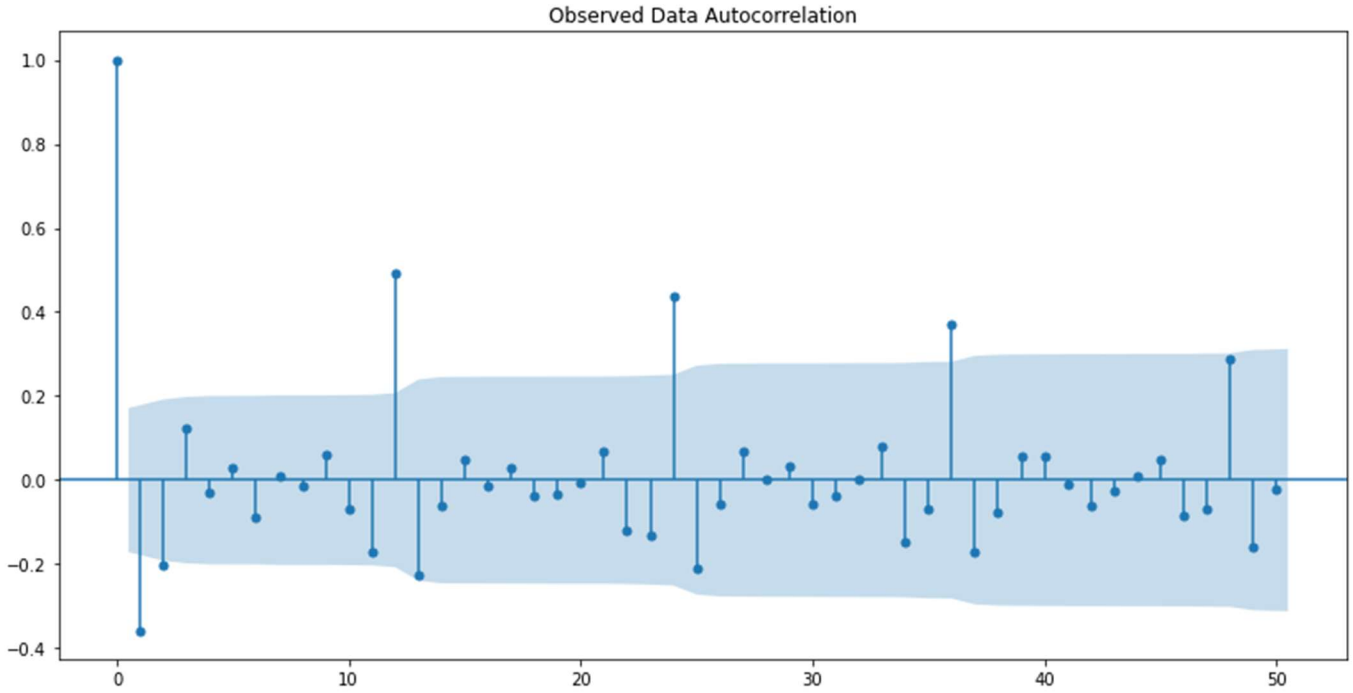
**The RMSE value of manual ARIMA model is 84.16. Since the ARIMA model do not capture the seasonality, this model does not perform well.**

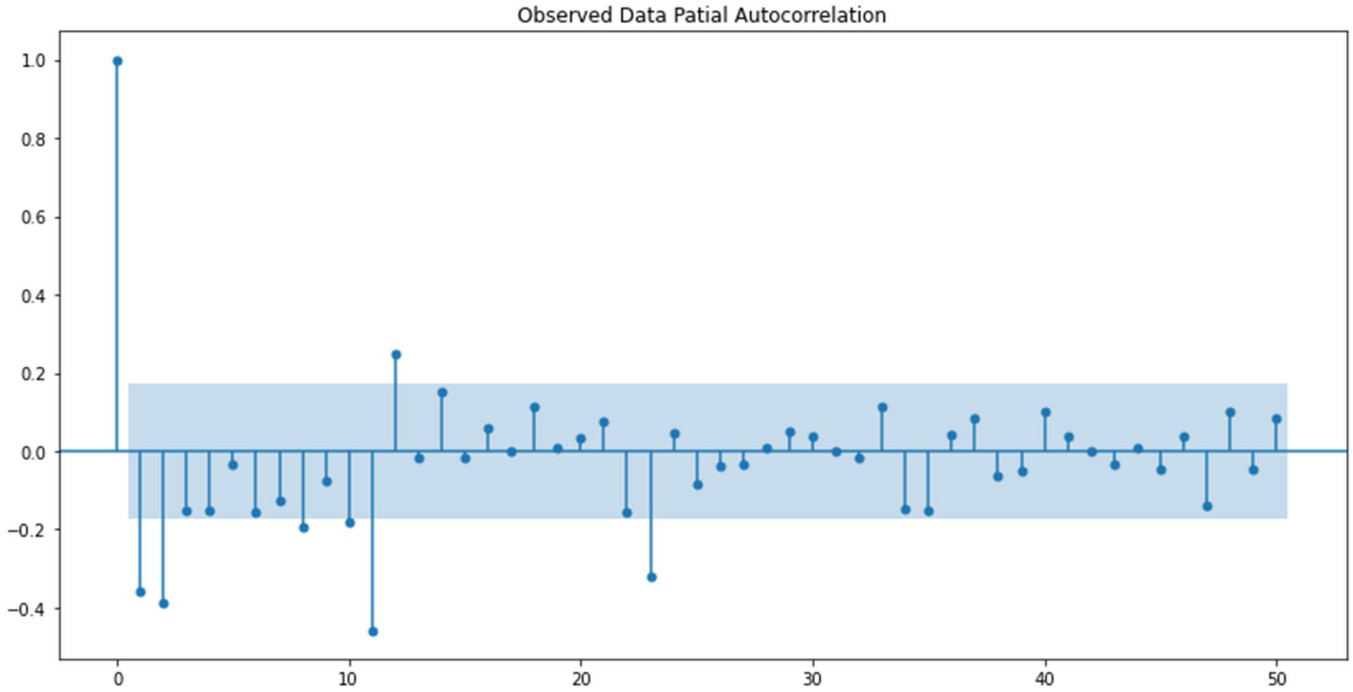
### RMSE Values:

