# Time Series Forecasting Project :

-Prapthi Pandian

## Table of Contents

## 1. Problem Statement-

The data of different types of wine sales in the 20th century is to be analyzed. 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 analyze and forecast Wine Sales in the 20th century.

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

**Printing the head of Sparkling data series-**

| | YearMonth | Sparkling |
|---|---|---|
| **0** | 1980-01 | 1686 |
| **1** | 1980-02 | 1591 |
| **2** | 1980-03 | 2304 |
| **3** | 1980-04 | 1712 |
| **4** | 1980-05 | 1471 |

**Creating the Time Stamps and adding to the data frame to make it a Time Series Data-**

```
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')
```

**Dataframe post setting the index-**

| | Sparkling |
|---|---|
| **Time_Stamp** | |
| **1980-01-31** | 1686 |
| **1980-02-29** | 1591 |
| **1980-03-31** | 2304 |
| **1980-04-30** | 1712 |
| **1980-05-31** | 1471 |

**Shape-**

```
(187, 1)
```

**Summary-**

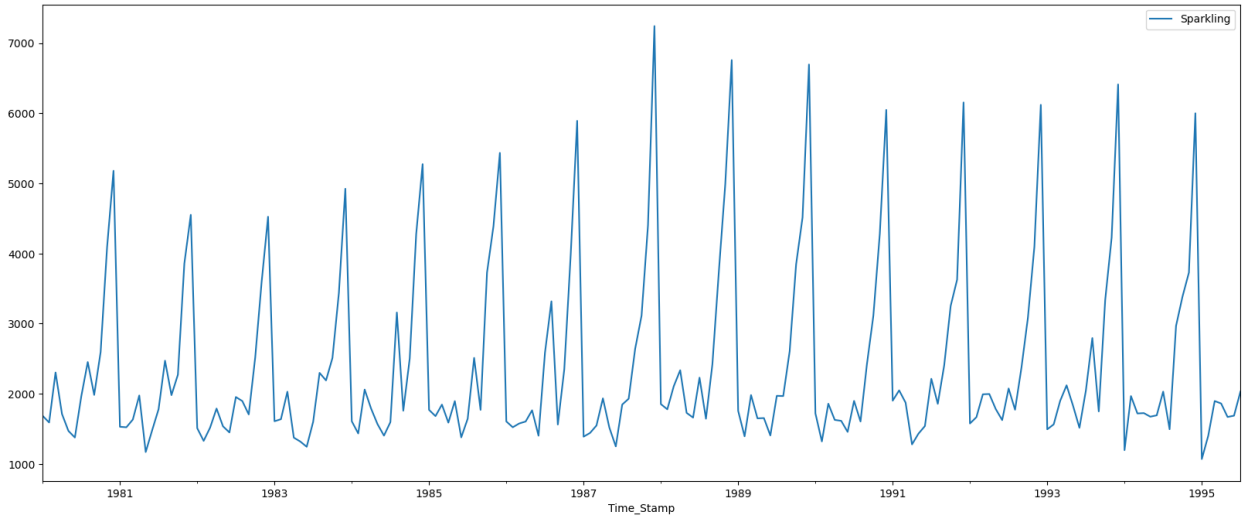
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Sparkling  187 non-null    int64
dtypes: int64(1)
memory usage: 2.9 KB
```

**Null values:**

```
Sparkling    0
dtype: int64
```

- The Dataframe has a total of 187 entries with no missing entries.
- It represents a time series dataset of 'Sparkling' wine sales from January 1980 to July 1995
- The data type of the 'Sparkling' column is integer.
- There are no null values present in the dataset.

**Plotting the Time Series to understand the behaviour of the data-**




- We can see that there is a seasonal pattern associated with it.

**2.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.**

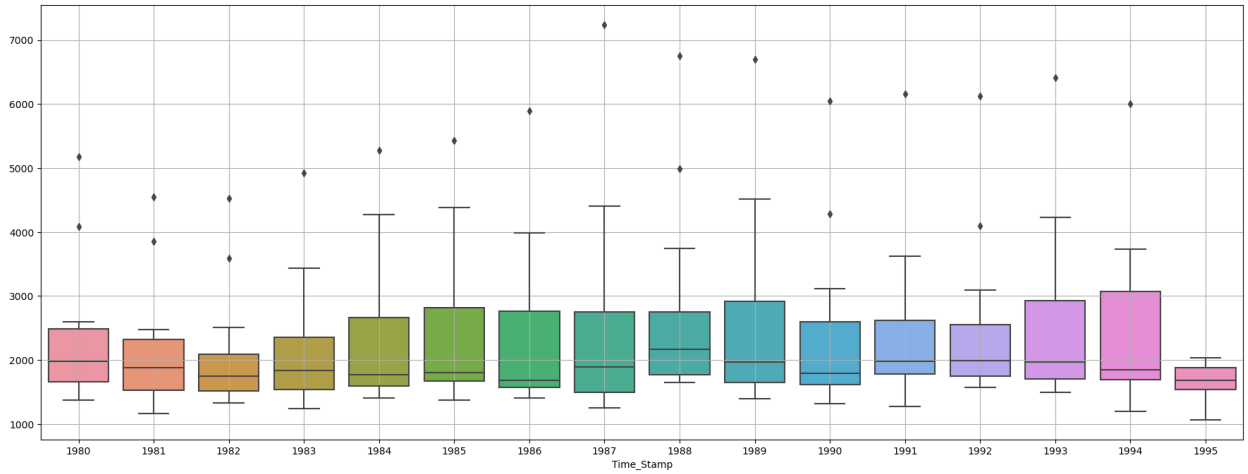
**Description-**

| | Sparkling |
|---|---|
| **count** | 187.000 |
| **mean** | 2402.417 |
| **std** | 1295.112 |
| **min** | 1070.000 |
| **25%** | 1605.000 |
| **50%** | 1874.000 |
| **75%** | 2549.000 |
| **max** | 7242.000 |

- The average value of the 'Sparkling' sales is 2402.417
- The minimum value in the 'Sparkling' sales is 1070.000 and the maximum value is 7242.000
- The first quartile, or 25th percentile indicates the value below which 25% of the data falls i.e. 1605.000 in this case.
- The median, or 50th percentile represents the middle value of the 'Sparkling' which is 1874.000
- The third quartile, or 75th percentile indicates the value below which 75% of the data falls i.e. 2549.000
- The maximum value of 7242.000 is substantially higher than the mean and median. This indicates the presence of outliers.
- The mean value (2402.417) is greater than the median value (1874.000). Hence, there is a tendency for the distribution to be right skewed.

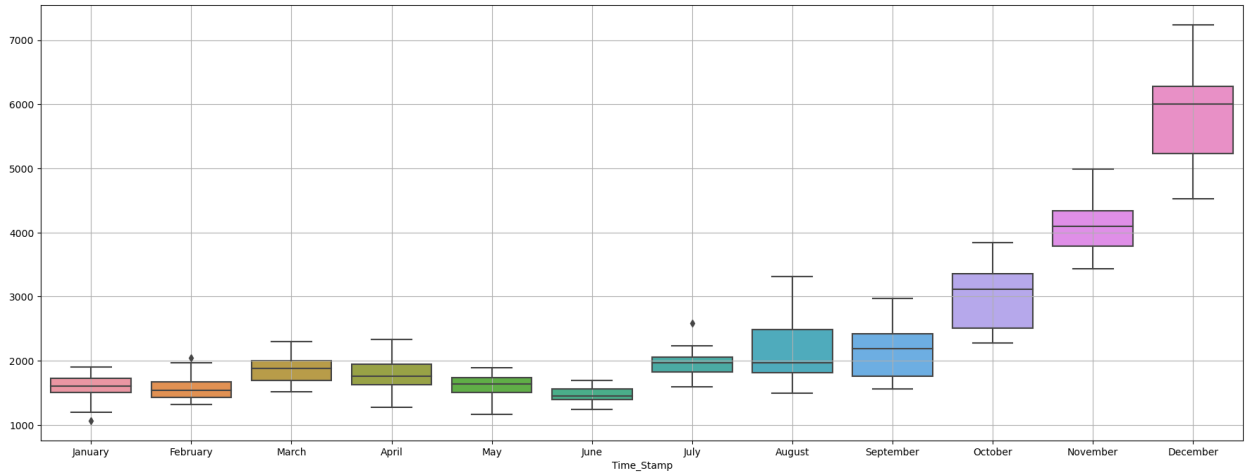
**Boxplot to understand the spread of sales across different years and within different months across years**

**Yearly boxplot-**

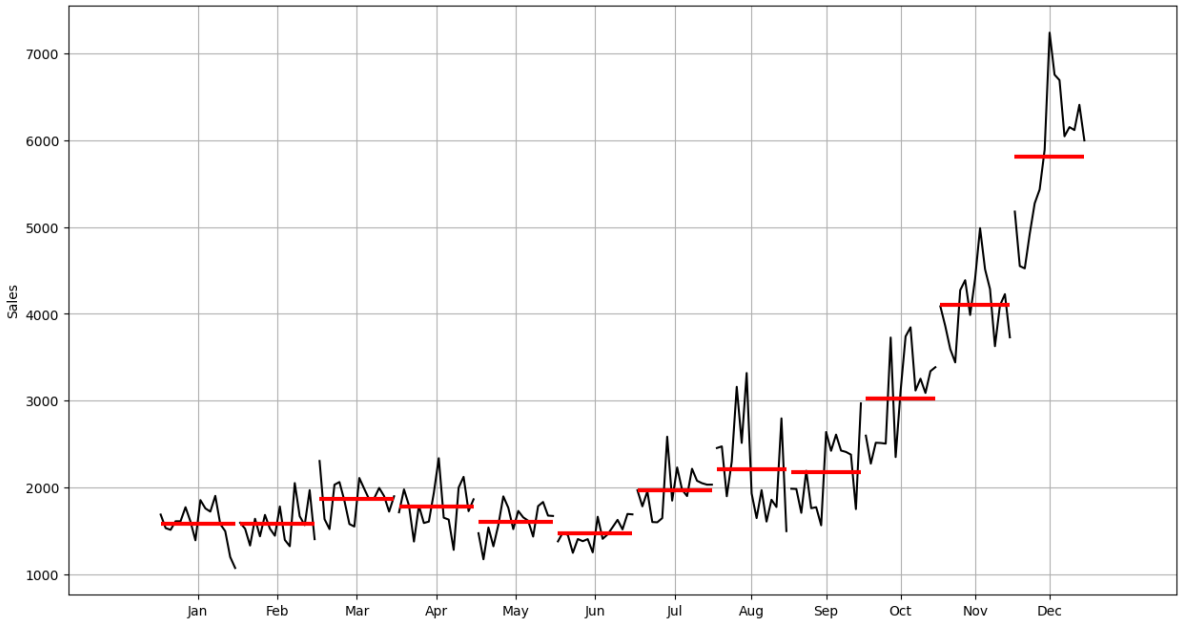



- The outliers represent that almost all years have been inconsistent except for 1995.
- Hence, there has been a seasonal pattern occurring every year causing the irregularities in data.

**Monthly boxplot-**



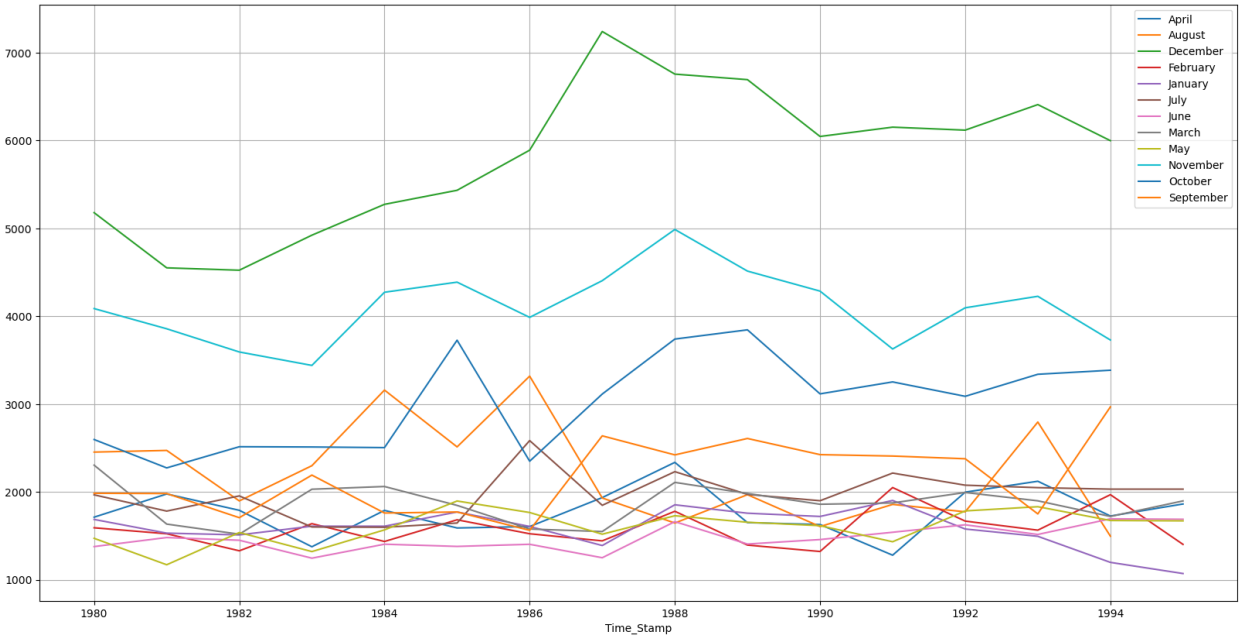

- The heaviest season has been December while the slightest season has been January, February.
- Season where we need to be prepared for higher fluctuations or uncertainties is July.



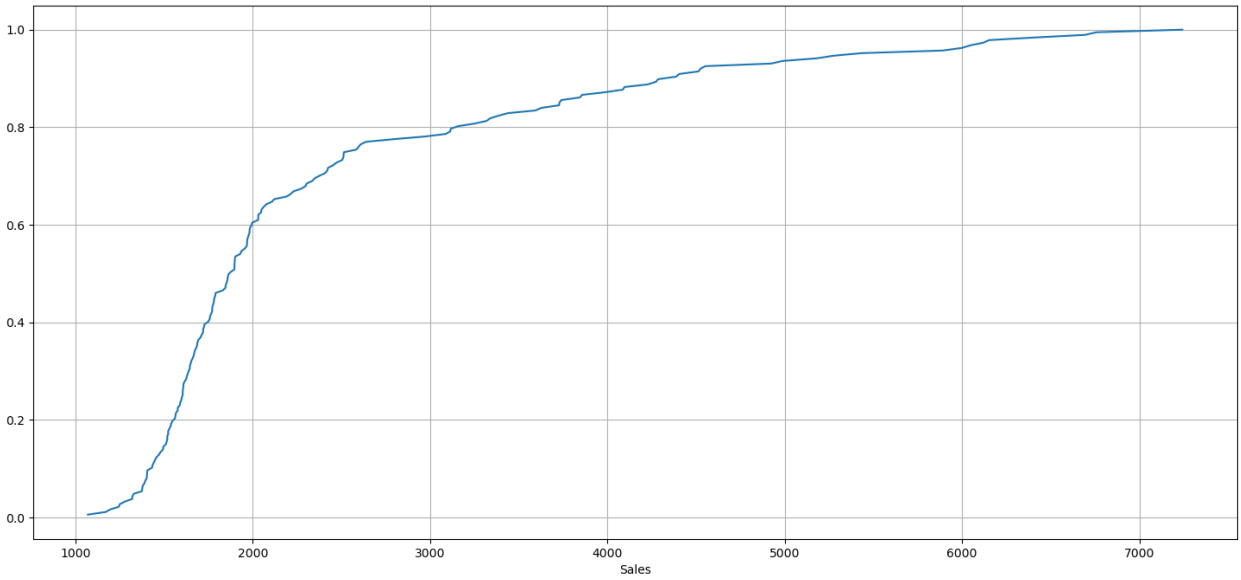

- This plot shows us the behaviour of the Time Series across various months. The red line is the median value.

**Plot of monthly sales across years-**

| Time_Stamp | April | August | December | February | January | July | June | March | May | November | October | September |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Time_Stamp** | | | | | | | | | | | | |
| **1980** | 1712.0 | 2453.0 | 5179.0 | 1591.0 | 1686.0 | 1966.0 | 1377.0 | 2304.0 | 1471.0 | 4087.0 | 2596.0 | 1984.0 |
| **1981** | 1976.0 | 2472.0 | 4551.0 | 1523.0 | 1530.0 | 1781.0 | 1480.0 | 1633.0 | 1170.0 | 3857.0 | 2273.0 | 1981.0 |
| **1982** | 1790.0 | 1897.0 | 4524.0 | 1329.0 | 1510.0 | 1954.0 | 1449.0 | 1518.0 | 1537.0 | 3593.0 | 2514.0 | 1706.0 |
| **1983** | 1375.0 | 2298.0 | 4923.0 | 1638.0 | 1609.0 | 1600.0 | 1245.0 | 2030.0 | 1320.0 | 3440.0 | 2511.0 | 2191.0 |
| **1984** | 1789.0 | 3159.0 | 5274.0 | 1435.0 | 1609.0 | 1597.0 | 1404.0 | 2061.0 | 1567.0 | 4273.0 | 2504.0 | 1759.0 |
| **1985** | 1589.0 | 2512.0 | 5434.0 | 1682.0 | 1771.0 | 1645.0 | 1379.0 | 1846.0 | 1896.0 | 4388.0 | 3727.0 | 1771.0 |
| **1986** | 1605.0 | 3318.0 | 5891.0 | 1523.0 | 1606.0 | 2584.0 | 1403.0 | 1577.0 | 1765.0 | 3987.0 | 2349.0 | 1562.0 |
| **1987** | 1935.0 | 1930.0 | 7242.0 | 1442.0 | 1389.0 | 1847.0 | 1250.0 | 1548.0 | 1518.0 | 4405.0 | 3114.0 | 2638.0 |
| **1988** | 2336.0 | 1645.0 | 6757.0 | 1779.0 | 1853.0 | 2230.0 | 1661.0 | 2108.0 | 1728.0 | 4988.0 | 3740.0 | 2421.0 |
| **1989** | 1650.0 | 1968.0 | 6694.0 | 1394.0 | 1757.0 | 1971.0 | 1406.0 | 1982.0 | 1654.0 | 4514.0 | 3845.0 | 2608.0 |
| **1990** | 1628.0 | 1605.0 | 6047.0 | 1321.0 | 1720.0 | 1899.0 | 1457.0 | 1859.0 | 1615.0 | 4286.0 | 3116.0 | 2424.0 |
| **1991** | 1279.0 | 1857.0 | 6153.0 | 2049.0 | 1902.0 | 2214.0 | 1540.0 | 1874.0 | 1432.0 | 3627.0 | 3252.0 | 2408.0 |
| **1992** | 1997.0 | 1773.0 | 6119.0 | 1667.0 | 1577.0 | 2076.0 | 1625.0 | 1993.0 | 1783.0 | 4096.0 | 3088.0 | 2377.0 |
| **1993** | 2121.0 | 2795.0 | 6410.0 | 1564.0 | 1494.0 | 2048.0 | 1515.0 | 1898.0 | 1831.0 | 4227.0 | 3339.0 | 1749.0 |
| **1994** | 1725.0 | 1495.0 | 5999.0 | 1968.0 | 1197.0 | 2031.0 | 1693.0 | 1720.0 | 1674.0 | 3729.0 | 3385.0 | 2968.0 |
| **1995** | 1862.0 | NaN | NaN | 1402.0 | 1070.0 | 2031.0 | 1688.0 | 1897.0 | 1670.0 | NaN | NaN | NaN |

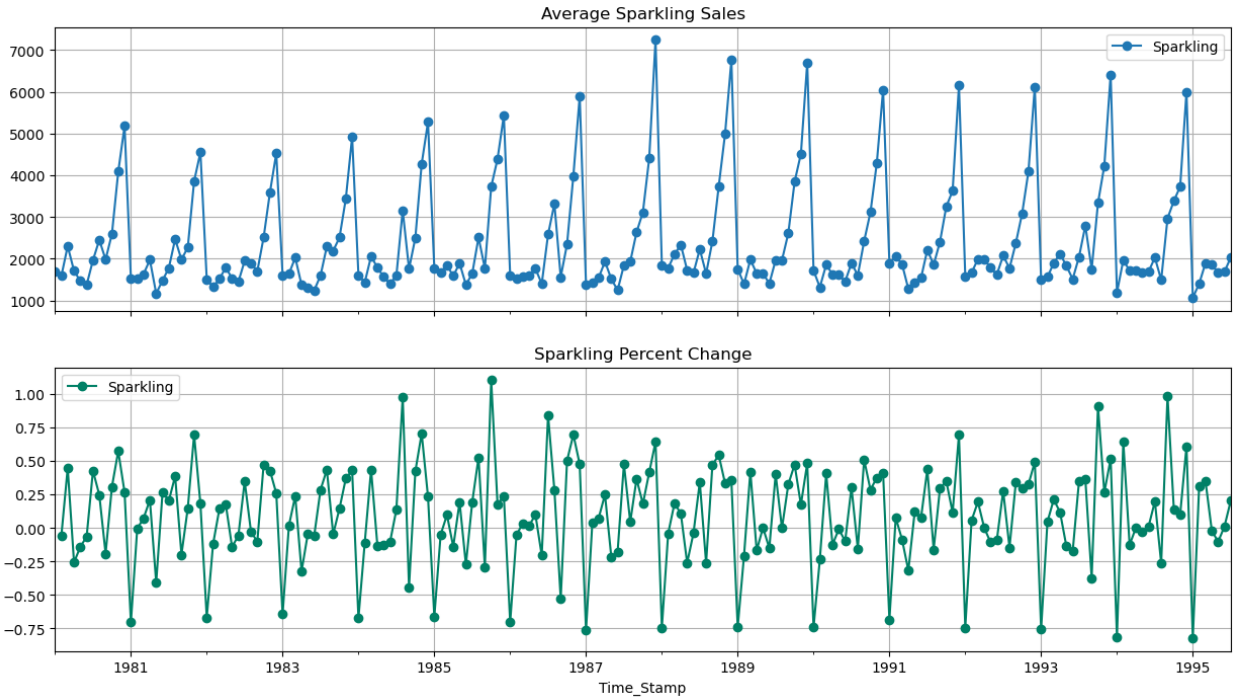



**Empirical Cumulative Distribution-**




- This particular graph tells us what percentage of data points refer to what number of Sparkling Sales.
- For instance, the sale between 2000 and 3000 units is 78%- 60% i.e. 18%
- Sale between 3000 and 4000 units is 83%- 78% i.e. 5%

**Average Sparkling Sales per month and the month on month percentage change of Sales-**

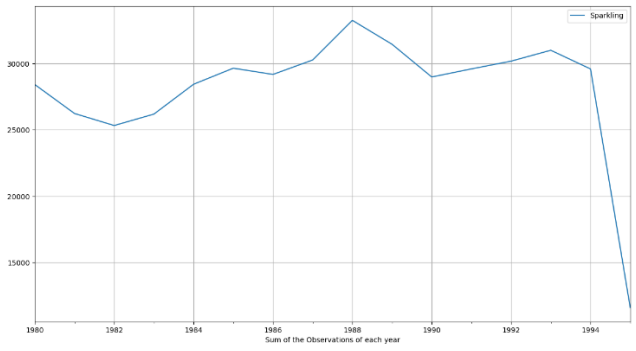



- The above two graphs tells us the Average 'Sparkling Sales' and the Percentage change of 'Sparkling Sales' with respect to the time.

**Reading this monthly data into a quarterly and yearly format-**

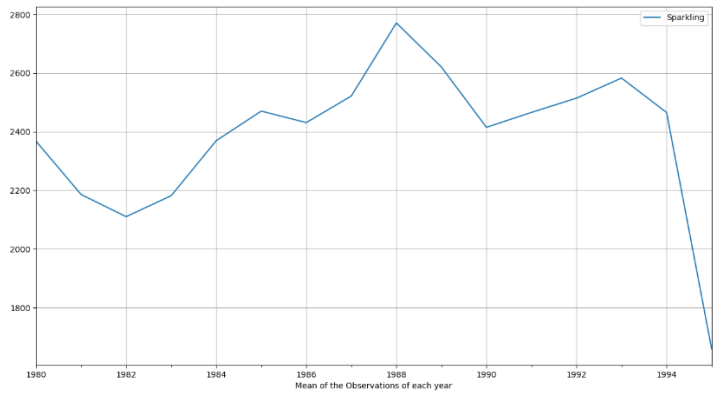
**Yearly Plot-**

Let us try to resample or aggregate the Time Series from an annual perspective

| | Head of dataframe | Plot |
|---|---|---|
| **Sum of observations of each month** | Sparkling<br>Time_Stamp<br>1980-12-31 28406<br>1981-12-31 26227<br>1982-12-31 25321<br>1983-12-31 26180<br>1984-12-31 28431 |  |

| | | |
|---|---|---|
| **Mean of the observations of the year considering each month as an observation** | | |

| Time_Stamp | Sparkling |
|---|---|
| 1980-12-31 | 2367.17 |
| 1981-12-31 | 2185.58 |
| 1982-12-31 | 2110.08 |
| 1983-12-31 | 2181.67 |
| 1984-12-31 | 2369.25 |

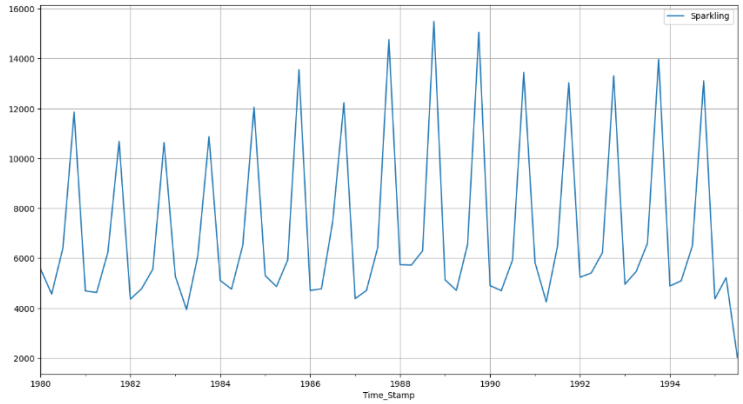



- There is an increasing trend in the sum of 'Sparkling' sales over the years, reaching a peak in 1988 and then with some fluctuations, there is a significant drop in sales in 1995.

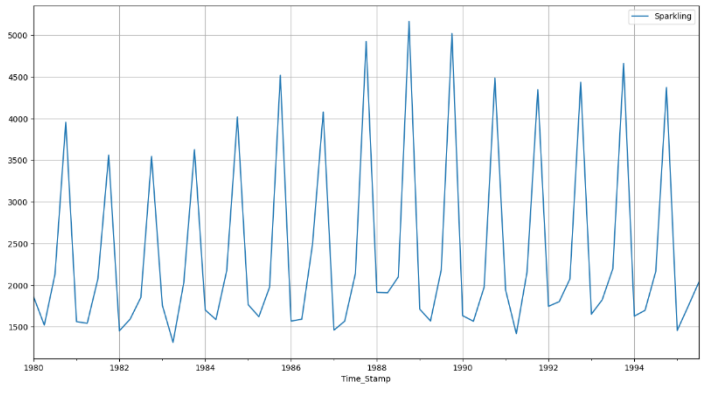
**Quarterly plot-**

| | Head of dataframe | Plot |
|---|---|---|
| **Sum of observations** | | |
| **Mean of observations** | | |

| Time_Stamp | Sparkling |
|---|---|
| 1980-03-31 | 5581 |
| 1980-06-30 | 4560 |
| 1980-09-30 | 6403 |
| 1980-12-31 | 11862 |
| 1981-03-31 | 4686 |




| Time_Stamp | Sparkling |
|---|---|
| 1980-03-31 | 1860.333333 |
| 1980-06-30 | 1520.000000 |
| 1980-09-30 | 2134.333333 |
| 1980-12-31 | 3954.000000 |
| 1981-03-31 | 1562.000000 |



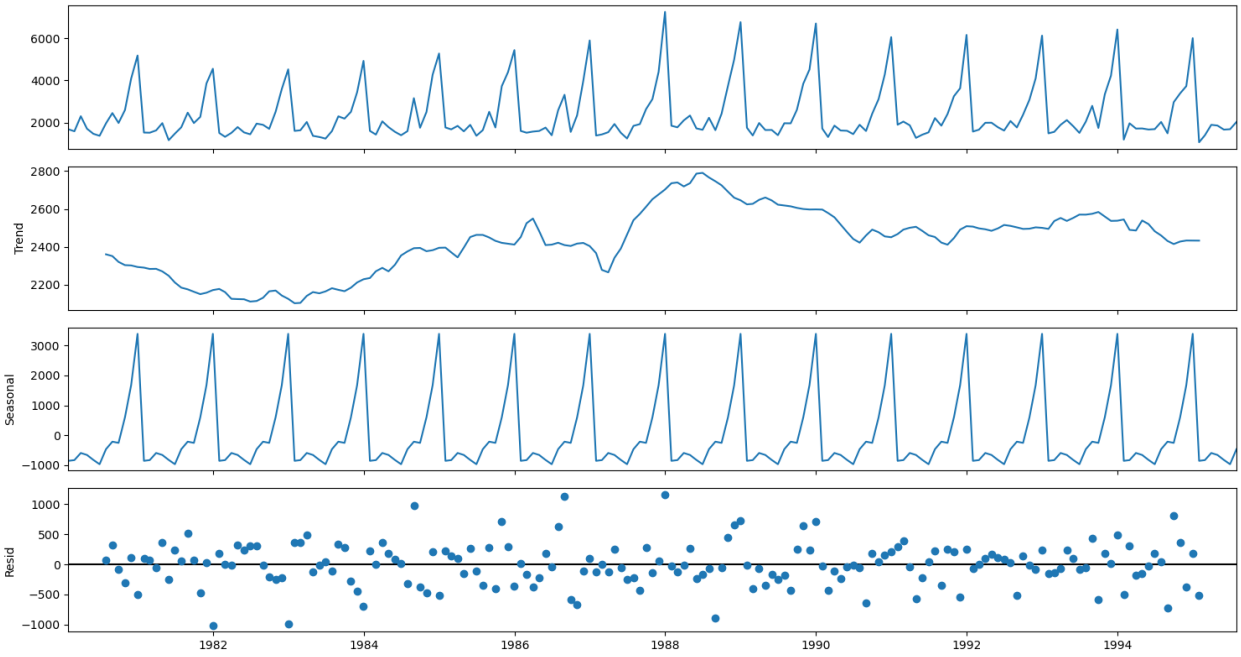
- We can notice high peak in sales in the end of second quarter and beginning of 3rd quarter.
- And a significant reduction in sales in the last quarter.

**Decomposition-**

- Decomposing the series into systematic component and irregular component
- **Systematic components**- Trend, seasonality which are interpretable and can be estimated
- **Irregular component-** error / noise associated with the series.
- Compares long term movement of series (Trend) and short-term movement (seasonality) to understand which has higher influence

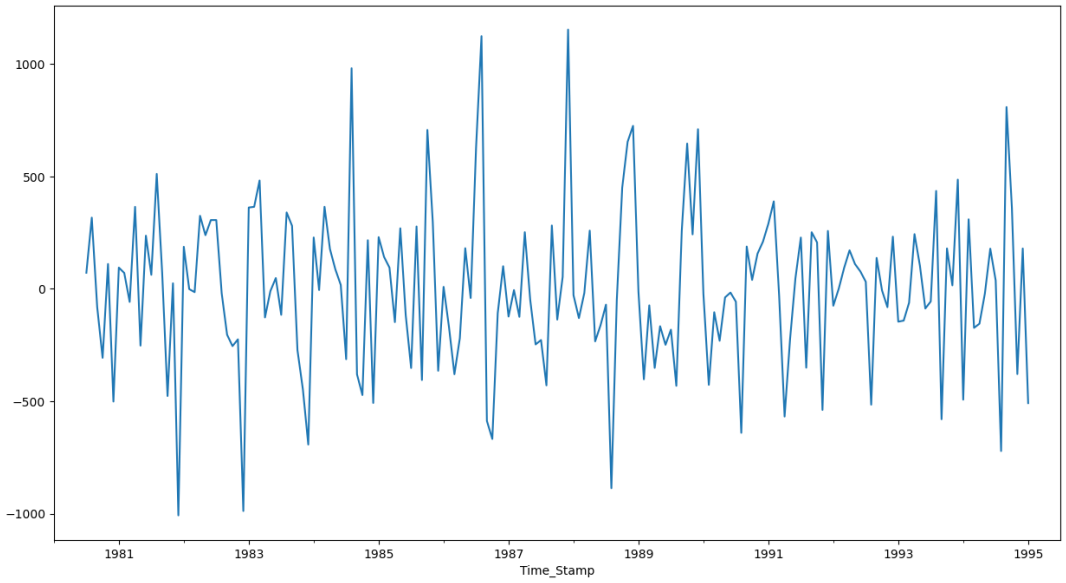
1. **Additive model-**

   When seasonal variation is constant over time.



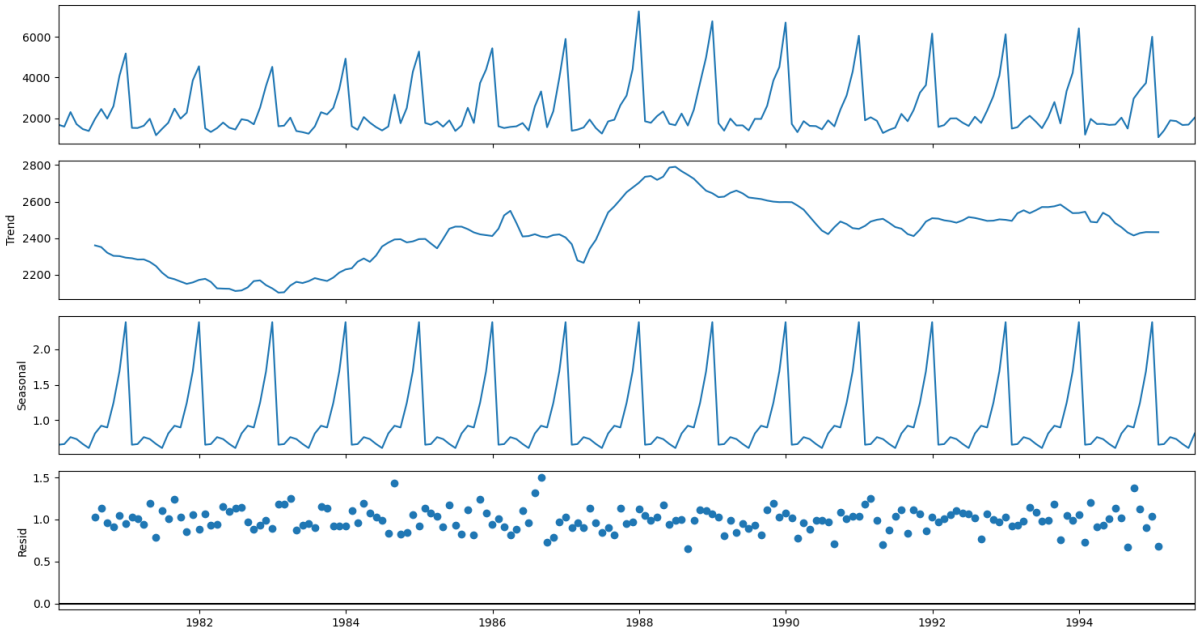
```
Trend
 Time_Stamp
1980-01-31        NaN
1980-02-29        NaN
1980-03-31        NaN
1980-04-30        NaN
1980-05-31        NaN
1980-06-30        NaN
1980-07-31    2360.67
1980-08-31    2351.33
1980-09-30    2320.54
1980-10-31    2303.58
1980-11-30    2302.04
1980-12-31    2293.79
Name: trend, dtype: float64


Seasonality
 Time_Stamp
1980-01-31    -854.26
1980-02-29    -830.35
1980-03-31    -592.36
1980-04-30    -658.49
1980-05-31    -824.42
1980-06-30    -967.43
1980-07-31    -465.50
1980-08-31    -214.33
1980-09-30    -254.68
1980-10-31     599.77
1980-11-30    1675.07
1980-12-31    3386.98
Name: seasonal, dtype: float64


Residual
 Time_Stamp
1980-01-31       NaN
1980-02-29       NaN
1980-03-31       NaN
1980-04-30       NaN
1980-05-31       NaN
1980-06-30       NaN
1980-07-31     70.84
1980-08-31    316.00
1980-09-30    -81.86
1980-10-31   -307.35
1980-11-30    109.89
1980-12-31   -501.78
Name: resid, dtype: float64
```

**Residual Plot-**



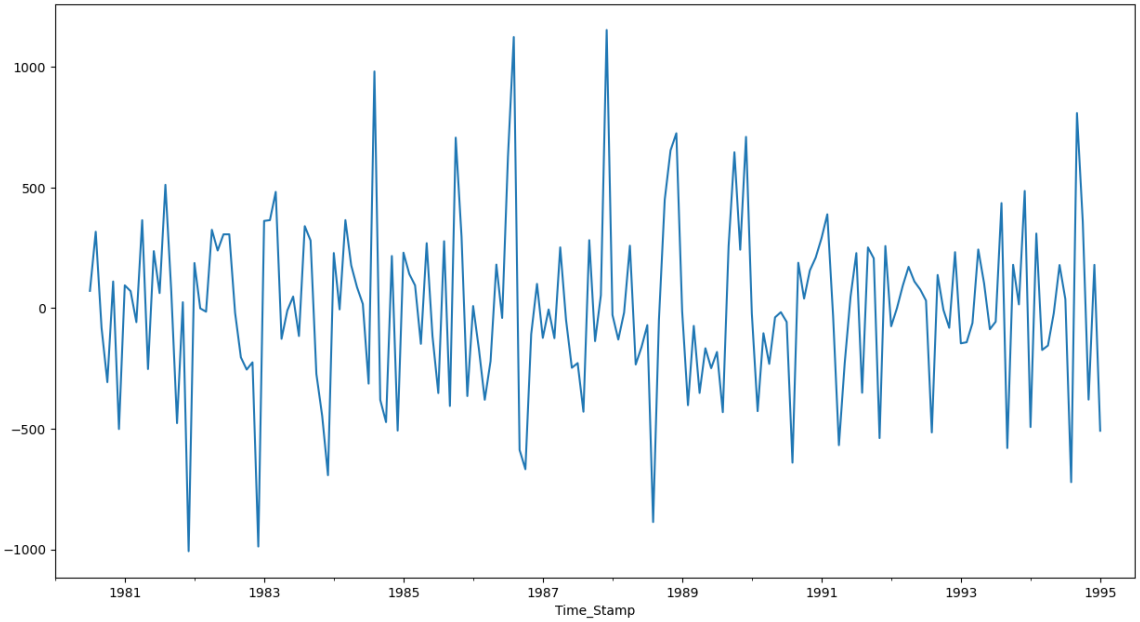

## 2. Decompose the time series multiplicatively-

When there is seasonal variation with time.



```
Trend
 Time_Stamp
1980-01-31        NaN
1980-02-29        NaN
1980-03-31        NaN
1980-04-30        NaN
1980-05-31        NaN
1980-06-30        NaN
1980-07-31    2360.67
1980-08-31    2351.33
1980-09-30    2320.54
1980-10-31    2303.58
1980-11-30    2302.04
1980-12-31    2293.79
Name: trend, dtype: float64


 Seasonality
  Time_Stamp
 1980-01-31    0.65
 1980-02-29    0.66
 1980-03-31    0.76
 1980-04-30    0.73
 1980-05-31    0.66
 1980-06-30    0.60
 1980-07-31    0.81
 1980-08-31    0.92
 1980-09-30    0.89
 1980-10-31    1.24
 1980-11-30    1.69
 1980-12-31    2.38
 Name: seasonal, dtype: float64


 Residual
  Time_Stamp
 1980-01-31     NaN
 1980-02-29     NaN
 1980-03-31     NaN
 1980-04-30     NaN
 1980-05-31     NaN
 1980-06-30     NaN
 1980-07-31    1.03
 1980-08-31    1.14
 1980-09-30    0.96
 1980-10-31    0.91
 1980-11-30    1.05
 1980-12-31    0.95
 Name: resid, dtype: float64
```

**Residual Plot-**




- On comparing the residual plots, we can observe that the error pattern looks the same. Hence, it seems like an additive model.

**2.3 Split the data into training and test. The test data should start in 1991.**

Training Data

| Time_Stamp | Sparkling |
|---|---|
| **1980-01-31** | 1686 |
| **1980-02-29** | 1591 |
| **1980-03-31** | 2304 |
| **1980-04-30** | 1712 |
| **1980-05-31** | 1471 |
| **...** | ... |
| **1990-08-31** | 1605 |
| **1990-09-30** | 2424 |
| **1990-10-31** | 3116 |
| **1990-11-30** | 4286 |
| **1990-12-31** | 6047 |

132 rows × 1 columns

Head of Test Data

| Time_Stamp | Sparkling |
|---|---|
| 1991-01-31 | 1902 |
| 1991-02-28 | 2049 |
| 1991-03-31 | 1874 |
| 1991-04-30 | 1279 |
| 1991-05-31 | 1432 |

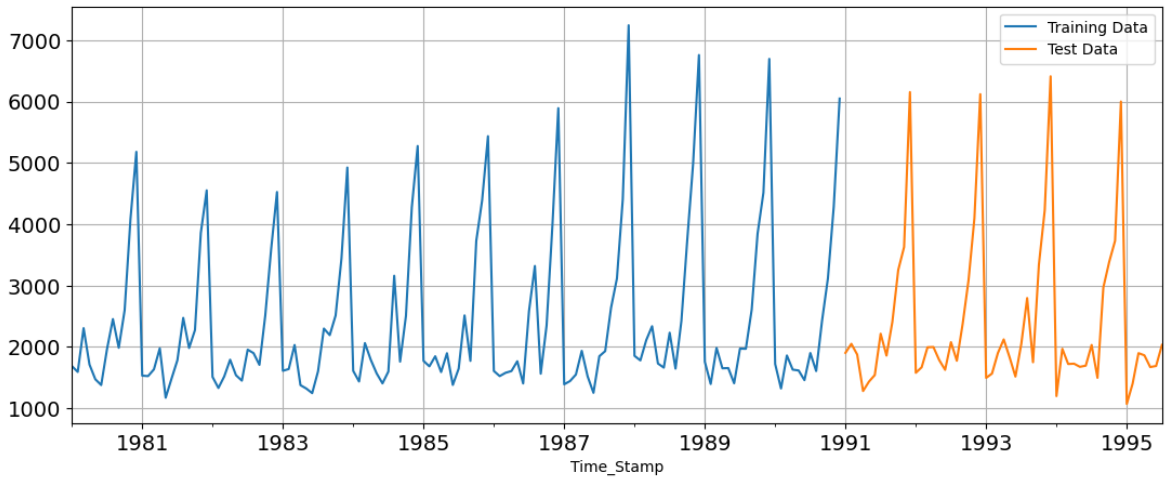
Tail of Test Data

| Time_Stamp | Sparkling |
|---|---|
| 1995-03-31 | 1897 |
| 1995-04-30 | 1862 |
| 1995-05-31 | 1670 |
| 1995-06-30 | 1688 |
| 1995-07-31 | 2031 |

**Shape:**

```
Train dataset shape: (132, 1)
Test dataset shape: (55, 1)
```

**Plot:**



**2.4 Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE.**

**RMSE (Root Mean Squared Error)**

- It is a commonly used metric to evaluate the accuracy of a time series model.
- It measures the average magnitude of the errors between predicted and observed values.
- For time series models, RMSE is used to assess how well the model forecasts future values over time.
- A lower RMSE value indicates better model performance, as it denotes that the model's predictions are relatively closer to the actual values.

We are going to build models on the training data and evaluate their performance on test data based on the RMSE values.

**Model 1: Linear Regression**

Not a time series model, but we are breaking it into time series specific.
Train is from 1st - 132nd value
Test is from 133rd value.