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |
| Auto_SARIMA_log(0, 1, 1)*(1, 0, 1, 12) | 17.920074 |
| Manual_ARIMA(0,1,0) | 84.160493 |

**The data has some seasonality so we should build a SARIMA model to get better accuracy.**

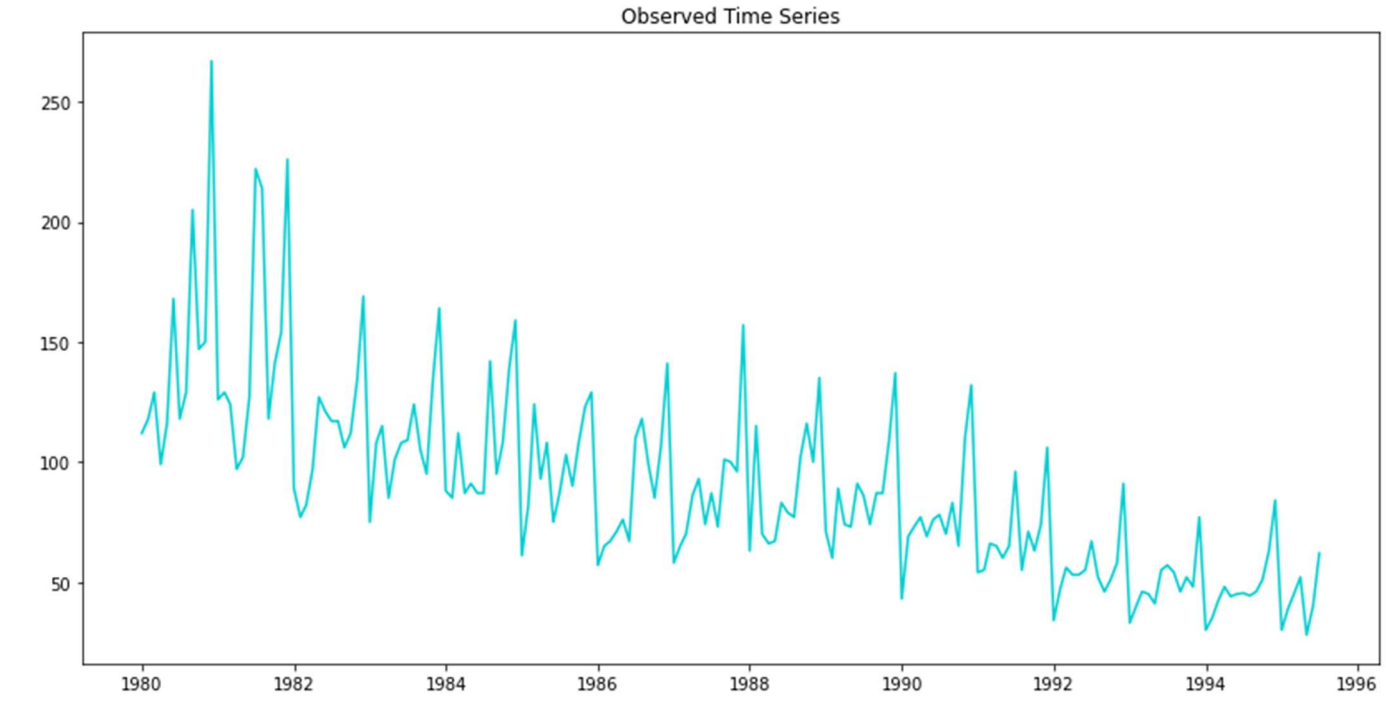
## Manual SARIMA Model:



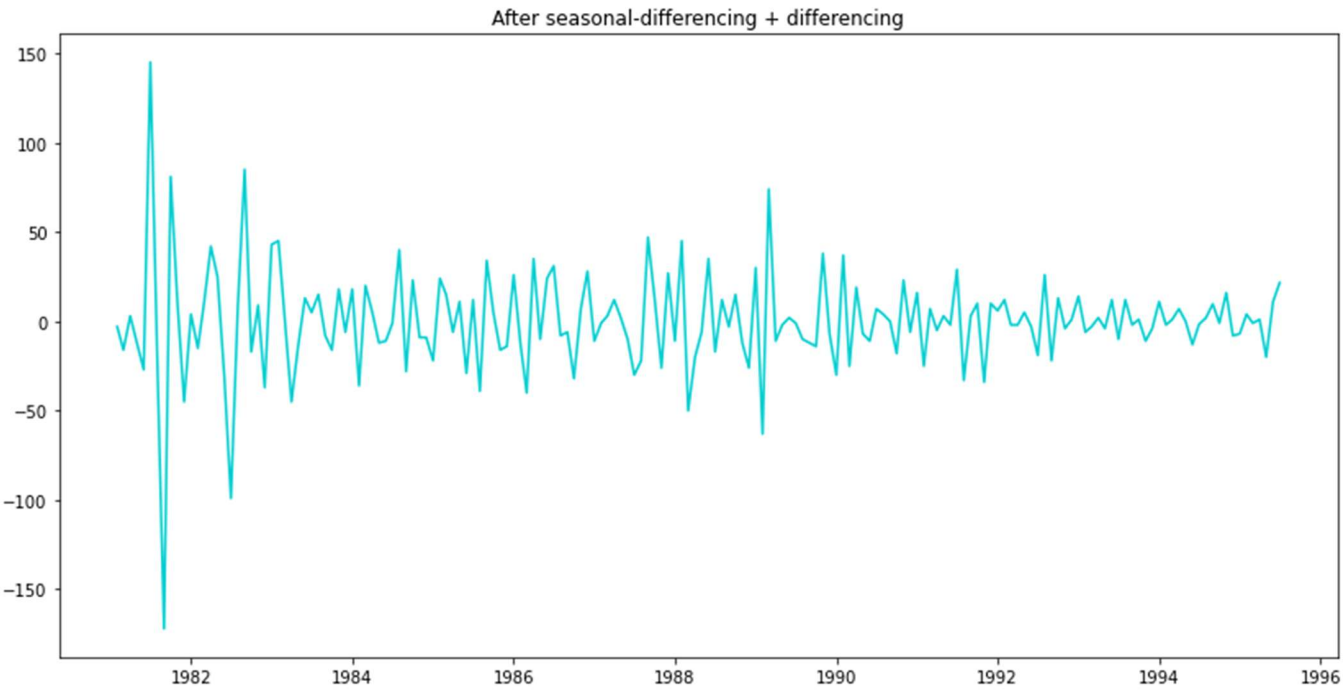




**ACF and PACF Plot of SARIMA Model**

From the ACF plot of the observed/ train data, it can be inferred that at seasonal interval of 12, the plot is not quickly tapering off. So, a seasonal differencing of 12 has to be taken.

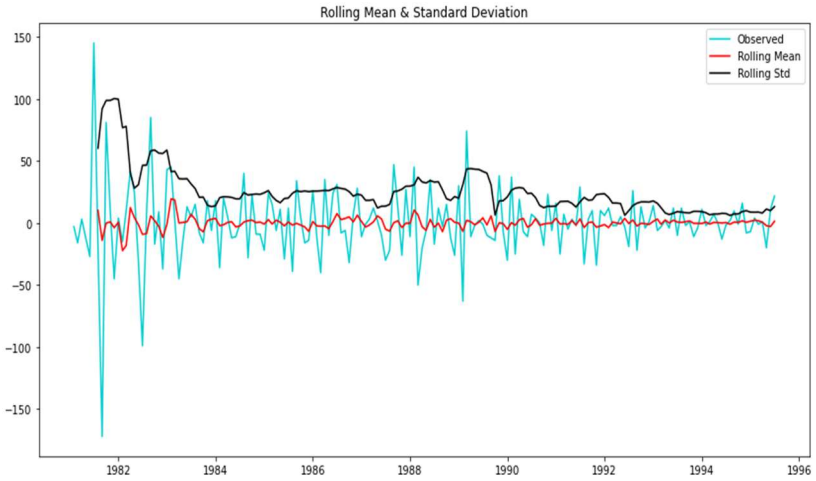



We see that there is both significant trend and seasonality. So, now we take a seasonal differencing and check the series.




**Seasoning Time Series with differencing**

An ADF test need to be done to check the stationarity after the above differencing. With a p-value below alpha 0.05 and test statistic below critical values, it can be confirmed that the data is stationary.

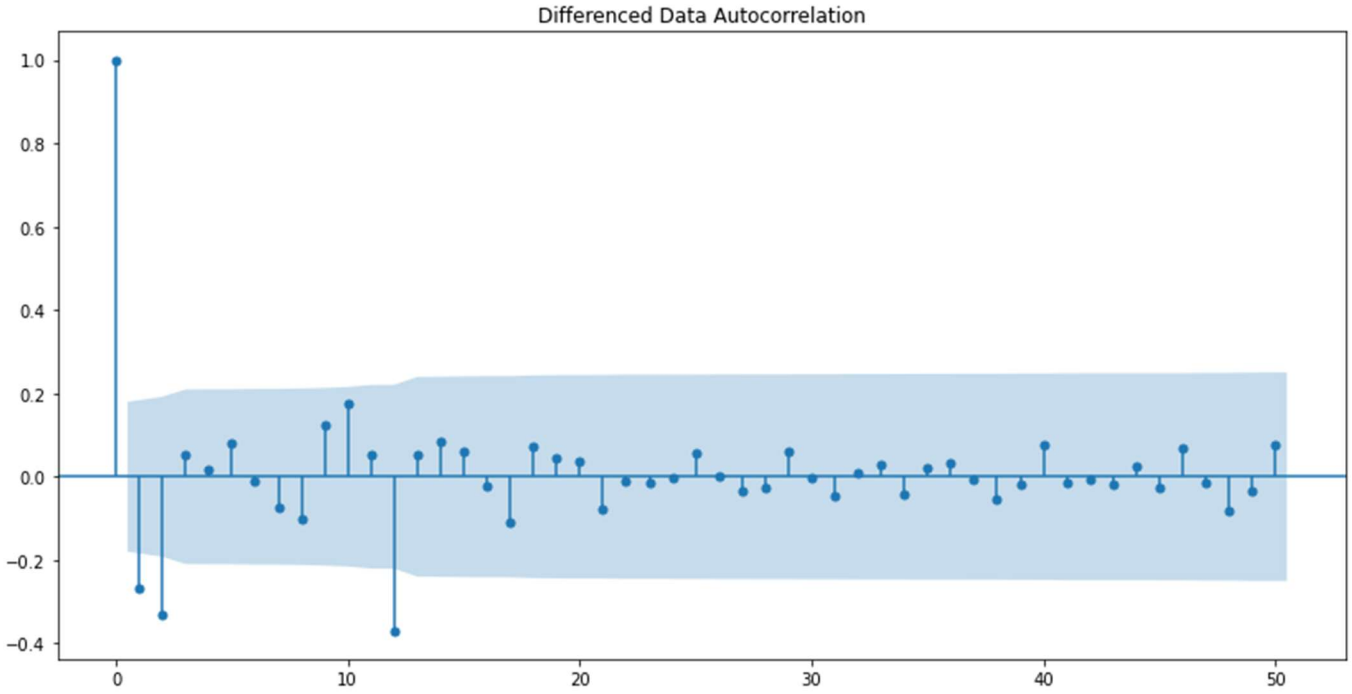


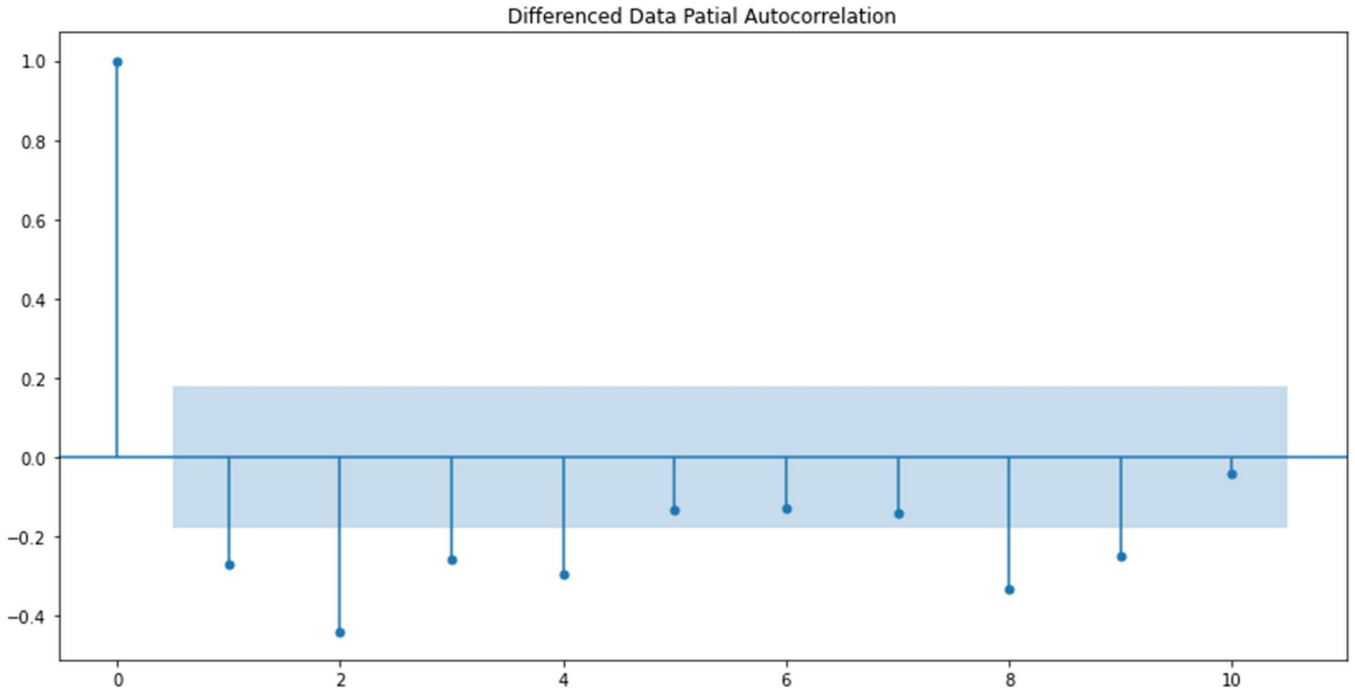

```
Results of Dickey-Fuller Test:
Test Statistic                  -4.605791
p-value                          0.000126
#Lags Used                      11.000000
Number of Observations Used    162.000000
Critical Value (1%)             -3.471374
Critical Value (5%)             -2.879552
Critical Value (10%)            -2.576373
dtype: float64
```

**ADF Test**

ACF and PACF plots of the seasonal-differenced + one order differenced data is created to find the values for (p,d,q)x(P,D,Q).