```
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, 3
4, 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, 121, 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, 157,
158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187]
```

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.

**Train data-**

```
First few rows of Training Data
             Sparkling  time
Time_Stamp
1980-01-31       1686     1
1980-02-29       1591     2
1980-03-31       2304     3
1980-04-30       1712     4
1980-05-31       1471     5

Last few rows of Training Data
             Sparkling  time
Time_Stamp
1990-08-31       1605   128
1990-09-30       2424   129
1990-10-31       3116   130
1990-11-30       4286   131
1990-12-31       6047   132
```

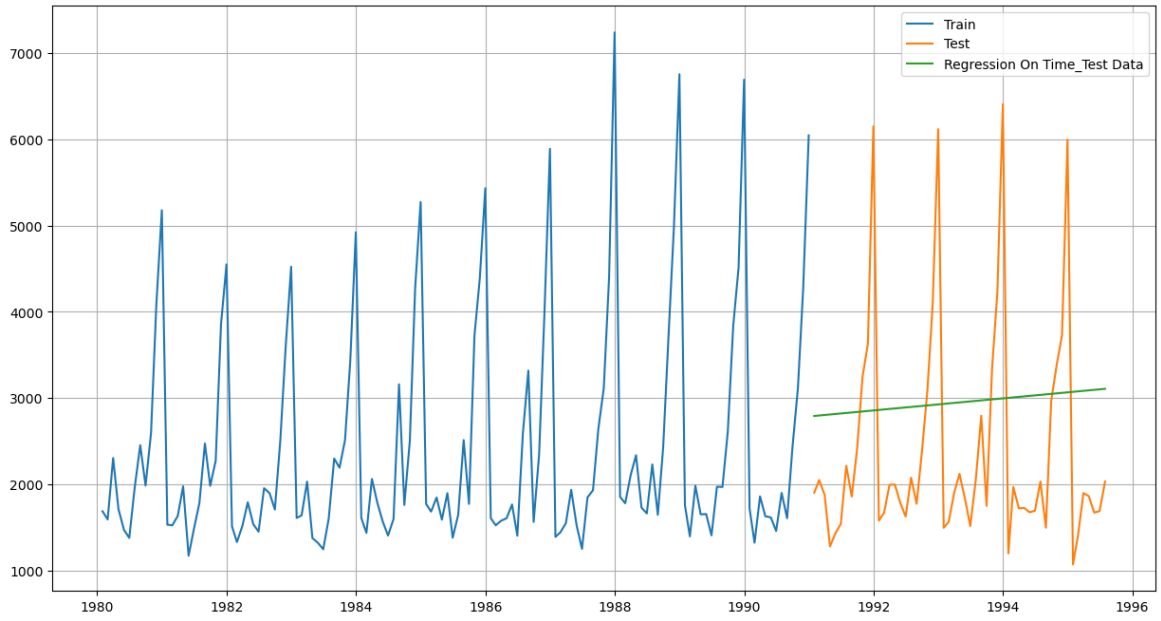
**Test data-**

```
First few rows of Test Data
            Sparkling  time
Time_Stamp
1991-01-31       1902   133
1991-02-28       2049   134
1991-03-31       1874   135
1991-04-30       1279   136
1991-05-31       1432   137

Last few rows of Test Data
            Sparkling  time
Time_Stamp
1995-03-31       1897   183
1995-04-30       1862   184
1995-05-31       1670   185
1995-06-30       1688   186
1995-07-31       2031   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.

```
▾ LinearRegression
LinearRegression()
```

**Plotting train, test and predictions on test-**




Linear regression can handle trend to an extent but not seasonality.

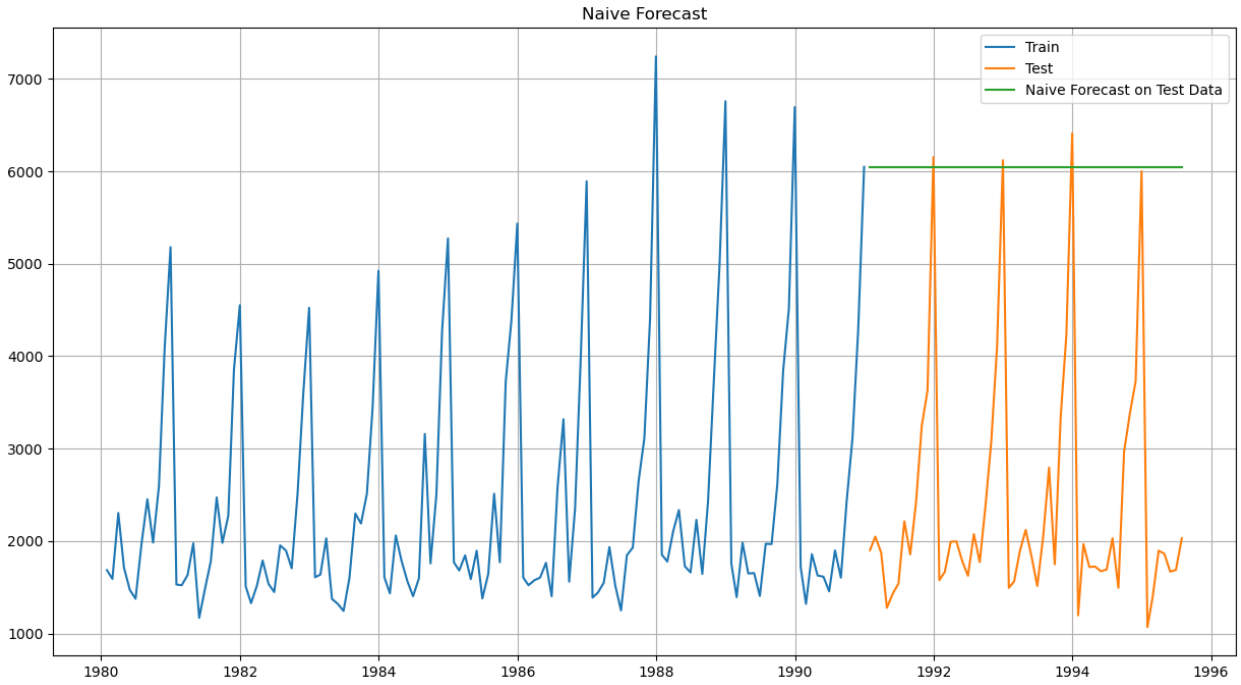
**Model Evaluation-**

For RegressionOnTime forecast on the Test Data, RMSE is **1389.14**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |

**Model 2: Naive Approach:**

- Uses the last observed value
- Ignores trend, seasonality.




**Model evaluation-**

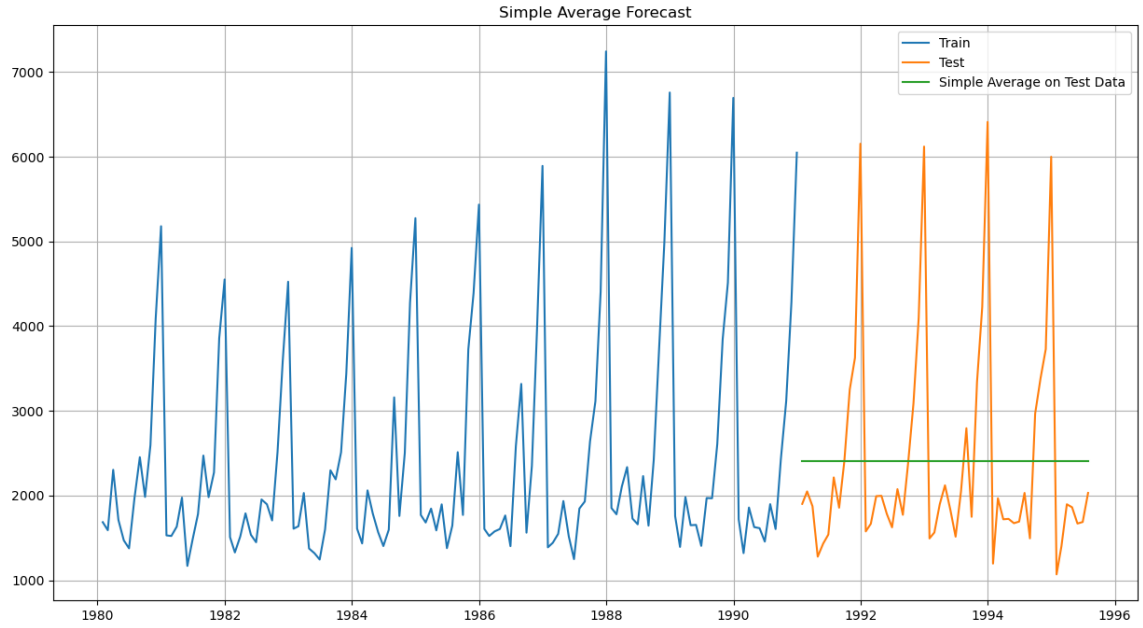
For RegressionOnTime forecast on the Test Data, RMSE is **3864.279**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |

**Method 3: Simple Average**

For this particular simple average method, we will forecast by using the average of the training values.

| | Sparkling | mean_forecast |
|---|---|---|
| **Time_Stamp** | | |
| **1991-01-31** | 1902 | 2403.780303 |
| **1991-02-28** | 2049 | 2403.780303 |
| **1991-03-31** | 1874 | 2403.780303 |
| **1991-04-30** | 1279 | 2403.780303 |
| **1991-05-31** | 1432 | 2403.780303 |




**Model evaluation-**

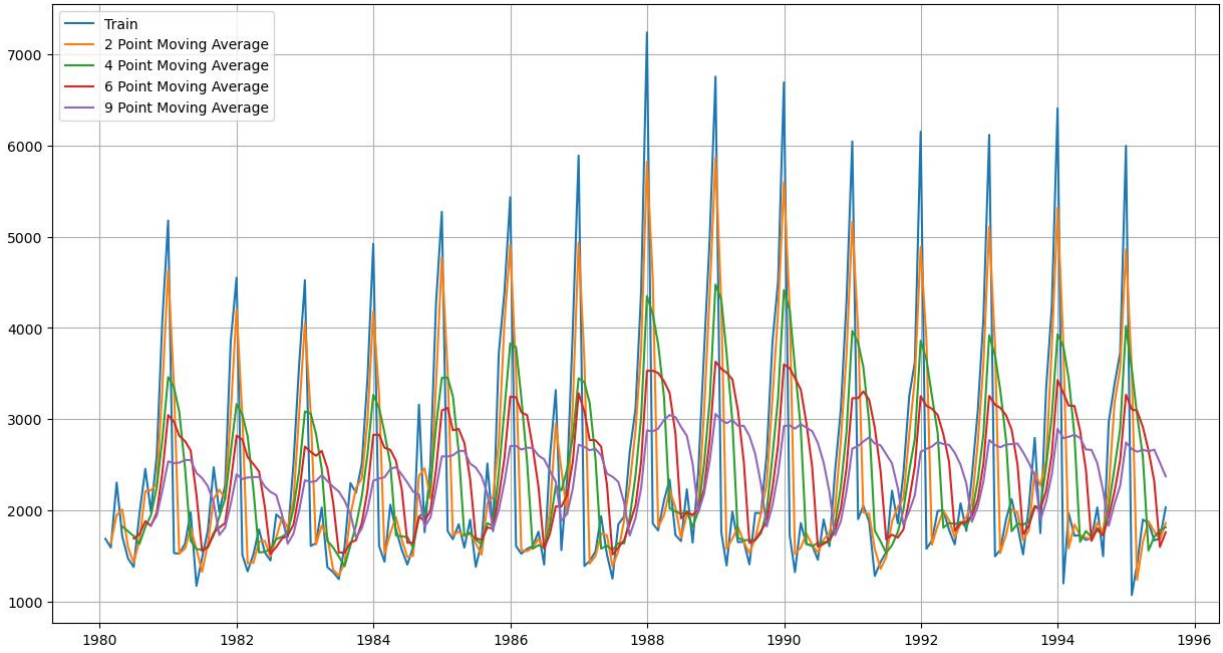
For Simple Average forecast on the Test Data, RMSE is **1275.082**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |

**Method 4: Moving Average(MA)-**

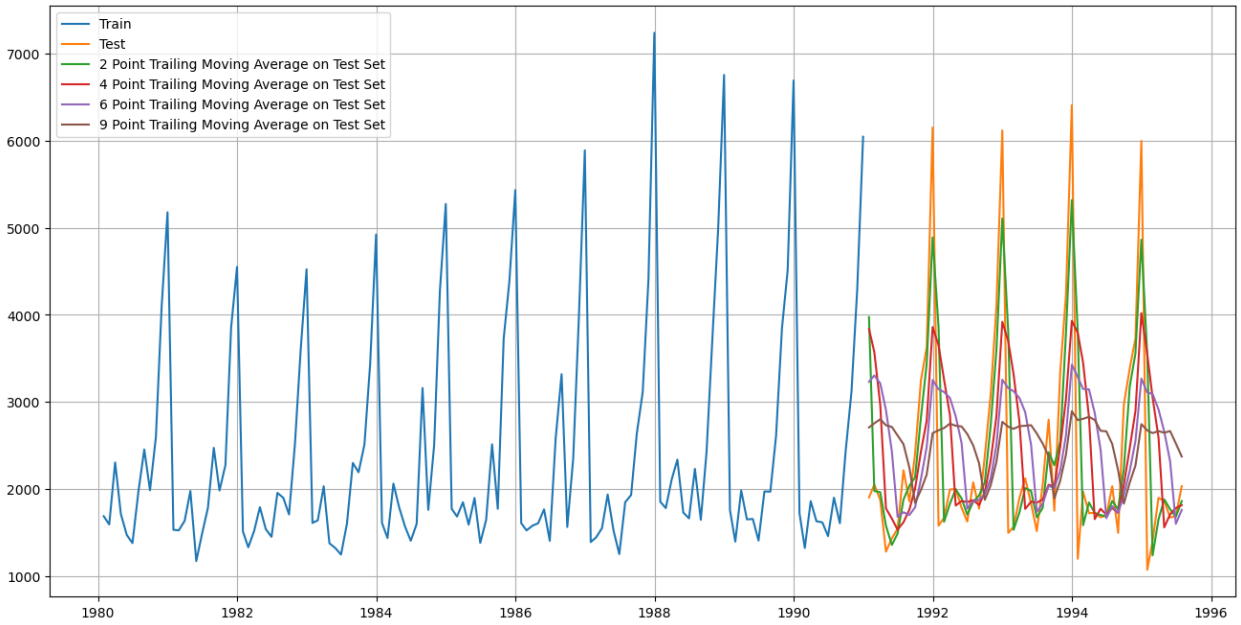
- For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.
- For Moving Average, we are going to average over the entire data.
- Here, we are considering 2 month, 4 months, 6 months and 9 months moving average

| Time_Stamp | Sparkling | Trailing_2 | Trailing_4 | Trailing_6 | Trailing_9 |
|---|---|---|---|---|---|
| **1980-01-31** | 1686 | NaN | NaN | NaN | NaN |
| **1980-02-29** | 1591 | 1638.5 | NaN | NaN | NaN |
| **1980-03-31** | 2304 | 1947.5 | NaN | NaN | NaN |
| **1980-04-30** | 1712 | 2008.0 | 1823.25 | NaN | NaN |
| **1980-05-31** | 1471 | 1591.5 | 1769.50 | NaN | NaN |




- From the graph, we can observe that the 2 point is most closest to actuals.