## ACF and PACF Plot :






**ACF and PACF Plot**

SARIMAX Results

| | | | |
|---|---|---|---|
| Dep. Variable: | y | No. Observations: | 132 |
| Model: | SARIMAX(4, 1, 2)x(0, 1, 2, 12) | Log Likelihood | -384.369 |
| Date: | Sat, 16 Jul 2022 | AIC | 786.737 |
| Time: | 09:40:34 | BIC | 809.433 |
| Sample: | 0 | HQIC | 795.898 |
| | - 132 | | |
| Covariance Type: | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ar.L1 | -0.8967 | 0.132 | -6.814 | 0.000 | -1.155 | -0.639 |
| ar.L2 | 0.0165 | 0.171 | 0.097 | 0.923 | -0.319 | 0.352 |
| ar.L3 | -0.1132 | 0.174 | -0.650 | 0.515 | -0.454 | 0.228 |
| ar.L4 | -0.1598 | 0.116 | -1.380 | 0.168 | -0.387 | 0.067 |
| ma.L1 | 0.1508 | 0.174 | 0.866 | 0.387 | -0.191 | 0.492 |
| ma.L2 | -0.8492 | 0.164 | -5.166 | 0.000 | -1.171 | -0.527 |
| ma.S.L12 | -0.3907 | 0.102 | -3.848 | 0.000 | -0.590 | -0.192 |
| ma.S.L24 | -0.0887 | 0.091 | -0.977 | 0.329 | -0.267 | 0.089 |
| sigma2 | 238.9649 | 0.001 | 2.02e+05 | 0.000 | 238.963 | 238.967 |

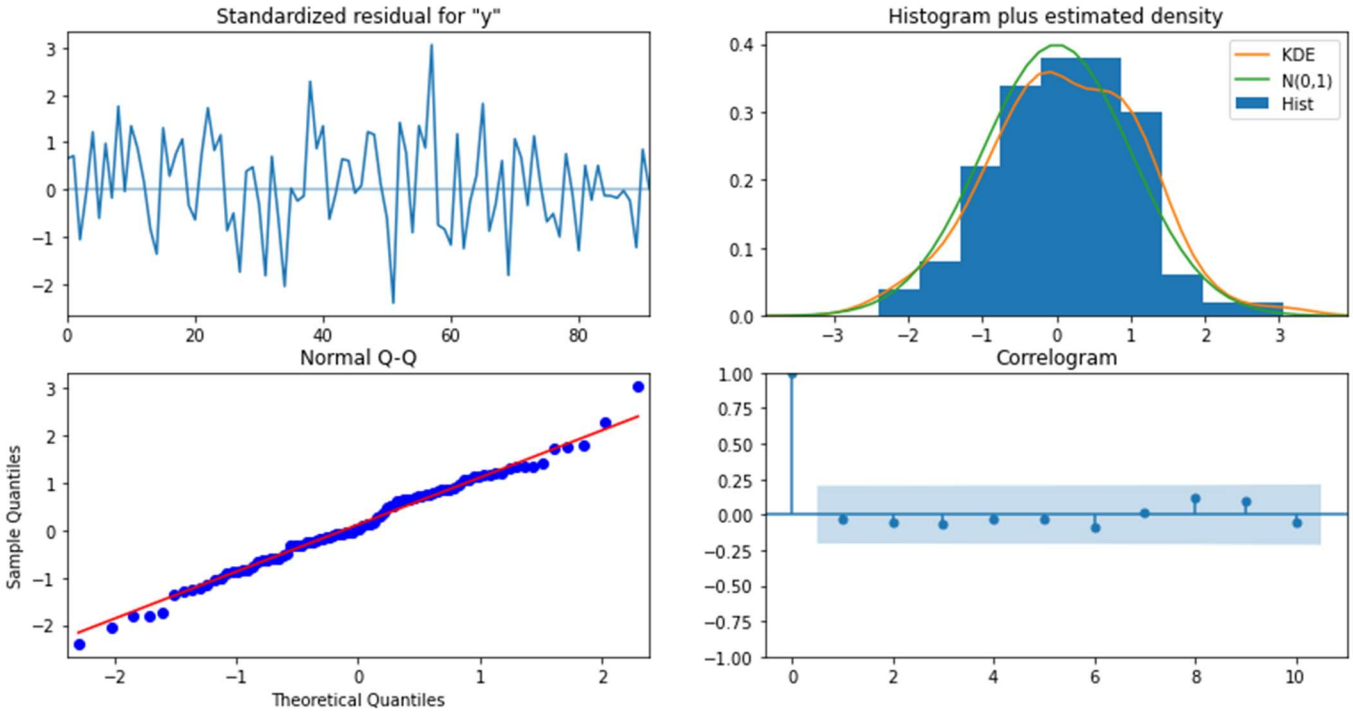
| | | | |
|---|---|---|---|
| Ljung-Box (L1) (Q): | 0.06 | Jarque-Bera (JB): | 0.01 |
| Prob(Q): | 0.80 | Prob(JB): | 0.99 |
| Heteroskedasticity (H): | 0.76 | Skew: | -0.01 |
| Prob(H) (two-sided): | 0.46 | Kurtosis: | 3.06 |

Figure 25: Model SARIMA Model Summary Result

## Observations:

- Here we have taken alpha = 0.05 and seasonal period as 12.
- From the PACF plot it can be seen that till 4th lag it's significant before cut-off, so AR term 'p = 4' is chosen. At seasonal lag of 12, seasonal AR 'P = 0'.
- From ACF plot it can be seen that till lag 2nd is significant before it cuts off, so MA term 'q =2' is selected and at seasonal lag of 12, a significant lag is apparent, so kept seasonal MA term 'Q = 1' initially.
- The seasonal MA term 'Q' was later optimized to 2, by validating model performance, as the data might be under-differenced.
- The final selected terms for SARIMA model are (4, 1, 2)*(0,1,2,12).
- The diagnostics plot of the model was derived and the standardized residuals are found to follow a mean of zero, and the histogram shows the residuals follow a normal distribution.
- The Normal Q-Q plot also shows that the quantiles come from a normal distribution as the point forms roughly a straight line.
- The correlogram shows the autocorrelation of the residuals and there are no significant lags above the confidence index.
- The RMSE values of the automated SARIMA model is 15.38.

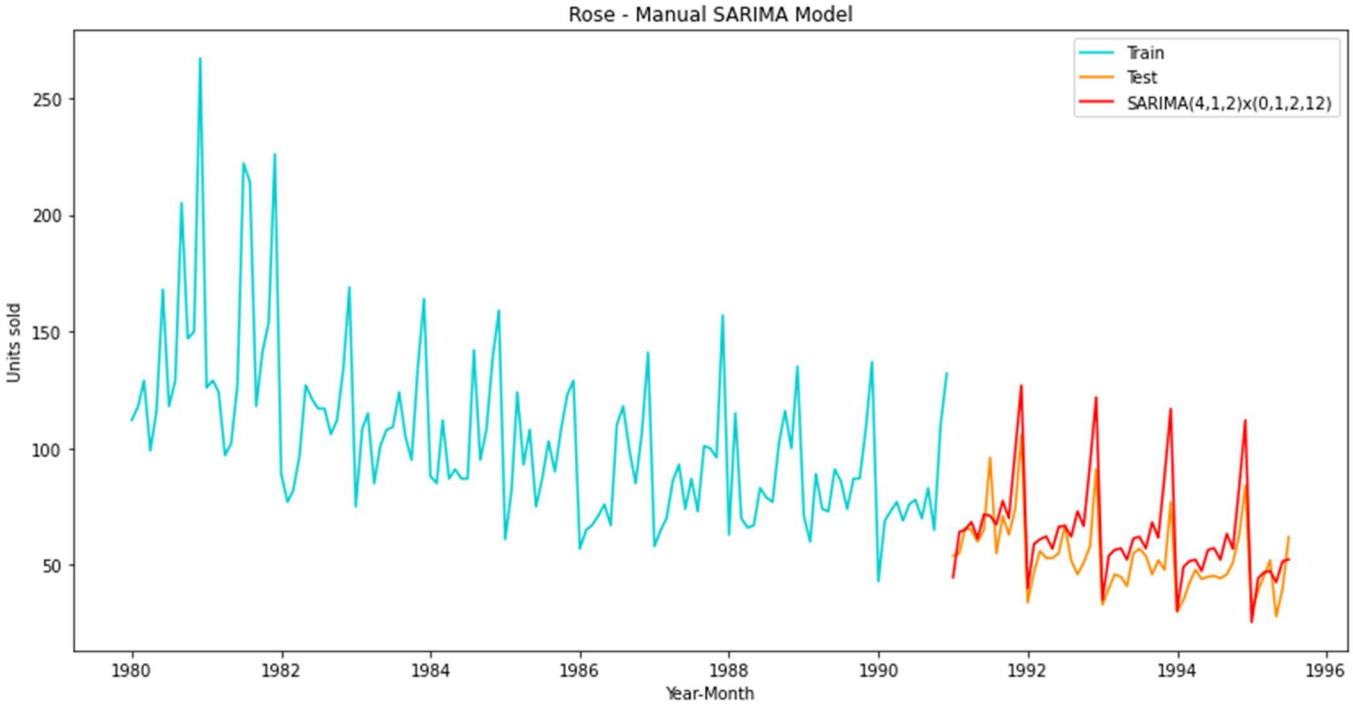
## Diagnostic Plot of Manual SARIMA Model