**Let us split the data into train and test and plot this Time Series-**



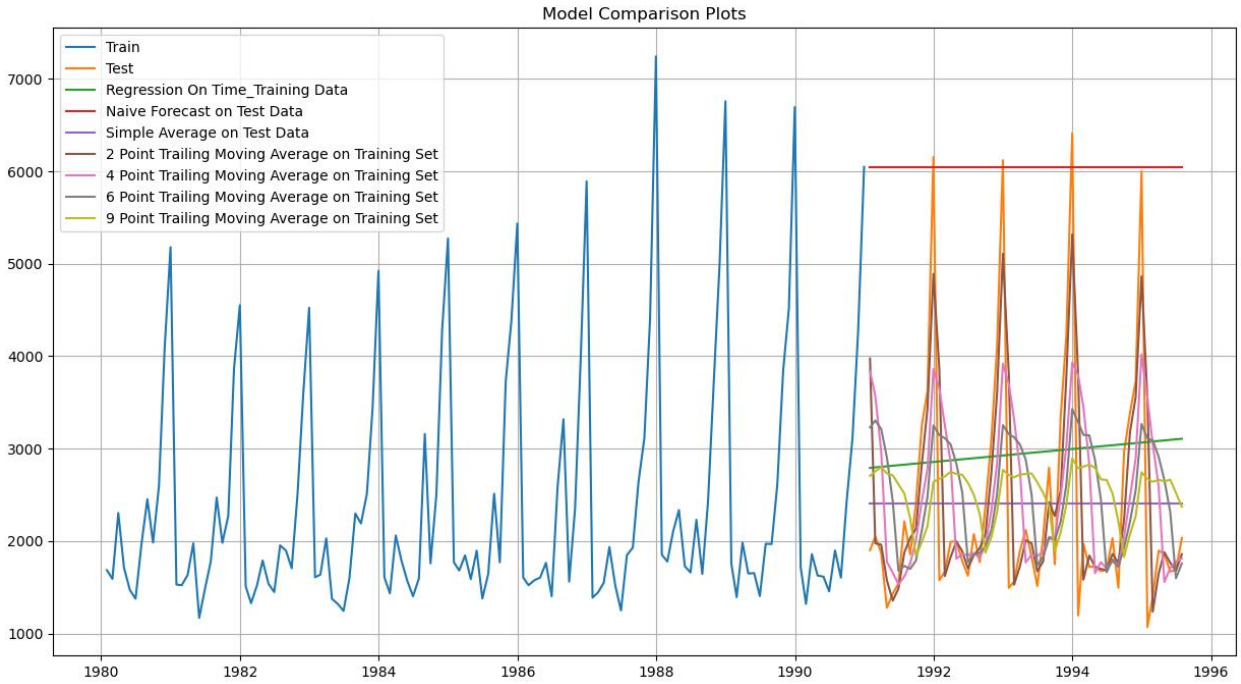

**Model Evaluation-**

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 813.401
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 1156.590
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 1283.927
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 1346.278
```

- On comparing actuals and predictions, we can see least amount of error in 2 months as expected.

**Consolidated plots of all Models-**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |
| **2pointTrailingMovingAverage** | 813.400684 |
| **4pointTrailingMovingAverage** | 1156.589694 |
| **6pointTrailingMovingAverage** | 1283.927428 |
| **9pointTrailingMovingAverage** | 1346.278315 |




**Method 5: Exponential Smoothing methods-**

- Exponential smoothing methods consist of flattening time series data.
- Exponential smoothing averages or exponentially weighted moving averages consist of forecast based on previous periods data with exponentially declining influence on the older observations.
- Exponential smoothing methods consist of special case exponential moving with notation ETS (Error, Trend, Seasonality) where each can be none(N), additive (N), additive damped (Ad), Multiplicative (M) or multiplicative damped (Md).
- One or more parameters control how fast the weights decay.
- These parameters have values between 0 and 1.

**5.1 SES - ETS(A, N, N) - Simple Exponential Smoothing with additive errors-**

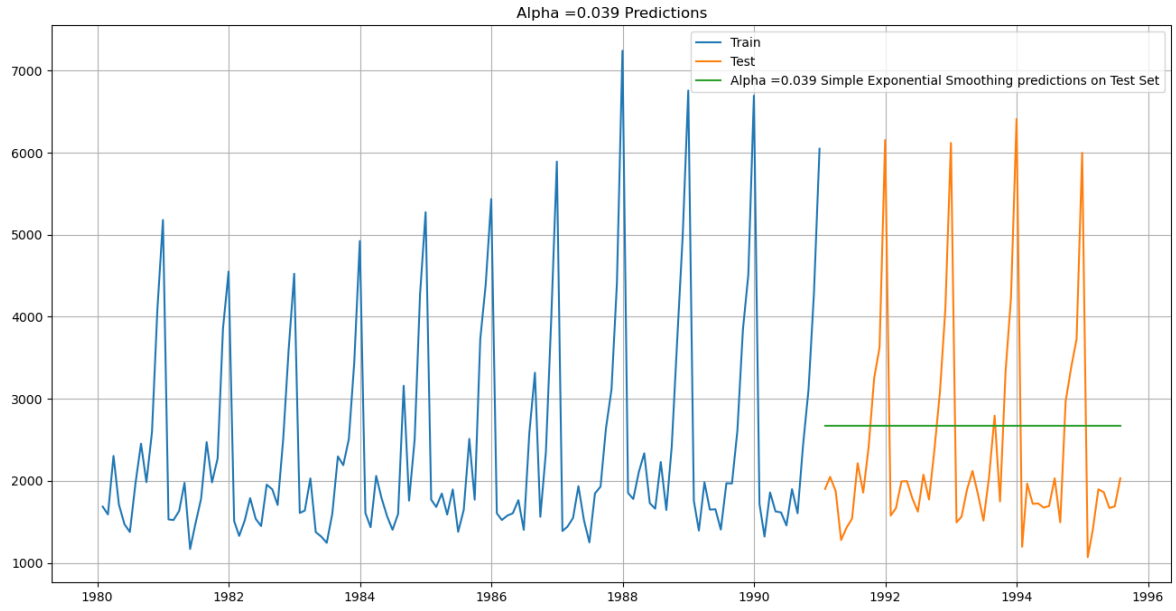
- The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing (SES).
- This method is suitable for forecasting data with no clear trend or seasonal pattern i.e. it only handles level.
- Parameter alpha is called the smoothing constant and its value lies between 0 and 1.
- Since the model uses only one smoothing constant, it is called Single Exponential Smoothing.
- *SimpleExpSmoothing* class must be instantiated and passed the training data.
- The fit() function is then called providing the fit configuration, the alpha value, smoothing_level. If this is omitted or set to None, the model will automatically optimize the value.

**Parameters-**

```
{'smoothing_level': 0.03953488372093023,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1686.0,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- Trend and seasonality is not considered hence its nan.

| Time_Stamp | Sparkling | predict |
|---|---|---|
| **1991-01-31** | 1902 | 2676.676366 |
| **1991-02-28** | 2049 | 2676.676366 |
| **1991-03-31** | 1874 | 2676.676366 |
| **1991-04-30** | 1279 | 2676.676366 |
| **1991-05-31** | 1432 | 2676.676366 |




**Model Evaluation-**

For Alpha =0.039, Simple Exponential Smoothing Model forecast on the Test Data, RMSE is **1304.927**

| | Test RMSE |
|---|---|
| RegressionOnTime | 1389.135175 |
| NaiveModel | 3864.279352 |
| SimpleAverageModel | 1275.081804 |
| 2pointTrailingMovingAverage | 813.400684 |
| 4pointTrailingMovingAverage | 1156.589694 |
| 6pointTrailingMovingAverage | 1283.927428 |
| 9pointTrailingMovingAverage | 1346.278315 |
| Alpha=0.039,SimpleExponentialSmoothing | 1304.927405 |

- The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.
- Here, Alpha is more closer to 0. Hence, the data is falling beyond actuals.

### 5.2 Double Exponential Smoothing (Holt's Model)

- One of the drawbacks of the simple exponential smoothing is that the model does not do well in the presence of the trend.
- Applicable when data has Trend but no seasonality.
- Level is the local mean.
- This model is an extension of SES known as Double Exponential model which estimates two smoothing parameters-
    i. One smoothing parameter α corresponds to the level series
    ii. A second smoothing parameter β corresponds to the trend series.

**Parameters-**

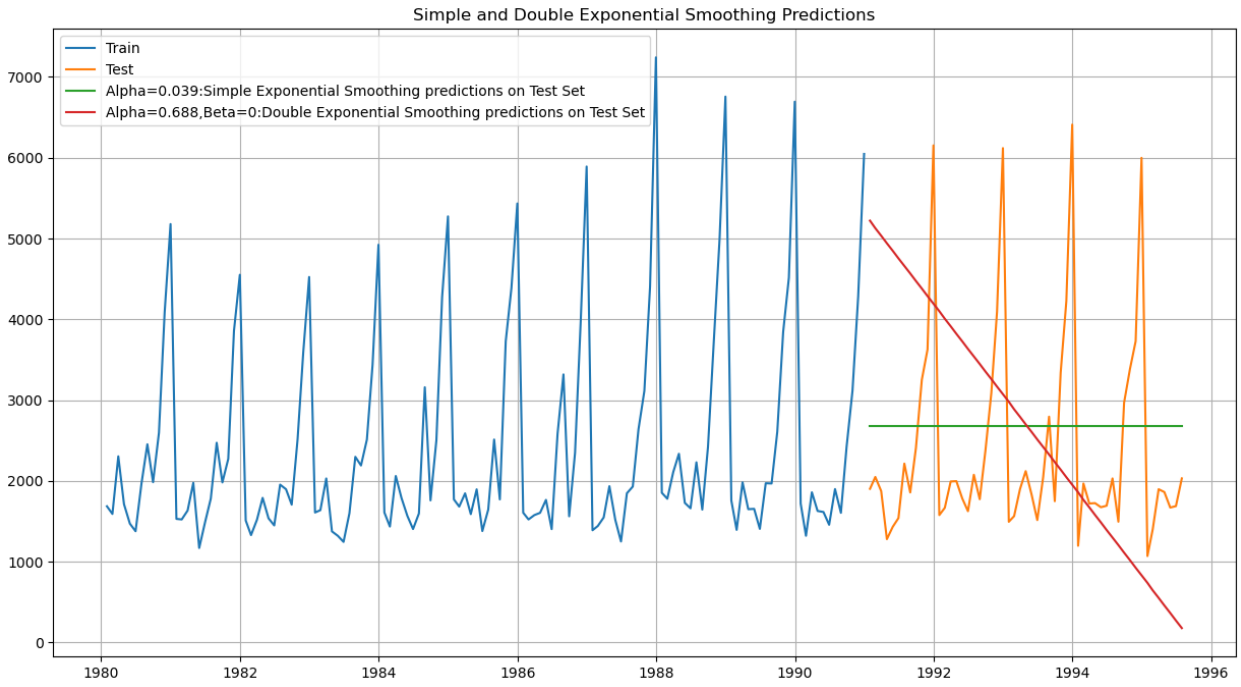
```
Holt model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 0.6885714285714285, 'smoothing_trend': 9.999999999999999e-05, 'smoothing_seasonal': nan, 'damping_trend': n
an, 'initial_level': 1686.0, 'initial_trend': -95.0, 'initial_seasons': array([], dtype=float64), 'use_boxcox': False, 'lamda':
None, 'remove_bias': False}
```

- Alpha is more closer to 1. Hence, the data is falling towards actuals.
- Beta closer to 0 which means old past trends are relevant to the forecast.
- Seasonality is not considered hence its nan.

**Predictions-**

| | |
|---|---|
| 1991-01-31 | 5221.278699 |
| 1991-02-28 | 5127.886554 |
| 1991-03-31 | 5034.494409 |
| 1991-04-30 | 4941.102264 |
| 1991-05-31 | 4847.710119 |
| 1991-06-30 | 4754.317974 |
| 1991-07-31 | 4660.925829 |
| 1991-08-31 | 4567.533684 |
| 1991-09-30 | 4474.141539 |
| 1991-10-31 | 4380.749394 |
| 1991-11-30 | 4287.357249 |
| 1991-12-31 | 4193.965104 |
| 1992-01-31 | 4100.572959 |
| 1992-02-29 | 4007.180813 |
| 1992-03-31 | 3913.788668 |
| 1992-04-30 | 3820.396523 |
| 1992-05-31 | 3727.004378 |
| 1992-06-30 | 3633.612233 |
| 1992-07-31 | 3540.220088 |
| 1992-08-31 | 3446.827943 |
| 1992-09-30 | 3353.435798 |
| 1992-10-31 | 3260.043653 |
| 1992-11-30 | 3166.651508 |
| 1992-12-31 | 3073.259363 |
| 1993-01-31 | 2979.867218 |
| 1993-02-28 | 2886.475073 |
| 1993-03-31 | 2793.082928 |
| 1993-04-30 | 2699.690783 |
| 1993-05-31 | 2606.298638 |
| 1993-06-30 | 2512.906493 |
| 1993-07-31 | 2419.514348 |
| 1993-08-31 | 2326.122203 |
| 1993-09-30 | 2232.730058 |
| 1993-10-31 | 2139.337913 |
| 1993-11-30 | 2045.945768 |
| 1993-12-31 | 1952.553623 |
| 1993-01-31 | 2979.867218 |
| 1993-02-28 | 2886.475073 |
| 1993-03-31 | 2793.082928 |
| 1993-04-30 | 2699.690783 |
| 1993-05-31 | 2606.298638 |
| 1993-06-30 | 2512.906493 |
| 1993-07-31 | 2419.514348 |
| 1993-08-31 | 2326.122203 |
| 1993-09-30 | 2232.730058 |
| 1993-10-31 | 2139.337913 |
| 1993-11-30 | 2045.945768 |
| 1993-12-31 | 1952.553623 |
| 1994-01-31 | 1859.161478 |
| 1994-02-28 | 1765.769333 |
| 1994-03-31 | 1672.377188 |
| 1994-04-30 | 1578.985043 |
| 1994-05-31 | 1485.592898 |
| 1994-06-30 | 1392.200753 |
| 1994-07-31 | 1298.808608 |

```
1994-08-31    1205.416463
1994-09-30    1112.024318
1994-10-31    1018.632173
1994-11-30     925.240028
1994-12-31     831.847883
1995-01-31     738.455738
1995-02-28     645.063593
1995-03-31     551.671448
1995-04-30     458.279303
1995-05-31     364.887158
1995-06-30     271.495013
1995-07-31     178.102868
Freq: M, dtype: float64
```




- We see that the double exponential smoothing is picking up the trend component along with the level component as well.

**Model evaluation-**

DES RMSE: **2007.238525758568**

| | Test RMSE |
|---|---|
| RegressionOnTime | 1389.135175 |
| NaiveModel | 3864.279352 |
| SimpleAverageModel | 1275.081804 |
| 2pointTrailingMovingAverage | 813.400684 |
| 4pointTrailingMovingAverage | 1156.589694 |
| 6pointTrailingMovingAverage | 1283.927428 |
| 9pointTrailingMovingAverage | 1346.278315 |
| Alpha=0.039,SimpleExponentialSmoothing | 1304.927405 |
| Alpha=0.688,Beta=0:DES | 2007.238526 |

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

**Holt-Winters - ETS(A, A, A) - Holt Winter's linear method with additive errors**

- Three parameters $\alpha$, $\beta$ and $\gamma$ are estimated in this model. Level, Trend and Seasonality are accounted for in this model.

**Parameters-**

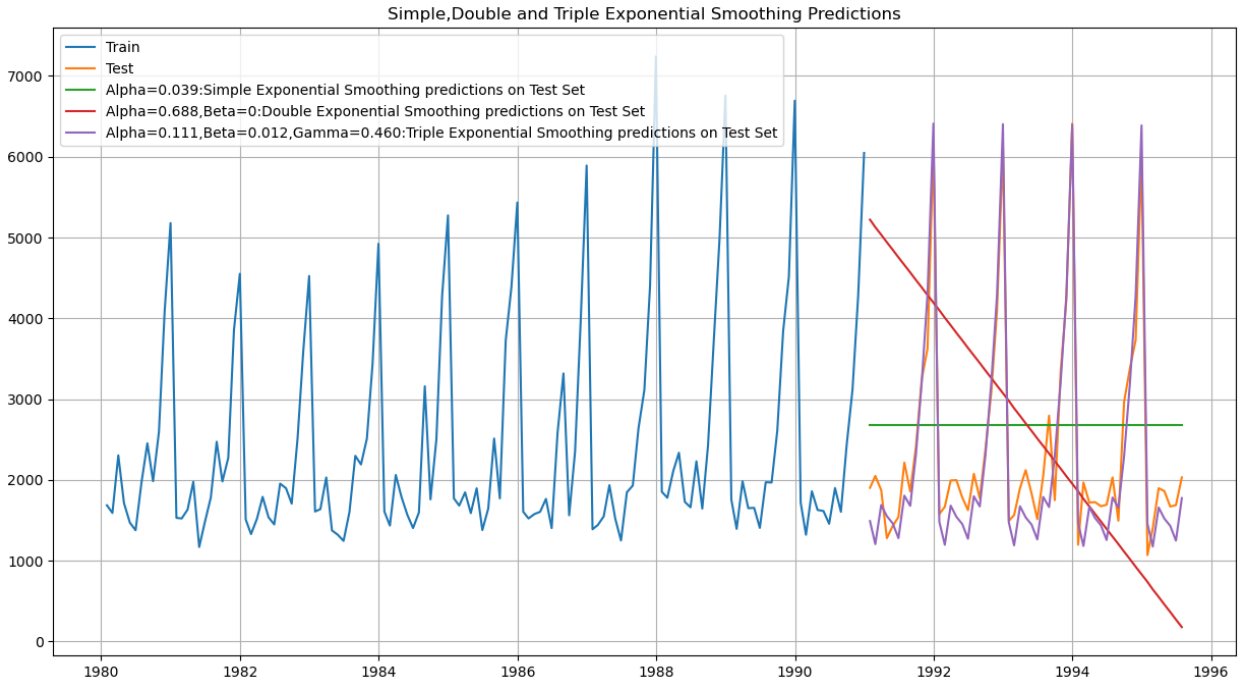
```
Holt Winters model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 0.11127227248079453, 'smoothing_trend': 0.012360804305088534, 'smoothing_seasonal': 0.46071766688111543, 'd
amping_trend': nan, 'initial_level': 2356.577980956387, 'initial_trend': -0.10243675533021725, 'initial_seasons': array([-636.2
3319334, -722.9832009 , -398.64410813, -473.43045416,
       -808.42473284, -815.34991402, -384.23065038,   72.99484403,
       -237.44226045,  272.32608272, 1541.37737052, 2590.07692296]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

- Alpha is more closer to 0. Hence, the data is falling farther from actuals.
- Beta closer to 0 which means old past trends are relevant to the forecast.
- Gamma closer to 0, old pasts seasons are relevant for forecast.

**Forecasting using this model for the duration of the test set-**

| | |
|---|---|
| 1991-01-31 | 1490.402890 |
| 1991-02-28 | 1204.525152 |
| 1991-03-31 | 1688.734182 |
| 1991-04-30 | 1551.226125 |
| 1991-05-31 | 1461.197883 |
| 1991-06-30 | 1278.646707 |
| 1991-07-31 | 1804.885616 |
| 1991-08-31 | 1678.955032 |
| 1991-09-30 | 2315.373126 |
| 1991-10-31 | 3224.976222 |
| 1991-11-30 | 4299.301434 |
| 1991-12-31 | 6410.712237 |
| 1992-01-31 | 1482.829908 |
| 1992-02-29 | 1196.952170 |
| 1992-03-31 | 1681.161200 |
| 1992-04-30 | 1543.653143 |
| 1992-05-31 | 1453.624901 |
| 1992-06-30 | 1271.073725 |
| 1992-07-31 | 1797.312634 |
| 1992-08-31 | 1671.382050 |
| 1992-09-30 | 2307.800144 |
| 1992-10-31 | 3217.403240 |
| 1992-11-30 | 4291.728452 |
| 1992-12-31 | 6403.139255 |
| 1993-01-31 | 1475.256926 |
| 1993-02-28 | 1189.379188 |
| 1993-03-31 | 1673.588218 |
| 1993-04-30 | 1536.080160 |
| 1993-05-31 | 1446.051919 |
| 1993-06-30 | 1263.500743 |
| 1993-07-31 | 1789.739652 |
| 1993-08-31 | 1663.809068 |
| 1993-09-30 | 2300.227162 |
| 1993-10-31 | 3209.830258 |
| 1993-11-30 | 4284.155470 |
| 1993-12-31 | 6395.566273 |
| 1993-01-31 | 1475.256926 |
| 1993-02-28 | 1189.379188 |
| 1993-03-31 | 1673.588218 |
| 1993-04-30 | 1536.080160 |
| 1993-05-31 | 1446.051919 |
| 1993-06-30 | 1263.500743 |
| 1993-07-31 | 1789.739652 |
| 1993-08-31 | 1663.809068 |
| 1993-09-30 | 2300.227162 |
| 1993-10-31 | 3209.830258 |
| 1993-11-30 | 4284.155470 |
| 1993-12-31 | 6395.566273 |
| 1994-01-31 | 1467.683944 |
| 1994-02-28 | 1181.806206 |
| 1994-03-31 | 1666.015236 |
| 1994-04-30 | 1528.507178 |
| 1994-05-31 | 1438.478937 |
| 1994-06-30 | 1255.927761 |
| 1994-07-31 | 1782.166669 |

```
1994-08-31    1656.236085
1994-09-30    2292.654180
1994-10-31    3202.257275
1994-11-30    4276.582488
1994-12-31    6387.993291
1995-01-31    1460.110962
1995-02-28    1174.233224
1995-03-31    1658.442254
1995-04-30    1520.934196
1995-05-31    1430.905955
1995-06-30    1248.354779
1995-07-31    1774.593687
Freq: M, dtype: float64
```




**Model evaluation-**

TES RMSE: **378.95102286703**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |
| **2pointTrailingMovingAverage** | 813.400684 |
| **4pointTrailingMovingAverage** | 1156.589694 |
| **6pointTrailingMovingAverage** | 1283.927428 |
| **9pointTrailingMovingAverage** | 1346.278315 |
| **Alpha=0.039,SimpleExponentialSmoothing** | 1304.927405 |
| **Alpha=0.688,Beta=0:DES** | 2007.238526 |
| **Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive** | 378.951023 |

**Holt-Winters - ETS(A, A, M) - Holt Winter's linear method with multiplicative seasonality-**

**Parameters-**

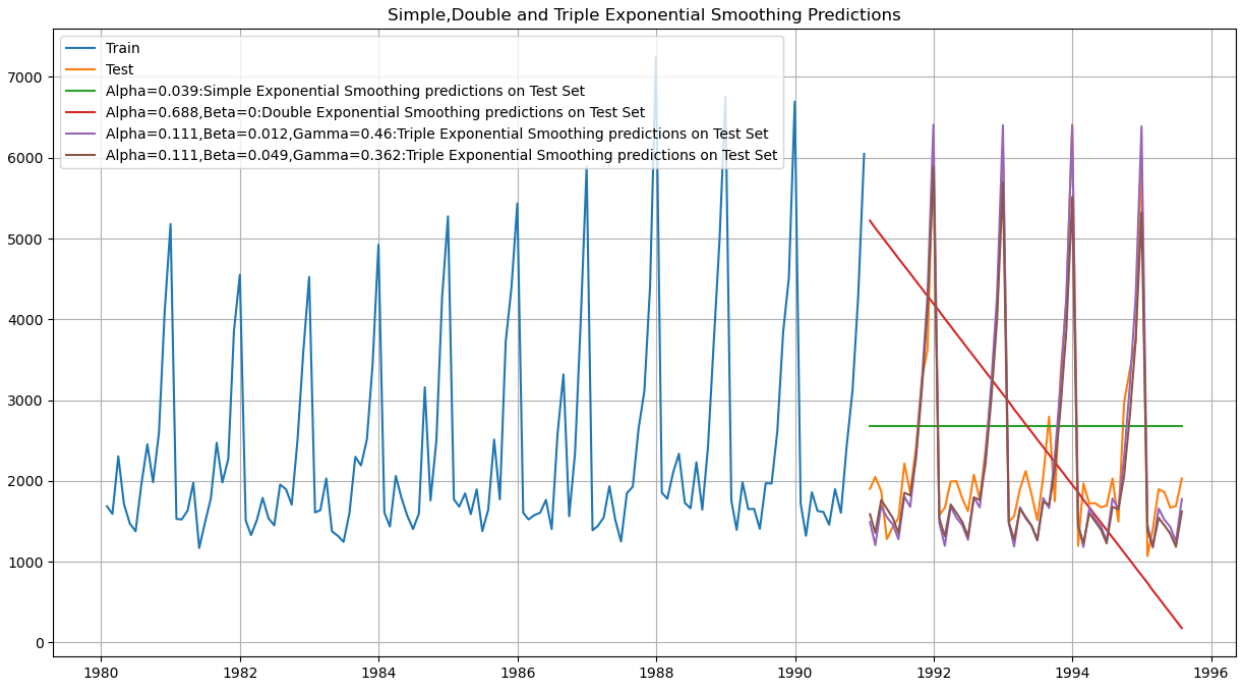
```
Holt Winters model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 0.11133818361298699, 'smoothing_trend': 0.049505131019509915, 'smoothing_seasonal': 0.3620795793580111, 'da
mping_trend': nan, 'initial_level': 2356.4967888704355, 'initial_trend': -10.187944726007238, 'initial_seasons': array([0.71296
382, 0.68242226, 0.90755008, 0.80515228, 0.65597218,
       0.65414505, 0.88617935, 1.13345121, 0.92046306, 1.21337874,
       1.87340336, 2.37811768]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

- Alpha is more closer to 0. Hence, the data is falling farther from actuals.
- Beta closer to 0  which means old past trends are relevant to the forecast.
- Gamma closer to 0, old pasts seasons are relevant for forecast.