## Model Evaluation:

| YearMonth | Rose | rose_forecasted | rose_forecasted_log | manual_rose_forecasted |
|---|---|---|---|---|
| **1991-01-01** | 54.0 | 44.214702 | 56.850909 | 44.733041 |
| **1991-02-01** | 55.0 | 62.327974 | 66.337363 | 64.208694 |
| **1991-03-01** | 66.0 | 67.315152 | 70.531259 | 65.110690 |
| **1991-04-01** | 65.0 | 63.162471 | 66.371329 | 68.453063 |
| **1991-05-01** | 60.0 | 66.476733 | 69.031507 | 61.423433 |

**Manual SARIMA Forecasted Values**




**Plot Actual v/s Forecast Result on test data**

## RMSE Values:

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.278369 |
| NaiveModel | 79.745697 |
| SimpleAverage | 53.488233 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |
| Auto_SARIMA_log(0, 1, 1)*(1, 0, 1, 12) | 17.920074 |
| Manual_ARIMA(0,1,0) | 84.160493 |
| Manual_SARIMA(4, 1, 2)*(0, 1, 2, 12) | 15.388806 |

### 1.8 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

### Solution:

| | TEST RSME |
|---|---|
| RegressionOnTime | **15.28** |
| NaiveModel | **79.75** |
| SimpleAverage | **53.49** |
| 2-point TMA | **11.53** |
| 4-point TMA | **14.46** |
| 6-point TMA | **14.57** |
| 9-point TMA | **14.73** |
| Alpha=0.0987, SES Optimized | **36.82** |
| Alpha=0.10,SES_Iterative | **36.86** |
| Alpha=0.0,Beta=0.0, DES Optimized | **15.72** |
| Alpha=0.1,Beta=0.1,DES_Iterative | **36.95** |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | **21.06** |
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | **9.94** |
| Auto_ARIMA (0, 1, 2) | **15.63** |
| Auto_SARIMA (0, 1, 2)*(2, 1, 2, 12) | **16.53** |
| Auto_SARIMA_log (0, 1, 1)*(1, 0, 1, 12) | **17.92** |
| Manual_ARIMA (0,1,0) | **84.16** |
| Manual_SARIMA (4, 1, 2)*(0, 1, 2, 12) | **15.39** |

Triple Exponential Smoothing (Holt Winter's) with alpha: 0.1, beta: 0.2 and gamma: 0.3 is found to be the best model, followed by 2-point trailing moving average model.