**Predictions-**

| | |
|---|---|
| 1991-01-31 | 1587.497468 |
| 1991-02-28 | 1356.394925 |
| 1991-03-31 | 1762.929755 |
| 1991-04-30 | 1656.165933 |
| 1991-05-31 | 1542.002730 |
| 1991-06-30 | 1355.102435 |
| 1991-07-31 | 1854.197719 |
| 1991-08-31 | 1820.513188 |
| 1991-09-30 | 2276.971718 |
| 1991-10-31 | 3122.024202 |
| 1991-11-30 | 4128.528561 |
| 1991-12-31 | 5890.064588 |
| 1992-01-31 | 1538.233708 |
| 1992-02-29 | 1314.193684 |
| 1992-03-31 | 1707.937498 |
| 1992-04-30 | 1604.369388 |
| 1992-05-31 | 1493.650618 |
| 1992-06-30 | 1312.499576 |
| 1992-07-31 | 1795.750753 |
| 1992-08-31 | 1762.976871 |
| 1992-09-30 | 2204.819253 |
| 1992-10-31 | 3022.831861 |
| 1992-11-30 | 3997.009544 |
| 1992-12-31 | 5701.930382 |
| 1993-01-31 | 1488.969948 |
| 1993-02-28 | 1271.992443 |
| 1993-03-31 | 1652.945240 |
| 1993-04-30 | 1552.572843 |
| 1993-05-31 | 1445.298507 |
| 1993-06-30 | 1269.896716 |
| 1993-07-31 | 1737.303788 |
| 1993-08-31 | 1705.440555 |
| 1993-09-30 | 2132.666788 |
| 1993-10-31 | 2923.639519 |
| 1993-11-30 | 3865.490526 |
| 1993-12-31 | 5513.796176 |
| 1993-01-31 | 1488.969948 |
| 1993-02-28 | 1271.992443 |
| 1993-03-31 | 1652.945240 |
| 1993-04-30 | 1552.572843 |
| 1993-05-31 | 1445.298507 |
| 1993-06-30 | 1269.896716 |
| 1993-07-31 | 1737.303788 |
| 1993-08-31 | 1705.440555 |
| 1993-09-30 | 2132.666788 |
| 1993-10-31 | 2923.639519 |
| 1993-11-30 | 3865.490526 |
| 1993-12-31 | 5513.796176 |
| 1994-01-31 | 1439.706189 |
| 1994-02-28 | 1229.791202 |
| 1994-03-31 | 1597.952983 |
| 1994-04-30 | 1500.776298 |
| 1994-05-31 | 1396.946396 |
| 1994-06-30 | 1227.293857 |
| 1994-07-31 | 1678.856822 |

```
1994-08-31    1647.904238
1994-09-30    2060.514323
1994-10-31    2824.447177
1994-11-30    3733.971509
1994-12-31    5325.661970
1995-01-31    1390.442429
1995-02-28    1187.589961
1995-03-31    1542.960726
1995-04-30    1448.979753
1995-05-31    1348.594284
1995-06-30    1184.690998
1995-07-31    1620.409857
Freq: M, dtype: float64
```




**Model accuracy-**

TES_am RMSE: **404.286809456071**

| | Test RMSE |
|---|---|
| RegressionOnTime | 1389.135175 |
| NaiveModel | 3864.279352 |
| SimpleAverageModel | 1275.081804 |
| 2pointTrailingMovingAverage | 813.400684 |
| 4pointTrailingMovingAverage | 1156.589694 |
| 6pointTrailingMovingAverage | 1283.927428 |
| 9pointTrailingMovingAverage | 1346.278315 |
| Alpha=0.039,SimpleExponentialSmoothing | 1304.927405 |
| Alpha=0.688,Beta=0:DES | 2007.238526 |
| Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive | 378.951023 |
| Alpha=0.111,Beta=0.049,Gamma=0.362:TES Multiplicative | 404.286809 |

**Inference-**

- Triple Exponential Smoothing has performed the best on the test as expected since the data had both trend and seasonality.

**2.5 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.**
**Note: Stationarity should be checked at alpha = 0.05.**

- A Time Series is considered to be stationary when statistical properties such as the variance and (auto) correlation are constant over time.
- Stationary Time Series allows us to think of the statistical properties of the time series as not changing in time, which enables us to build appropriate statistical models for forecasting based on past data.
- Stationarity means that the autocorrelation of lag 'k' depends on k, but not on time t.

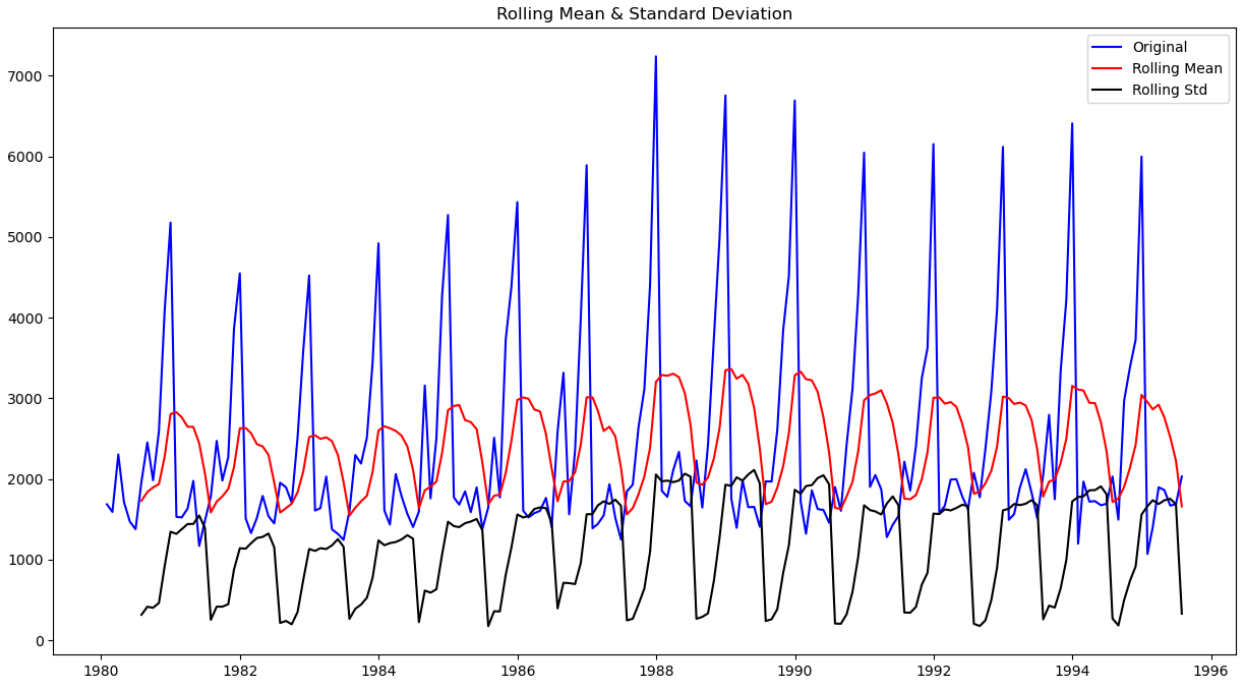
**How to check for Stationarity-**

**Dickey-Fuller Test -**

Dicky Fuller Test on the timeseries is run to check for stationarity of data.

**Null Hypothesis :** Time Series is non-stationary.
**Alternate Hypothesis :** Time Series is stationary.

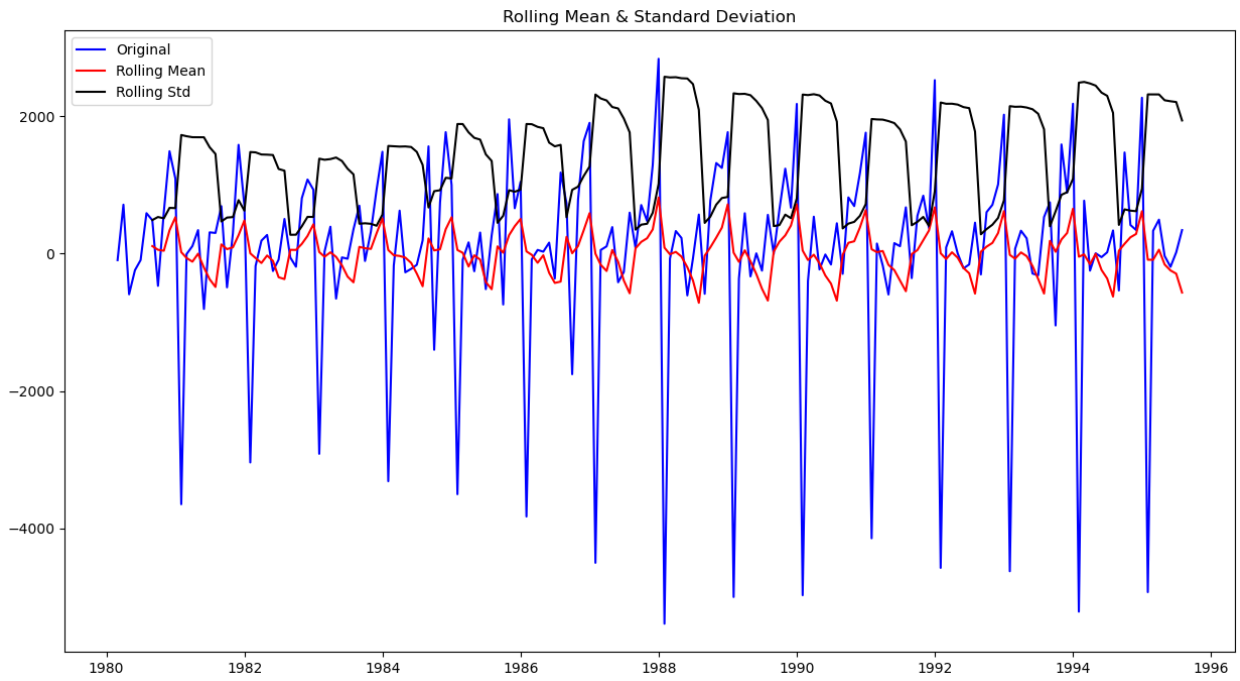
So, ideally if p-value < 0.05 then null hypothesis: TS is non-stationary is rejected else the TS is non-stationary is failed to be rejected .




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.360497
p-value                          0.601061
#Lags Used                      11.000000
Number of Observations Used    175.000000
Critical Value (1%)             -3.468280
Critical Value (5%)             -2.878202
Critical Value (10%)            -2.575653
dtype: float64
```

- Here, p-value > 0.05. Hence, 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.

**Differencing-**




```
Results of Dickey-Fuller Test:
Test Statistic                -45.050301
p-value                         0.000000
#Lags Used                     10.000000
Number of Observations Used   175.000000
Critical Value (1%)            -3.468280
Critical Value (5%)            -2.878202
Critical Value (10%)           -2.575653
dtype: float64
```

- We see that at alpha = 0.05 the Time Series is indeed stationary. (p-value <0.05)

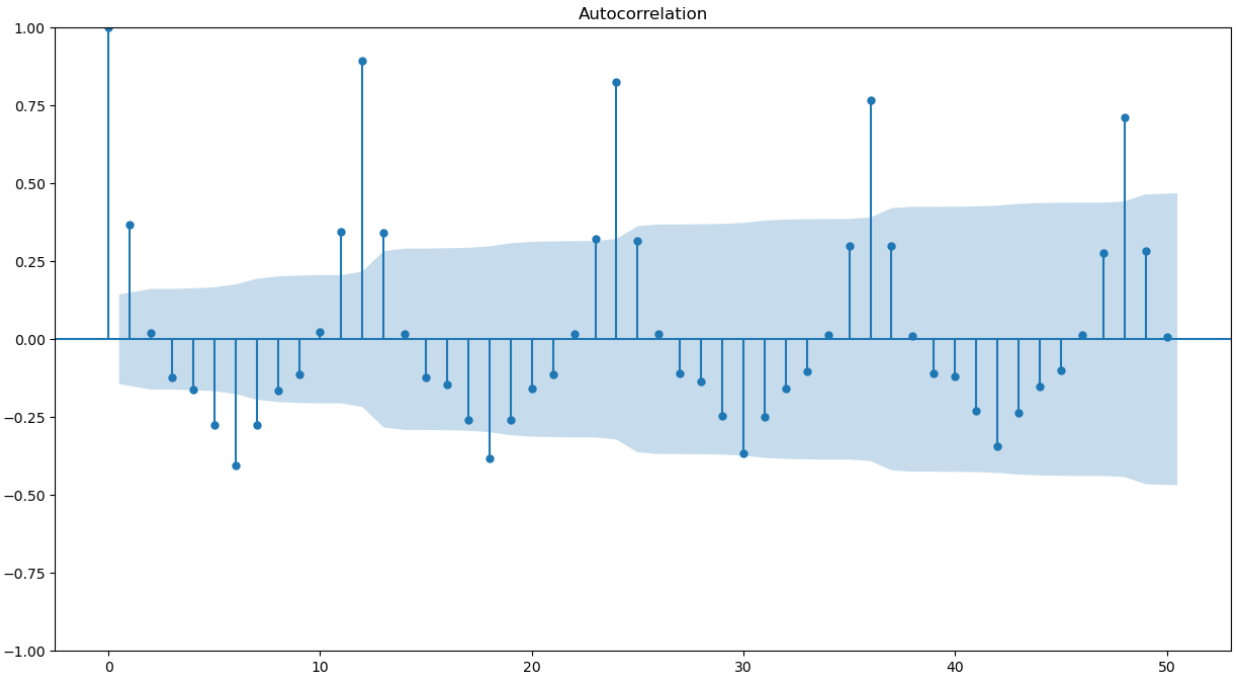
**Autocorrelation function plots on the whole data-**

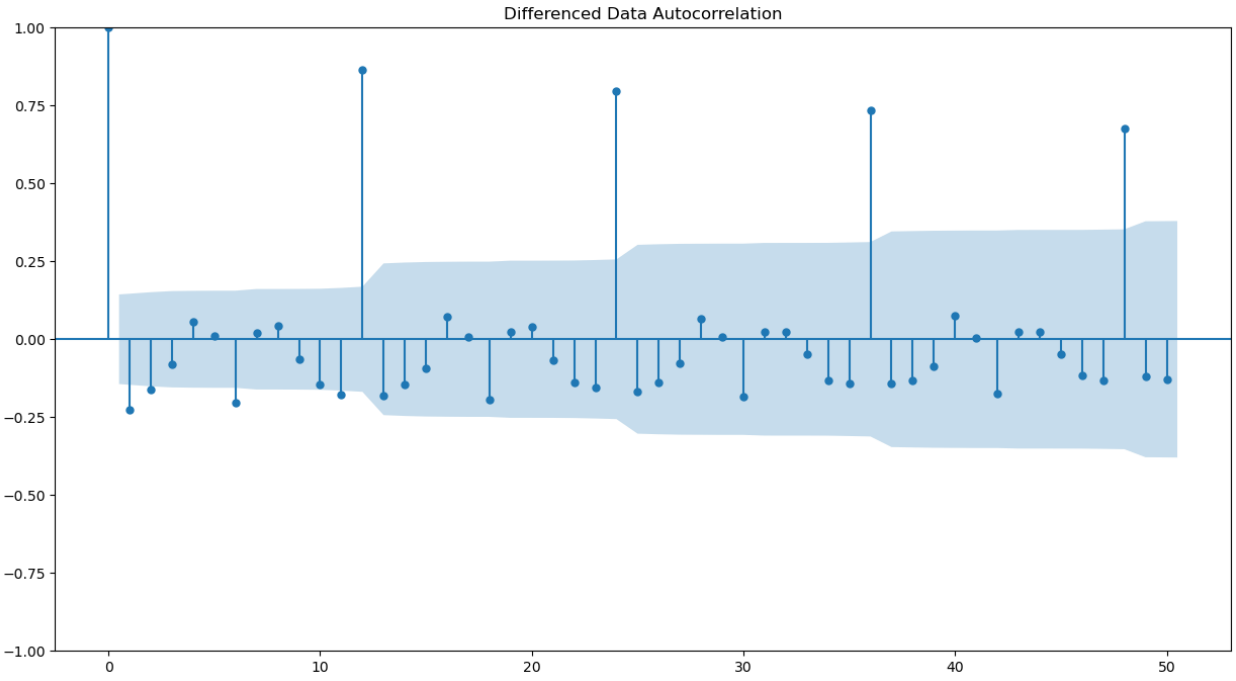
- PACF (Partial Autocorrelation Function) Plot-
  - To determine value of ‘p’
  - Measures relationship after eliminating effect of lags

- ACF Plot-
  - To determine value of ‘q’
  - Measures how much Time series is correlated with itself at different lags

- Autocorrelation decreases as lag increases.

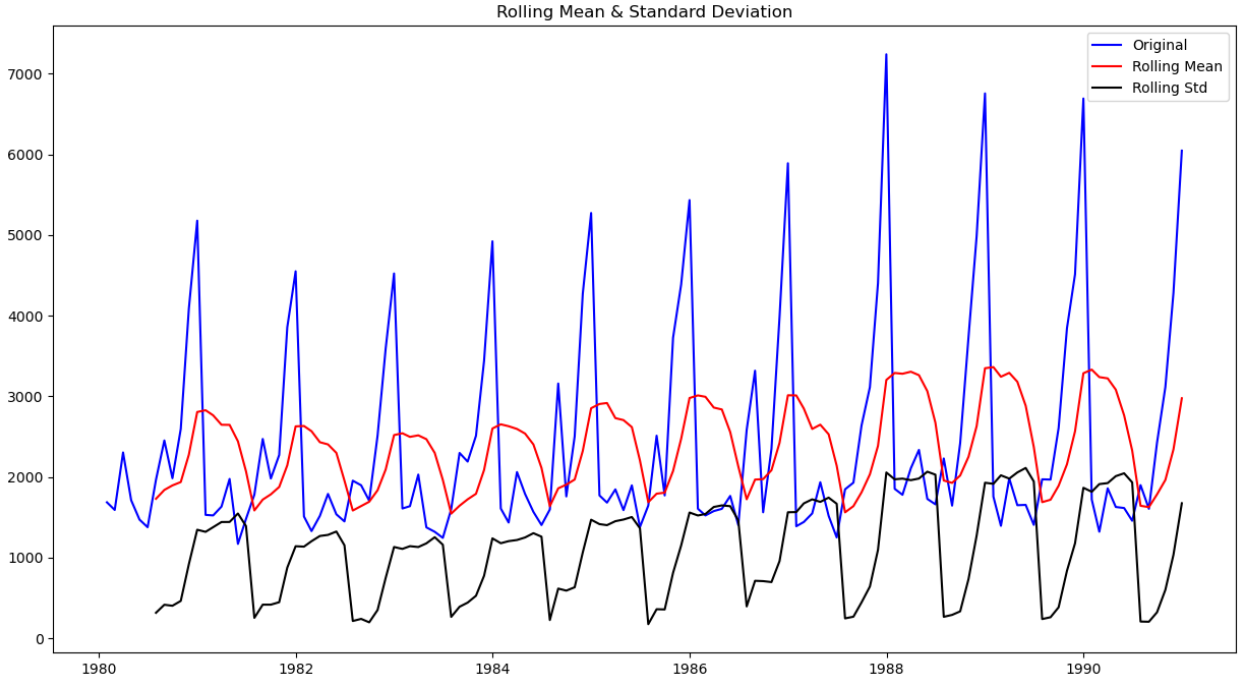
- The terms ‘Lag’ is the number of data points we are looking back.




- From the above plots, we can say that there seems to be a seasonality in the data.
- From ACF Plot, we can see that the seasonality repeats at 12 months and at 6 months, it is repeating in mirror image. Seasonality can be 6 or 12.
- From the PCF plot, the manual analysis of Q value= 2

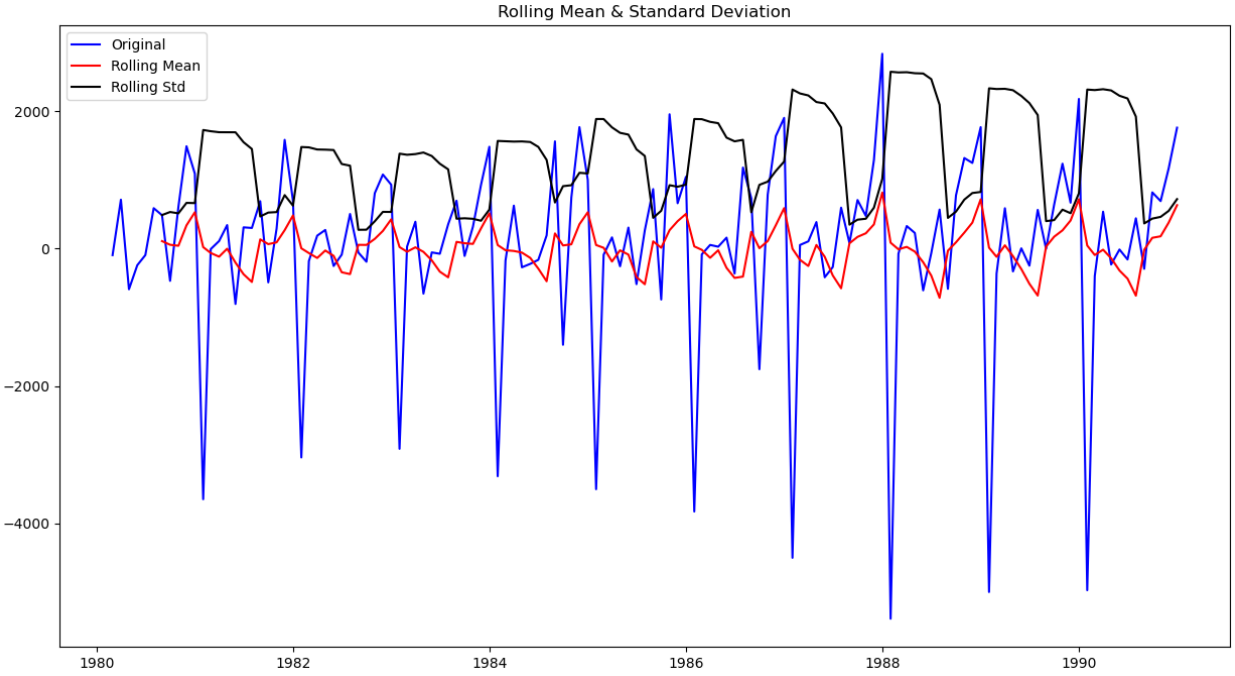
**Check for stationarity of the Training Data Time Series-**




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.208926
p-value                          0.669744
#Lags Used                      12.000000
Number of Observations Used    119.000000
Critical Value (1%)             -3.486535
Critical Value (5%)             -2.886151
Critical Value (10%)            -2.579896
dtype: float64
```