### 1.9 Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.

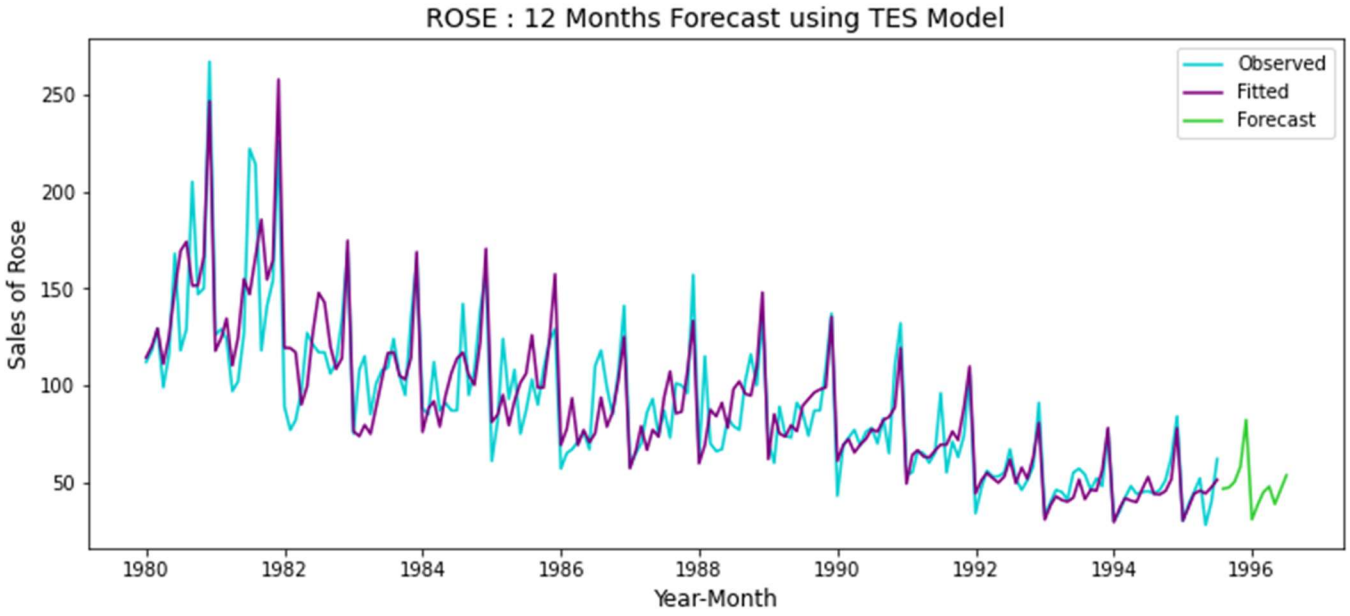
**Solution:**

**Observations:**

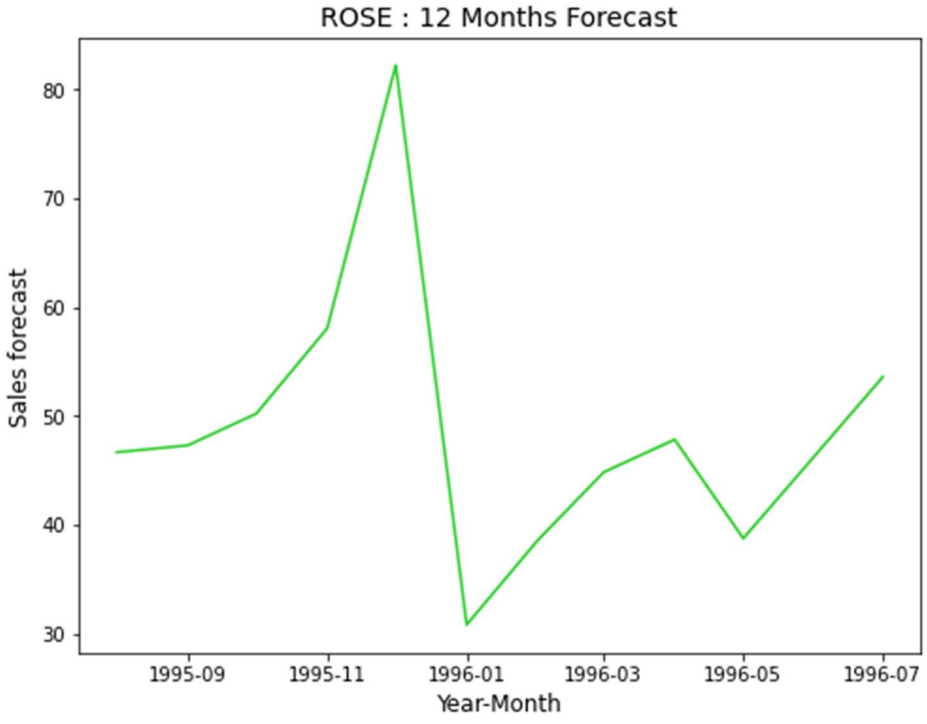
- Based on the overall model evaluation and comparison, Triple Exponential Smoothing (Holt Winter's) is selected for final prediction into 12 months in future.
- TES model alpha: 0.1, beta: 0.2 and gamma: 0.3 & trend: 'additive', seasonal: 'multiplicative' is found to be the best model in terms of accuracy scored against the full data.
- The model predicts continuation of the trend in sales and seasonality in year-end sales. The prediction shows a stabilization of downward trend, as the sales will be almost same as previous observed year.
- The RMSE value of TES obtained for the entire dataset is 17.88

| | Test RMSE |
|---|---|
| Alpha=0.1,Beta=0.2,gamma=0.3,TES_Iterative | 9.943563 |
| 2 point TMA | 11.530054 |
| 4 point TMA | 14.458402 |
| 6 point TMA | 14.572976 |
| 9 point TMA | 14.732918 |
| RegressionOnTime | 15.278369 |
| Manual_SARIMA(4, 1, 2)*(0, 1, 2, 12) | 15.388806 |
| Auto_ARIMA(0, 1, 2) | 15.627280 |
| Alpha=0.0,Beta=0.0, DES Optimized | 15.718202 |
| Auto_SARIMA(0, 1, 2)*(2, 1, 2, 12) | 16.529473 |
| Auto_SARIMA_log(0, 1, 1)*(1, 0, 1, 12) | 17.920074 |
| Alpha=0.065,Beta=0.054,gamma=0.0 TES Optimized | 21.056902 |
| Alpha=0.0987, SES Optimized | 36.824464 |
| Alpha=0.10,SES_Iterative | 36.856268 |
| Alpha=0.1,Beta=0.1,DES_Iterative | 36.950000 |
| SimpleAverage | 53.488233 |
| NaiveModel | 79.745697 |
| SERIES_ROSE DATASET.ipynb Manual_ARIMA(0,1,0) | 84.160493 |

## Actual Plot and Future Forecast Result




## Future Forecast Result



## 1.10 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

**Solution:**

```
1995-08-01    46.645790
1995-09-01    47.277864
1995-10-01    50.192393
1995-11-01    58.032965
1995-12-01    82.211766
1996-01-01    30.793144
1996-02-01    38.536058
1996-03-01    44.822234
1996-04-01    47.814473
1996-05-01    38.727986
1996-06-01    46.255070
1996-07-01    53.559025
Freq: MS, dtype: float64
```

**Future Forecast Result and summary statistics**

**Descriptive Statistics:**

```
count    12.000000
mean     48.739064
std      12.747211
min      30.793144
25%      43.298672
50%      46.961827
75%      51.034051
max      82.211766
dtype: float64
```

### Inferences:

- The model forecasts sale of 585 units of Rose wine in 12 months into future. Which is an average sale of 48 units per month.
- The seasonal sale in December 1995 will reach a maximum of 82 units, before it drops to the lowest sale in January 1996; at 30 units.
- Unlike Sparkling wine, Rose wine sells very low number of units and the standard deviation is only 12.75. Which means that higher demand does not impact procurement and production.
- The ABC estate wine should investigate the low demand for Rose wine in market and make corrective actions in marketing and promotions.