- We see that the series is non-stationary at alpha = 0.05 as p-value (0.669744) > 0.05. Hence, null hypothesis is failed to be rejected.

**Taking a difference of order 1-**




```
Results of Dickey-Fuller Test:
Test Statistic                -8.005007e+00
p-value                        2.280104e-12
#Lags Used                     1.100000e+01
Number of Observations Used    1.190000e+02
Critical Value (1%)           -3.486535e+00
Critical Value (5%)           -2.886151e+00
Critical Value (10%)          -2.579896e+00
dtype: float64
```

- p-value is < 0.05. Hence TS is stationary.

**2.6 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.**

**2.6.1 ARIMA:**

- Auto Regressive Integrated Moving Average is a way of modeling time series data for forecasting or predicting future data points

- Improving AR Models by making Time Series stationary through Moving Average Forecasts
- ARIMA models consist of 3 components:-
    - AR model: The data is modeled based on past observations.
    - Integrated component: Whether the data needs to be differenced/transformed.
    - MA model: Previous forecast errors are incorporated into the model.

**ARIMA Model building to estimate best p, d, q parameters using Lowest AIC Approach -**

- AIC- Akaike Information Criteria
- Lowest AIC value compared among different orders of 'p' is considered
- Lower the AIC, better the 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)
```

- d is constant with value of 1. Only 'p' and 'q' varies

**Calculating AIC value for different parameters-**

| | param | AIC |
|---|---|---|
| 0 | (0, 1, 0) | 2267.663036 |
| 1 | (0, 1, 1) | 2263.060016 |
| 2 | (0, 1, 2) | 2234.408323 |
| 3 | (1, 1, 0) | 2266.608539 |
| 4 | (1, 1, 1) | 2235.755095 |
| 5 | (1, 1, 2) | 2234.527200 |
| 6 | (2, 1, 0) | 2260.365744 |
| 7 | (2, 1, 1) | 2233.777626 |
| 8 | (2, 1, 2) | 2213.509213 |

**Sorting parameters based on AIC value-**

| | param | AIC |
|---|---|---|
| **8** | (2, 1, 2) | 2213.509213 |
| **7** | (2, 1, 1) | 2233.777626 |
| **2** | (0, 1, 2) | 2234.408323 |
| **5** | (1, 1, 2) | 2234.527200 |
| **4** | (1, 1, 1) | 2235.755095 |
| **6** | (2, 1, 0) | 2260.365744 |
| **1** | (0, 1, 1) | 2263.060016 |
| **3** | (1, 1, 0) | 2266.608539 |
| **0** | (0, 1, 0) | 2267.663036 |

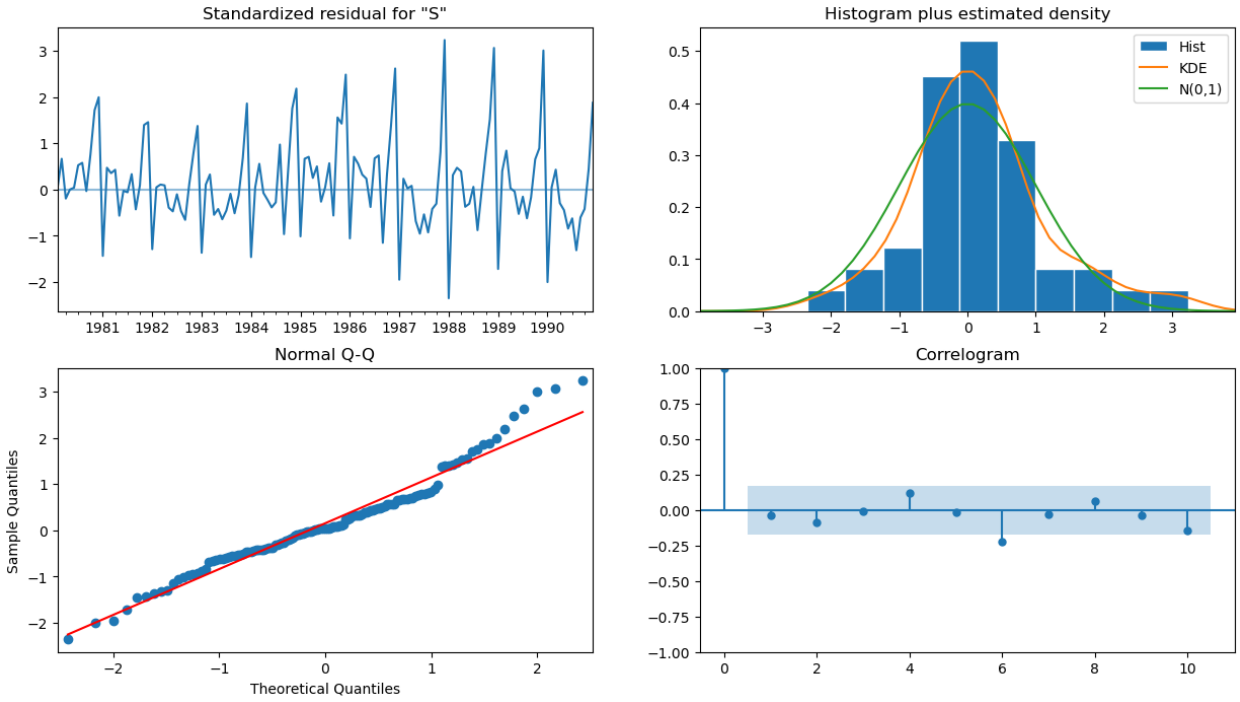
**Building ARIMA model with parameters considered best with lowest AIC value of 2213.50 i.e. (2,1,2) p=2, d=1, q=2**

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Sparkling   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1101.755
Date:                Fri, 26 Jan 2024   AIC                           2213.509
Time:                        18:49:16   BIC                           2227.885
Sample:                    01-31-1980   HQIC                          2219.351
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.3121      0.046     28.782      0.000       1.223       1.401
ar.L2         -0.5593      0.072     -7.740      0.000      -0.701      -0.418
ma.L1         -1.9917      0.109    -18.216      0.000      -2.206      -1.777
ma.L2          0.9999      0.110      9.108      0.000       0.785       1.215
sigma2      1.099e+06   1.99e-07   5.51e+12      0.000     1.1e+06     1.1e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):                14.46
Prob(Q):                              0.67   Prob(JB):                         0.00
Heteroskedasticity (H):               2.43   Skew:                             0.61
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.08
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 5.89e+27. Standard errors may be unstable.
```

- 2 level of AR and 2 level of MA.
- MA contributing more in error.
- Series may not be dependent directly on lag 2
- Prob(Q)= 0.67 > 0.05, hence residuals are independent i.e no significant autocorrelation in the residuals

- Prob(JB)=0  indicates that the residuals do not follow a normal distribution
- Skewness= 0.61 which indicates a positive right skewness
- Kurtosis of 4.08 indicates presence of outliers or extreme values.
- The model is heteroskedastic.

**Diagnostics Plot-**




**4 plots in the residuals diagnostic plots tell us** :

- **Standardized residuals plot:** The top left plot shows 1-step-ahead standardized residuals. If model is working correctly, then no pattern should be obvious in the residuals.
- **Histogram plus estimated density plot:** This plot shows the distribution of the residuals. The orange line shows a smoothed version of this histogram, and the green line shows a normal distribution. If the model is good these two lines should be the same. Here there are small differences between them, which indicate that our model is doing just well enough.
- **Normal Q-Q plot:** The Q-Q plot compare the distribution of residuals to normal distribution. If the distribution of the residuals is normal, then all the points should lie along the red line, except for some values at the end, which is exactly happening in this case.
- **Correlogram plot:** The correlogram plot is the ACF plot of the residuals rather than the data. 95% of the correlations for lag >0  should not be significant (within the blue shades). If there is a significant correlation in the residuals, it means that there is information in the data that was not captured by the model, which is clearly not in this case.

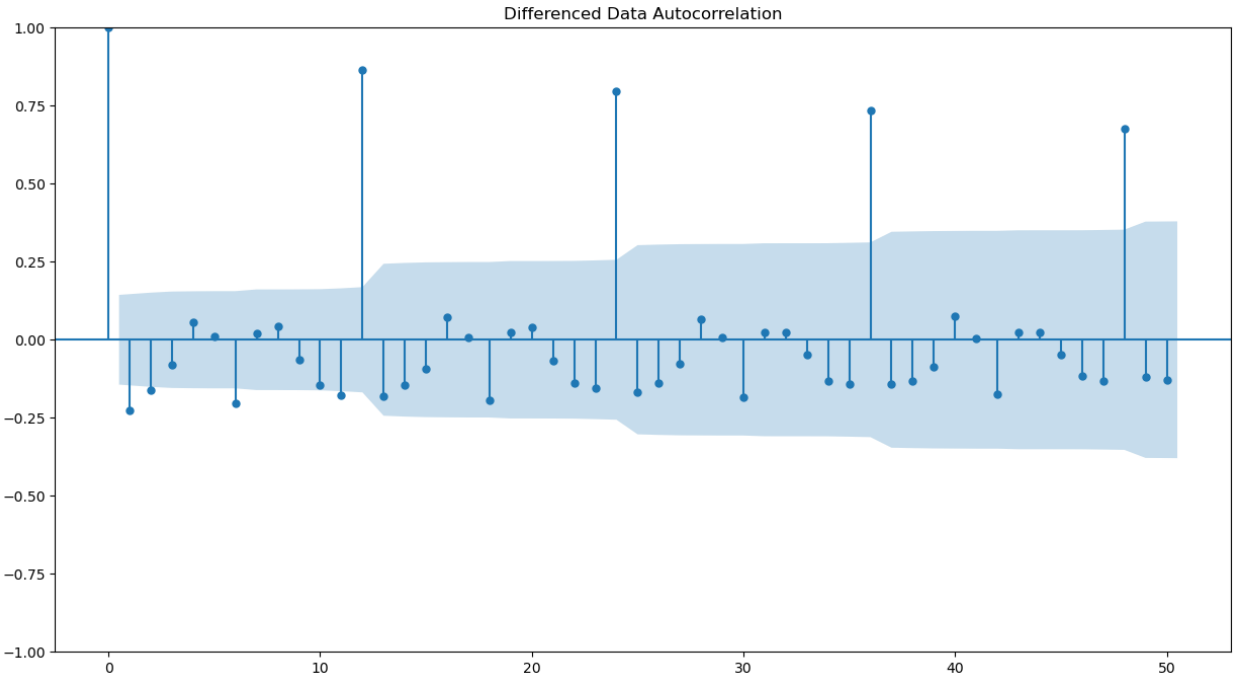
**Predict on the Test Set using this model and evaluate the model-**

RMSE: **1299.9797563580828**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |
| **2pointTrailingMovingAverage** | 813.400684 |
| **4pointTrailingMovingAverage** | 1156.589694 |
| **6pointTrailingMovingAverage** | 1283.927428 |
| **9pointTrailingMovingAverage** | 1346.278315 |
| **Alpha=0.039,SimpleExponentialSmoothing** | 1304.927405 |
| **Alpha=0.688,Beta=0:DES** | 2007.238526 |
| **Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive** | 378.951023 |
| **Alpha=0.111,Beta=0.049,Gamma=0.362:TES Multiplicative** | 404.286809 |
| **ARIMA(2,1,2)** | 1299.979756 |

**2.6.2 SARIMA Model-**

- The ARIMA models can be extended/improved to handle seasonal components of a data series.
- The seasonal autoregressive moving average model is given by SARIMA (p, d, q)(P, D, Q)F
- The model consists of:
    - Autoregressive and moving average components (p, q)
    - Seasonal autoregressive and moving average components (P, Q)
    - The ordinary and seasonal difference components of order ‘d’ and ‘D’
    - Seasonal frequency ‘F’
- The value for the parameters (p,d,q) and (P, D, Q) can be decided by comparing different values for each and taking the lowest AIC value for the model build.
- The value for F can be consolidated by ACF plot




- We see that there can be a seasonality of 6 as well as 12.

**Setting the seasonality as 6 to estimate parameters using auto SARIMA model-**

| | param | seasonal | AIC |
|---|---|---|---|
| 0 | (0, 1, 0) | (0, 0, 0, 6) | 2251.359720 |
| 1 | (0, 1, 0) | (0, 0, 1, 6) | 2152.378076 |
| 2 | (0, 1, 0) | (0, 0, 2, 6) | 1955.635554 |
| 3 | (0, 1, 0) | (1, 0, 0, 6) | 2164.409758 |
| 4 | (0, 1, 0) | (1, 0, 1, 6) | 2079.559984 |
| .. | ... | ... | ... |
| 76 | (2, 1, 2) | (1, 0, 1, 6) | 1955.605895 |
| 77 | (2, 1, 2) | (1, 0, 2, 6) | 1825.895689 |
| 78 | (2, 1, 2) | (2, 0, 0, 6) | 1763.293101 |
| 79 | (2, 1, 2) | (2, 0, 1, 6) | 1765.216566 |
| 80 | (2, 1, 2) | (2, 0, 2, 6) | 1729.363550 |

[81 rows x 3 columns]

**Sorting based on AIC values-**

| | param | seasonal | AIC |
|---|---|---|---|
| **53** | (1, 1, 2) | (2, 0, 2, 6) | 1727.670866 |
| **26** | (0, 1, 2) | (2, 0, 2, 6) | 1727.888805 |
| **80** | (2, 1, 2) | (2, 0, 2, 6) | 1729.363550 |
| **17** | (0, 1, 1) | (2, 0, 2, 6) | 1741.703671 |
| **44** | (1, 1, 1) | (2, 0, 2, 6) | 1743.379778 |

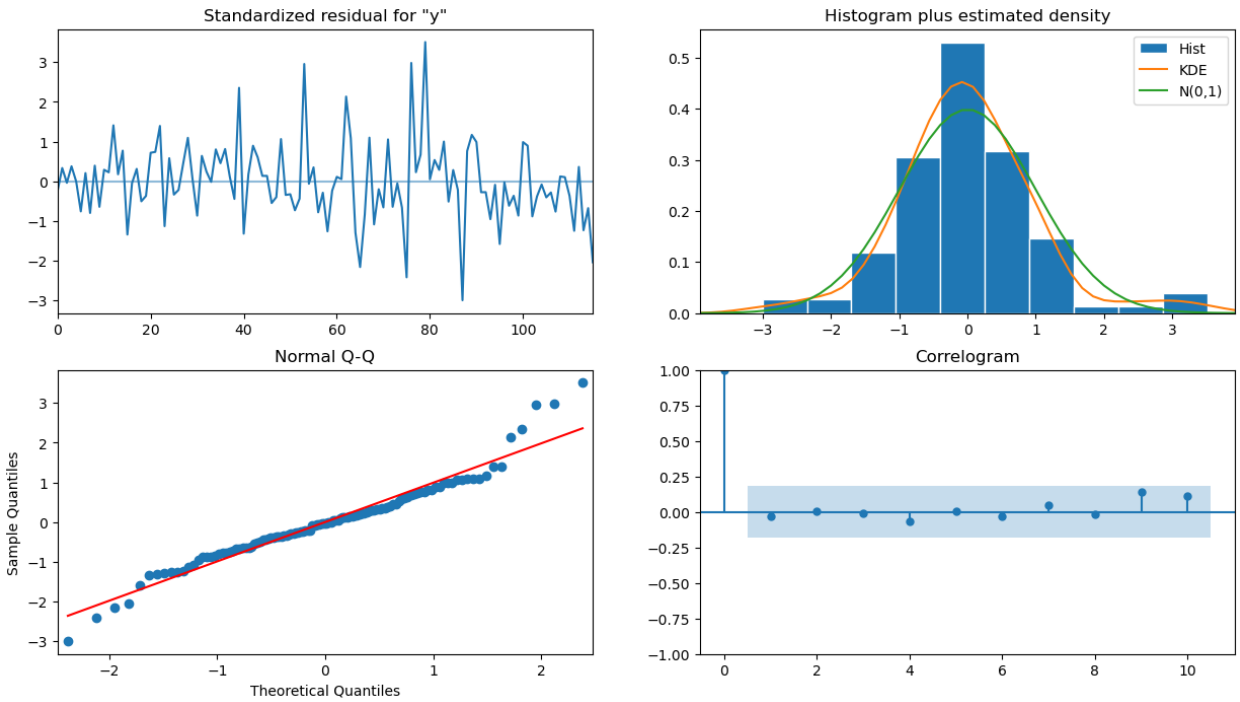
Here,
p = non-seasonal AR order = 1,
d = non-seasonal differencing = 1,
q = non-seasonal MA order = 2,
P = seasonal AR order = 2,
D = seasonal differencing = 0,
Q = seasonal MA order = 2,
S = time span of repeating seasonal pattern = 6

**Building SARIMA model with parameters considered best with lowest AIC value of 1727.671 i.e. (p,d,q) (P,D,Q,F): (1,1,2) (2,0,2,6)-**

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  132
Model:             SARIMAX(1, 1, 2)x(2, 0, 2, 6)   Log Likelihood                -855.835
Date:                           Fri, 26 Jan 2024   AIC                            1727.671
Time:                                   18:54:42   BIC                            1749.700
Sample:                                        0   HQIC                           1736.613
                                             - 132
Covariance Type:                             opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6451      0.286     -2.256      0.024      -1.206      -0.085
ma.L1         -0.3355      0.227     -1.475      0.140      -0.781       0.110
ma.L2         -0.8805      0.277     -3.179      0.001      -1.423      -0.338
ar.S.L6       -0.0045      0.027     -0.165      0.869      -0.057       0.049
ar.S.L12       1.0361      0.018     56.096      0.000       1.000       1.072
ma.S.L6        0.0676      0.152      0.444      0.657      -0.231       0.366
ma.S.L12      -0.6124      0.093     -6.592      0.000      -0.795      -0.430
sigma2      1.152e+05   1.79e+04      6.456      0.000    8.03e+04     1.5e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.09   Jarque-Bera (JB):                 25.26
Prob(Q):                              0.77   Prob(JB):                          0.00
Heteroskedasticity (H):               2.63   Skew:                              0.47
Prob(H) (two-sided):                  0.00   Kurtosis:                          5.09
===================================================================================
```

- We got L12 as frequency of time series is Monthly and seasonality is intra year. And we have given (2,0,2).
- Most contributing to forecast is ar.S.L12 as coeff is high and p-value is 0.
- Prob(Q), a high p-value (close to 1) suggests that there is no evidence of significant autocorrelation in the residuals. Hence residuals are independent.
- Prob(JB) is 0, residuals are not normally distributed.
- Skewness= 0.47 which indicates a positive right skewness
- Kurtosis of 5.09 indicates presence of outliers or extreme values.
- Heteroskedasticity of 2.63 indicates the variance of the residuals over time.

**Diagnostics Plot:**




- Lag is not beyond Confidence region so correlation is less. We have considered significant zones appropriately i.e. learnt from errors.
- Errors seem normally distributed from the plot.

**Predict on the Test Set using this model and evaluate the model-**

| y | mean | mean_se | mean_ci_lower | mean_ci_upper |
|---|---|---|---|---|
| 0 | 1330.304021 | 380.553546 | 584.432776 | 2076.175265 |
| 1 | 1177.201978 | 392.106015 | 408.688311 | 1945.715645 |
| 2 | 1625.828724 | 392.300967 | 856.932957 | 2394.724491 |
| 3 | 1546.293507 | 397.705597 | 766.804859 | 2325.782154 |
| 4 | 1308.594051 | 398.926118 | 526.713226 | 2090.474875 |

**Model evaluation:**

RMSE: **626.992402111805**

| | Test RMSE |
|---|---|
| RegressionOnTime | 1389.135175 |
| NaiveModel | 3864.279352 |
| SimpleAverageModel | 1275.081804 |
| 2pointTrailingMovingAverage | 813.400684 |
| 4pointTrailingMovingAverage | 1156.589694 |
| 6pointTrailingMovingAverage | 1283.927428 |
| 9pointTrailingMovingAverage | 1346.278315 |
| Alpha=0.039,SimpleExponentialSmoothing | 1304.927405 |
| Alpha=0.688,Beta=0:DES | 2007.238526 |
| Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive | 378.951023 |
| Alpha=0.111,Beta=0.049,Gamma=0.362:TES Multiplicative | 404.286809 |
| ARIMA(2,1,2) | 1299.979756 |
| SARIMA(1,1,2)(2,0,2,6) | 626.992402 |

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

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |
| **2pointTrailingMovingAverage** | 813.400684 |
| **4pointTrailingMovingAverage** | 1156.589694 |
| **6pointTrailingMovingAverage** | 1283.927428 |
| **9pointTrailingMovingAverage** | 1346.278315 |
| **Alpha=0.039,SimpleExponentialSmoothing** | 1304.927405 |
| **Alpha=0.688,Beta=0:DES** | 2007.238526 |
| **Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive** | 378.951023 |
| **Alpha=0.111,Beta=0.049,Gamma=0.362:TES Multiplicative** | 404.286809 |
| **ARIMA(2,1,2)** | 1299.979756 |
| **SARIMA(1,1,2)(2,0,2,6)** | 626.992402 |

- TES (Triple Exponential Smoothing) models with different alpha, beta, and gamma values provide relatively lower RMSE values (378.95 and 404.29, respectively). These models are performing better than other simpler models.
- The SARIMA model with the specified parameters provides an intermediate RMSE value of 626.99. It seems to perform better than simpler models but not as well as the TES models.
- Moving average models with different window sizes show increasing RMSE values. The larger the window size, the less responsive the model is to recent changes in the data resulting in higher error.
- The Simple Average Model, which predicts future values based on the average of historical values has an RMSE of 1275.08. This model may not capture trends or seasonality well, leading to a higher error.
- The ARIMA model with orders (2,1,2) has an RMSE of 1299.98. It considers autoregressive and moving average components along with differencing. The performance is comparable to simpler models.
- Simple Exponential Smoothing with a low alpha value has an RMSE of 1304.93. The low alpha places less weight on recent observations potentially making it less responsive to changes.
- A regression model on time has an RMSE of 1389.14. This approach assumes a linear relationship with time and may not capture complex patterns.

- The DES model with specific alpha and beta values has a relatively high RMSE of 2007.24. The high alpha may be causing the model to overly rely on recent observations potentially leading to higher errors.
- The Naive Model, which predicts future values based on the most recent observation has the highest RMSE of 3864.28. This model does not consider any patterns or trends in the data.

**2.8 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.**

| | Test RMSE |
|---|---|
| Alpha=0.111,Beta=0.012,Gamma=0.46:TES Additive | 378.951023 |
| Alpha=0.111,Beta=0.049,Gamma=0.362:TES Multiplicative | 404.286809 |
| SARIMA(1,1,2)(2,0,2,6) | 626.992402 |
| 2pointTrailingMovingAverage | 813.400684 |
| 4pointTrailingMovingAverage | 1156.589694 |
| SimpleAverageModel | 1275.081804 |
| 6pointTrailingMovingAverage | 1283.927428 |
| ARIMA(2,1,2) | 1299.979756 |
| Alpha=0.039,SimpleExponentialSmoothing | 1304.927405 |
| 9pointTrailingMovingAverage | 1346.278315 |
| RegressionOnTime | 1389.135175 |
| Alpha=0.688,Beta=0:DES | 2007.238526 |
| NaiveModel | 3864.279352 |

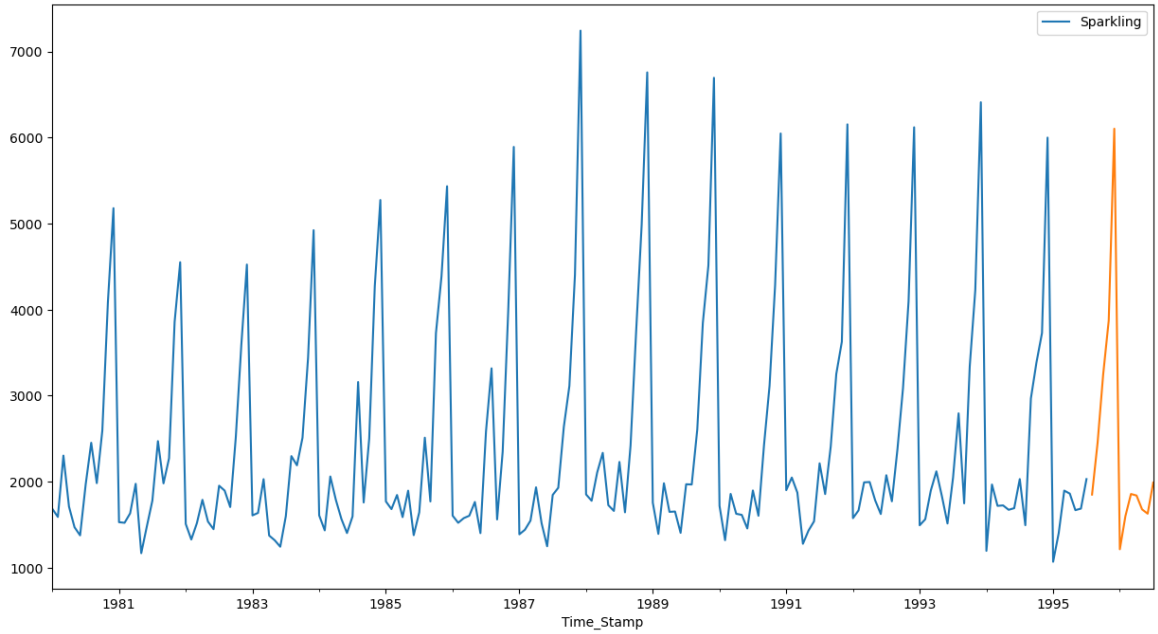
- On comparing the RMSE values, TES additive with params alpha= 0.111, Beta= 0.012 and Gamma= 0.46 seems the most optimum model.

**Building it on whole data and testing its accuracy-**

**Model evaluation-**

RMSE: **368.1199705679527**

**Plotting the predictions for next 12 months-**



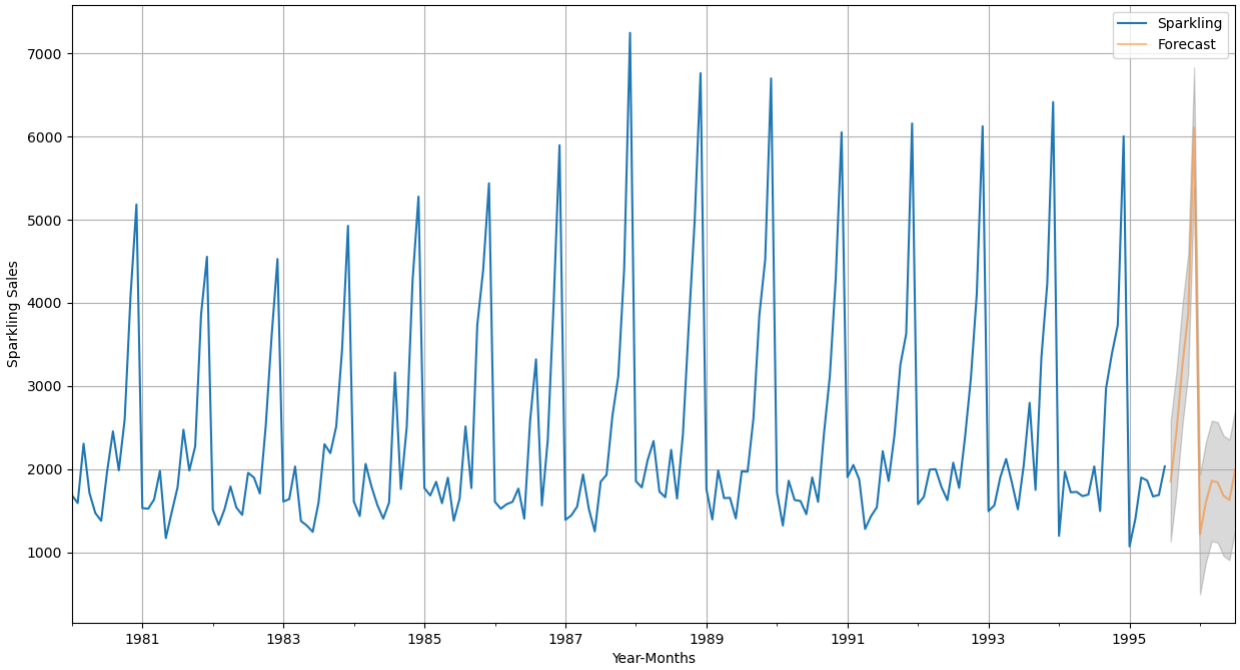

One assumption that we have made over here while calculating the confidence bands is that the standard deviation of the forecast distribution is almost equal to the residual standard deviation.

We have calculated the upper and lower confidence bands at 95% confidence level
Here we are taking the multiplier to be 1.96 as we want to plot with respect to a 95% confidence intervals.

**Forecast dataframe along with their confidence intervals-**

| | lower_CI | prediction | upper_ci |
|---|---|---|---|
| **1995-08-31** | 1127.564038 | 1851.014405 | 2574.464772 |
| **1995-09-30** | 1731.765860 | 2455.216227 | 3178.666594 |
| **1995-10-31** | 2522.446436 | 3245.896803 | 3969.347170 |
| **1995-11-30** | 3150.038433 | 3873.488800 | 4596.939167 |
| **1995-12-31** | 5379.092987 | 6102.543354 | 6825.993721 |

**Plotting forecast along with confidence band-**




**Forecasted values-**

Forecasted Values_Sparkling.csv

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

- The dataset contains 187 entries with no missing values, representing sparkling wine sales from January 1980 to July 1995.
- It exhibits a seasonal pattern with fluctuations, indicating potential external factors influencing sales.
- The presence of outliers suggests irregularities, and the seasonality is observed with peaks in December and lower sales in January and February.
- The ARIMA and SARIMA models were applied to capture the underlying patterns and seasonality in the sales data.
- The SARIMA model with parameters (0,1,2)x(2,0,2,12) performed well, considering lower RMSE values and capturing seasonality.

- The Triple Exponential Smoothing (TES) model with specific alpha, beta, and gamma values demonstrated the lowest RMSE, making it an optimal choice for forecasting.
- The model forecasts sale of around 29339 units of Sparkling wine sales on an average in next 12 months (Aug 1995- July 1996). And approximately 2444 units per month.
- Trends and Seasonal Patterns:
    - The data reveals an increasing trend in sparkling wine sales over the years, reaching a peak in 1988. However, there is a significant drop in sales in 1995.
    - Seasonal patterns show higher sales in the end of the second quarter and the beginning of the third quarter, with a reduction in the last quarter.

**Business Strategy:**

- Companies should be prepared for higher sales fluctuations or uncertainties in July, as indicated by the data.
- Special marketing or promotional activities can be planned during peak sales seasons to meet the increased consumer demand.
- Companies should manage inventory effectively by increasing the stock during peak seasons and reducing it during lower sales periods.
- Investigate and understand the factors contributing to outliers, especially in 1995, to address irregularities and improve forecasting accuracy.
- Consider launching new products or promotions during high-sales seasons to maximize revenue.
- Continuously monitor sales patterns, customer preferences, and market trends to adapt strategies and to stay competitive.
- Implement marketing strategies, inventory optimization strategies in order to be aligned with the sales trends.
- Enhance customer engagement during peak seasons through loyalty programs or discounts to encourage purchases.

By incorporating these recommendations, the company can perform better by aligning its strategies with observed patterns, improve forecasting accuracy, and enhance overall business performance. Also, regular monitoring and adaptation to changing market conditions are essential for sustained success.

**3.1 Read the data as an appropriate Time Series data and plot the data.**

**Printing the head-**

| | 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 |

**Creating the Time Stamps and adding to the data frame to make it a Time Series Data-**

```
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')
```

**Dataframe post setting the index-**

| | 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 |

**Null values-**

```
Rose    2
dtype: int64
```

- 2 null values are found in the dataset.
- Time series does not admit missing data. All data observations must be contiguous
- We are imputing the missing values using Spline interpolation with order 2

**Post imputing null values-**

**Printing tail of data set to check-**

| Time_Stamp | Rose |
|---|---|
| 1994-05-31 | 44.0 |
| 1994-06-30 | 45.0 |
| 1994-07-31 | 45.4 |
| 1994-08-31 | 44.6 |
| 1994-09-30 | 46.0 |
| 1994-10-31 | 51.0 |
| 1994-11-30 | 63.0 |
| 1994-12-31 | 84.0 |
| 1995-01-31 | 30.0 |
| 1995-02-28 | 39.0 |
| 1995-03-31 | 45.0 |
| 1995-04-30 | 52.0 |
| 1995-05-31 | 28.0 |
| 1995-06-30 | 40.0 |
| 1995-07-31 | 62.0 |

```
Rose    0
dtype: int64
```

- Zero null values are present now.

**Shape-**

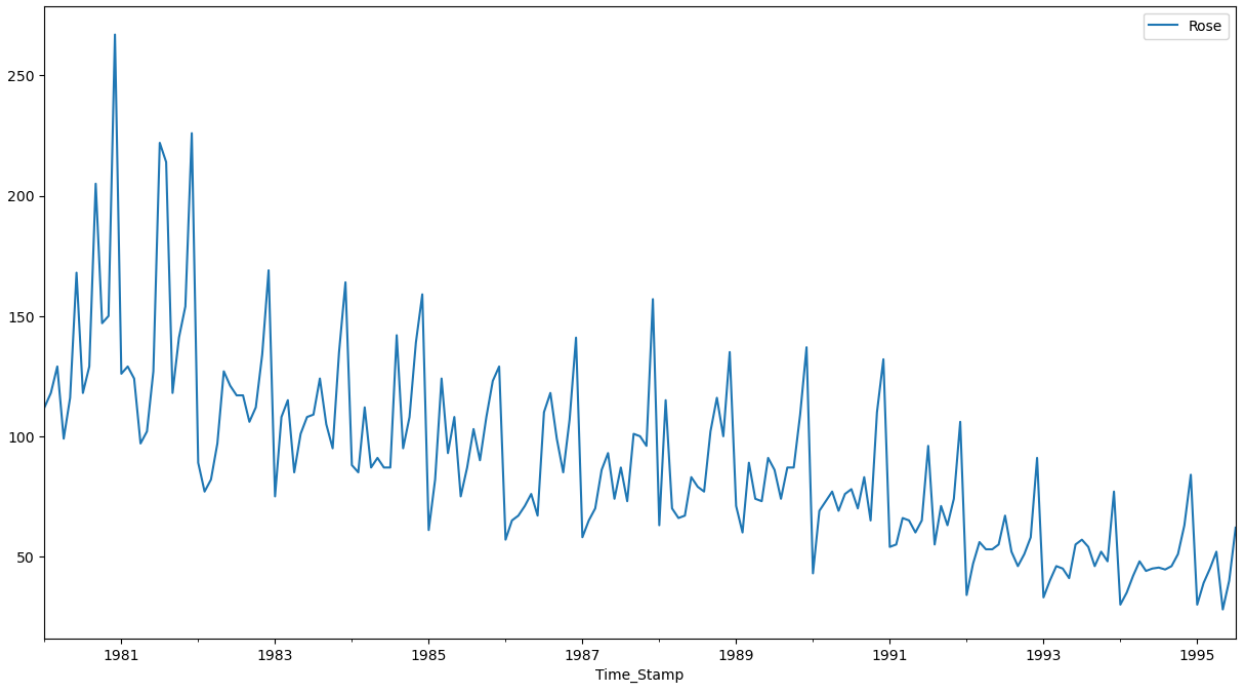
```
(187, 1)
```

**Summary-**

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Rose    187 non-null    float64
dtypes: float64(1)
memory usage: 2.9 KB
```

- The Dataframe has a total of 187 entries.
- It represents a time series dataset of 'Rose' wine sales from January 1980 to July 1995
- The data type of the 'Sparkling' column is float.

**Plotting the Time Series to understand the behaviour of the data-**




- We can see that there is a decreasing trend with a seasonal pattern associated with it.

**3.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.**

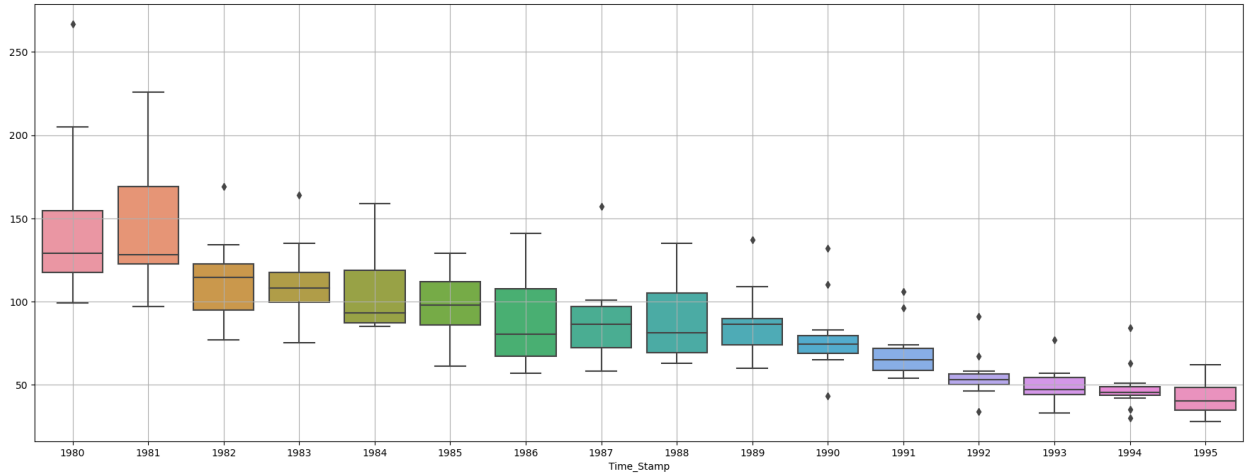
**Description-**

| | Rose |
|---|---|
| **count** | 187.000 |
| **mean** | 89.909 |
| **std** | 39.244 |
| **min** | 28.000 |
| **25%** | 62.500 |
| **50%** | 85.000 |
| **75%** | 111.000 |
| **max** | 267.000 |

- The average value of the Rose sales is 89.909
- The minimum value in the Rose wine sales is 28 and the maximum value is 267
- The first quartile, or 25th percentile indicates the value below which 25% of the data falls i.e. 62.500 in this case.
- The median, or 50th percentile represents the middle value of the Rose which is 85
- The third quartile, or 75th percentile indicates the value below which 75% of the data falls i.e. 111
- The maximum value of 267 is substantially higher than the mean and median. This indicates the presence of outliers.
- The mean value (89.909) is relatively close to the median value (85). Hence, the distribution might be symmetrical.

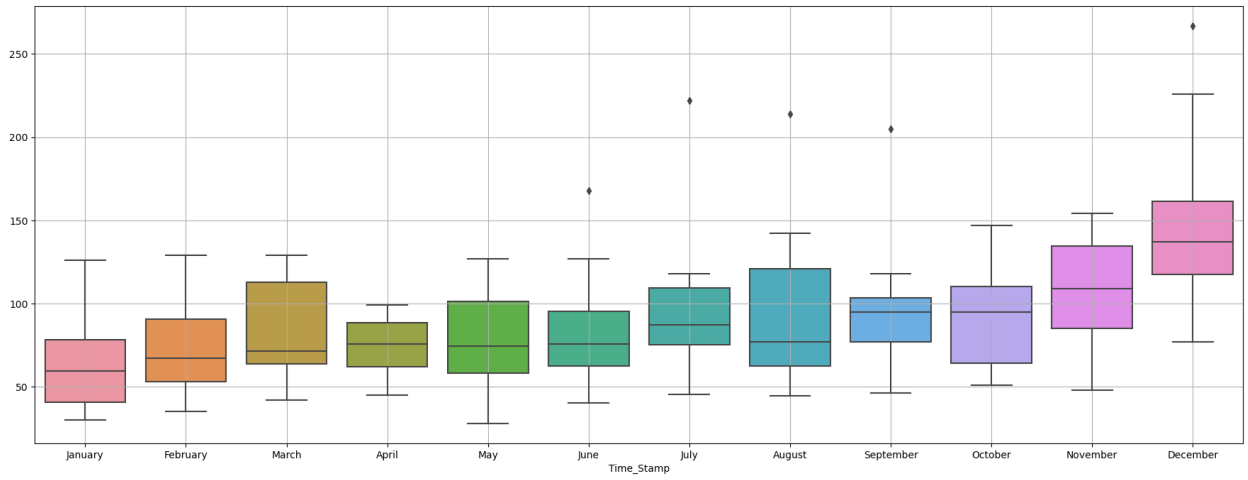
**Boxplot to understand the spread of sales across different years and within different months across years**

**Yearly boxplot-**

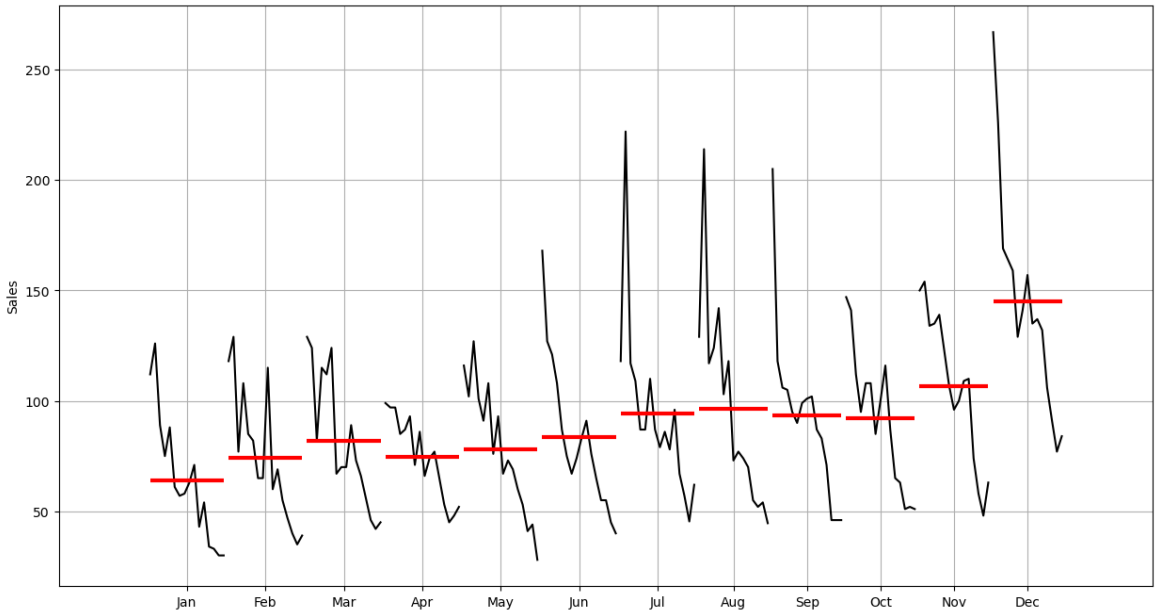



- There has been higher demand of Rose wine in 1981 comparatively
- Some of the inconsistent years for the sales include 1980, 1982, 1983, 1987, 1989, 1990, 1991, 1992, 1993, 1994.
- And the most consistent year is 1985.

**Monthly boxplot-**



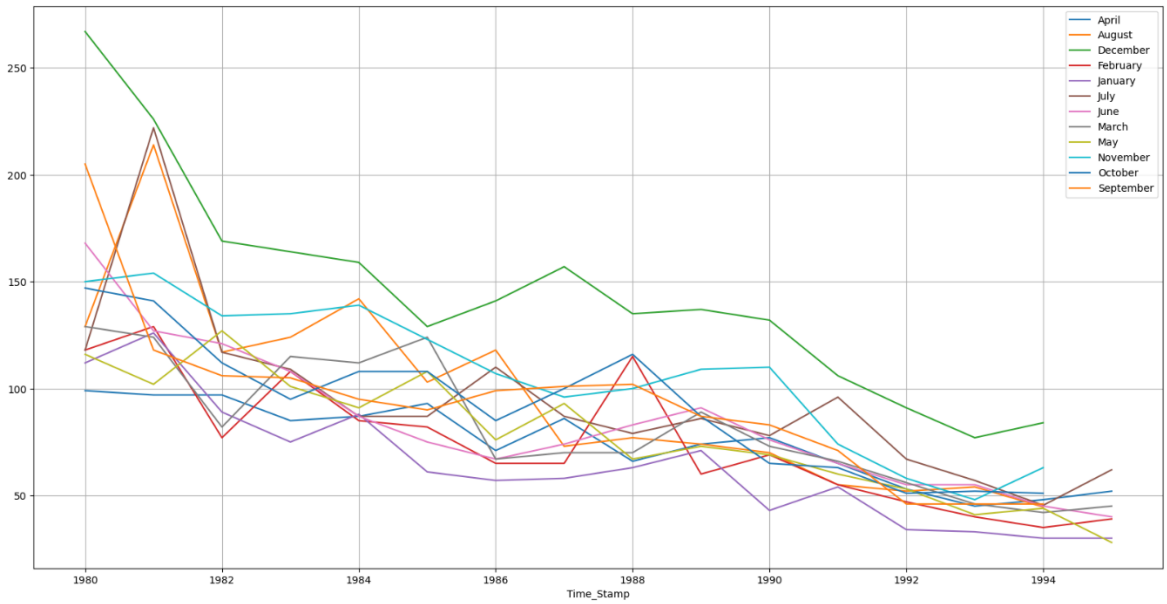

- The heaviest season has been December while the slightest season has been January.
- Season where we need to be prepared for higher fluctuations or uncertainties is June, July, August, September, December.



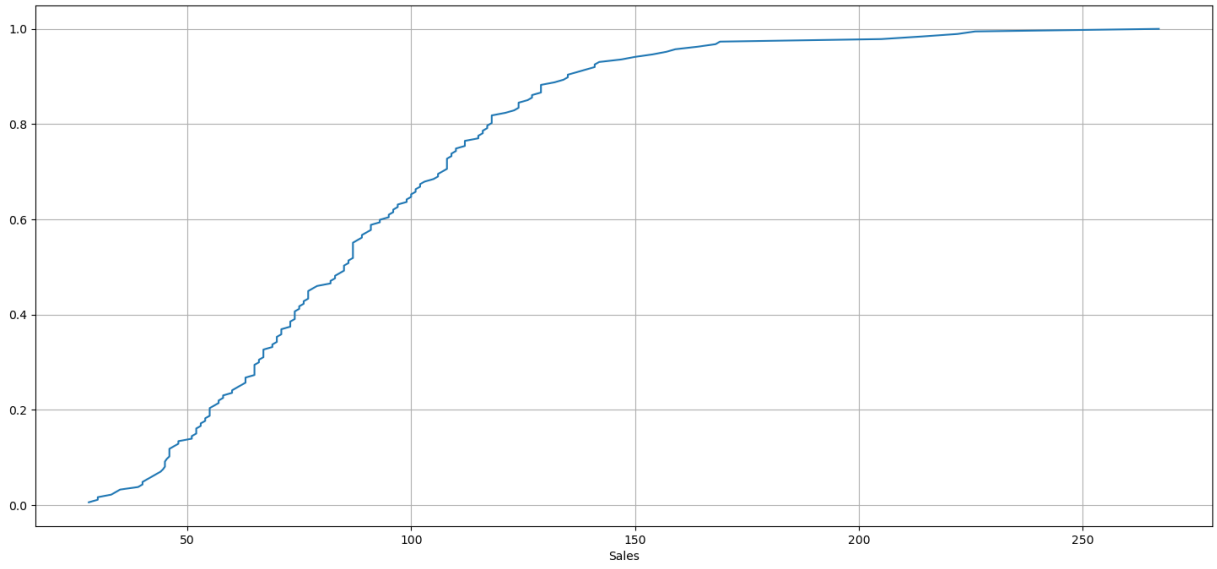

- This plot shows us the behaviour of the Time Series across various months. The red line is the median value.

**Plot of monthly sales across years-**

| Time_Stamp | April | August | December | February | January | July | June | March | May | November | October | September |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time_Stamp | | | | | | | | | | | | |
| **1980** | 99.0 | 129.0 | 267.0 | 118.0 | 112.0 | 118.0 | 168.0 | 129.0 | 116.0 | 150.0 | 147.0 | 205.0 |
| **1981** | 97.0 | 214.0 | 226.0 | 129.0 | 126.0 | 222.0 | 127.0 | 124.0 | 102.0 | 154.0 | 141.0 | 118.0 |
| **1982** | 97.0 | 117.0 | 169.0 | 77.0 | 89.0 | 117.0 | 121.0 | 82.0 | 127.0 | 134.0 | 112.0 | 106.0 |
| **1983** | 85.0 | 124.0 | 164.0 | 108.0 | 75.0 | 109.0 | 108.0 | 115.0 | 101.0 | 135.0 | 95.0 | 105.0 |
| **1984** | 87.0 | 142.0 | 159.0 | 85.0 | 88.0 | 87.0 | 87.0 | 112.0 | 91.0 | 139.0 | 108.0 | 95.0 |
| **1985** | 93.0 | 103.0 | 129.0 | 82.0 | 61.0 | 87.0 | 75.0 | 124.0 | 108.0 | 123.0 | 108.0 | 90.0 |
| **1986** | 71.0 | 118.0 | 141.0 | 65.0 | 57.0 | 110.0 | 67.0 | 67.0 | 76.0 | 107.0 | 85.0 | 99.0 |
| **1987** | 86.0 | 73.0 | 157.0 | 65.0 | 58.0 | 87.0 | 74.0 | 70.0 | 93.0 | 96.0 | 100.0 | 101.0 |
| **1988** | 66.0 | 77.0 | 135.0 | 115.0 | 63.0 | 79.0 | 83.0 | 70.0 | 67.0 | 100.0 | 116.0 | 102.0 |
| **1989** | 74.0 | 74.0 | 137.0 | 60.0 | 71.0 | 86.0 | 91.0 | 89.0 | 73.0 | 109.0 | 87.0 | 87.0 |
| **1990** | 77.0 | 70.0 | 132.0 | 69.0 | 43.0 | 78.0 | 76.0 | 73.0 | 69.0 | 110.0 | 65.0 | 83.0 |
| **1991** | 65.0 | 55.0 | 106.0 | 55.0 | 54.0 | 96.0 | 65.0 | 66.0 | 60.0 | 74.0 | 63.0 | 71.0 |
| **1992** | 53.0 | 52.0 | 91.0 | 47.0 | 34.0 | 67.0 | 55.0 | 56.0 | 53.0 | 58.0 | 51.0 | 46.0 |
| **1993** | 45.0 | 54.0 | 77.0 | 40.0 | 33.0 | 57.0 | 55.0 | 46.0 | 41.0 | 48.0 | 52.0 | 46.0 |
| **1994** | 48.0 | 44.6 | 84.0 | 35.0 | 30.0 | 45.4 | 45.0 | 42.0 | 44.0 | 63.0 | 51.0 | 46.0 |
| **1995** | 52.0 | NaN | NaN | 39.0 | 30.0 | 62.0 | 40.0 | 45.0 | 28.0 | NaN | NaN | NaN |

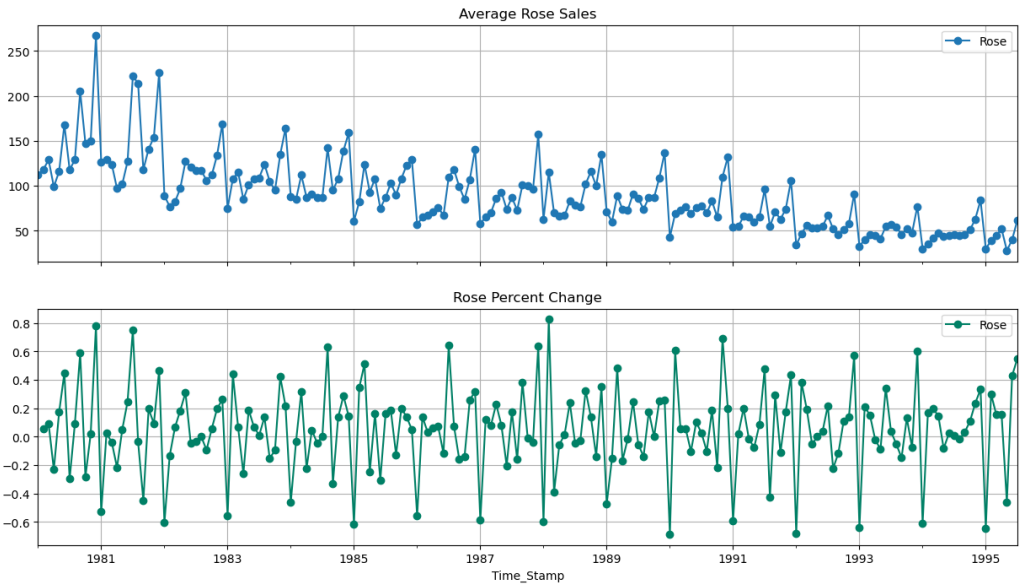



**Empirical Cumulative Distribution-**




- This particular graph tells us what percentage of data points refer to what number of Rose Sales.
- For instance, the sale between 100 and 150 units is 95% - 62% i.e. 33%
- Sale between 150 and 200 units is 97% - 95% i.e. 2%

**Average Rose Sales per month and the month on month percentage change of Sales-**

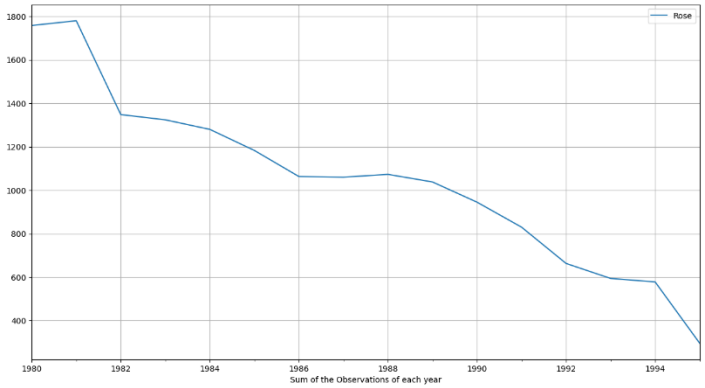


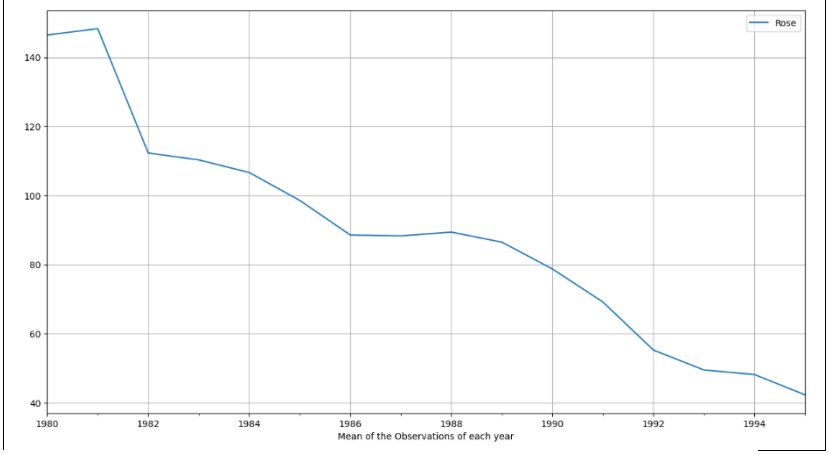

- The above two graphs tells us the Average Rose wine sales and the Percentage change of Rose Sales with respect to the time.

**Reading this monthly data into a quarterly and yearly format-**

**Yearly Plot-**

Let us try to resample or aggregate the Time Series from an annual perspective

| | Head of dataframe | Plot |
|---|---|---|
| **Sum of observations of each month** | Rose<br>Time_Stamp<br>1980-12-31 1758.0<br>1981-12-31 1780.0<br>1982-12-31 1348.0<br>1983-12-31 1324.0<br>1984-12-31 1280.0 |  |

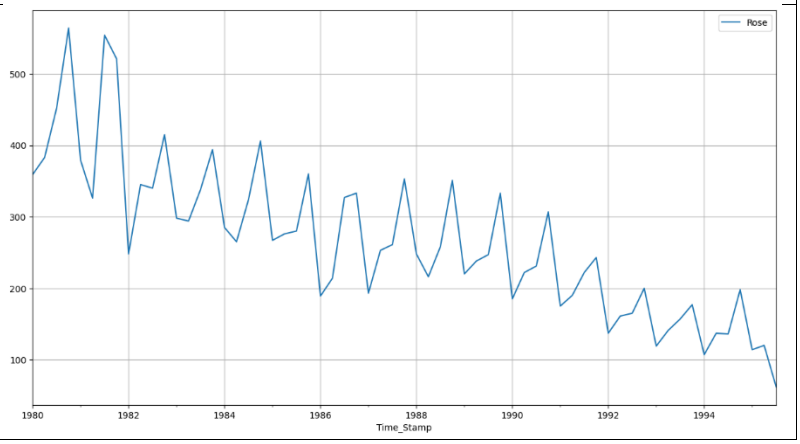
| | | |
|---|---|---|
| **Mean of the observations of the year considering each month as an observation** | |  |

| Time_Stamp | Rose |
|---|---|
| 1980-12-31 | 146.50 |
| 1981-12-31 | 148.33 |
| 1982-12-31 | 112.33 |
| 1983-12-31 | 110.33 |
| 1984-12-31 | 106.67 |

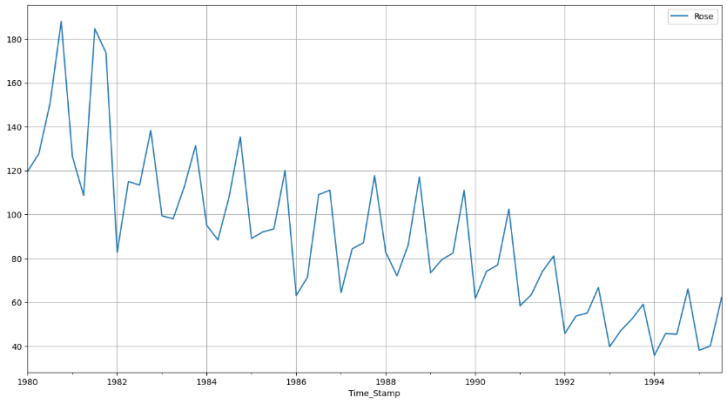


- There is a declining trend in Rose sales over the years, and then with some fluctuations, there is a significant drop in sales in 1995.
- The values of 'Rose' range from a maximum of 1780 to a minimum of 296
- And a huge decrease noticed during 1981 to 1982.

**Quarterly plot-**

| | **Head of dataframe** | **Plot** |
|---|---|---|
| **Sum of observations** | |  |

| Time_Stamp | Rose |
|---|---|
| 1980-03-31 | 359.0 |
| 1980-06-30 | 383.0 |
| 1980-09-30 | 452.0 |
| 1980-12-31 | 564.0 |
| 1981-03-31 | 379.0 |

| | | |
|---|---|---|
| **Mean of observations** | Rose<br>Time_Stamp<br>1980-03-31 119.666667<br>1980-06-30 127.666667<br>1980-09-30 150.666667<br>1980-12-31 188.000000<br>1981-03-31 126.333333 |  |

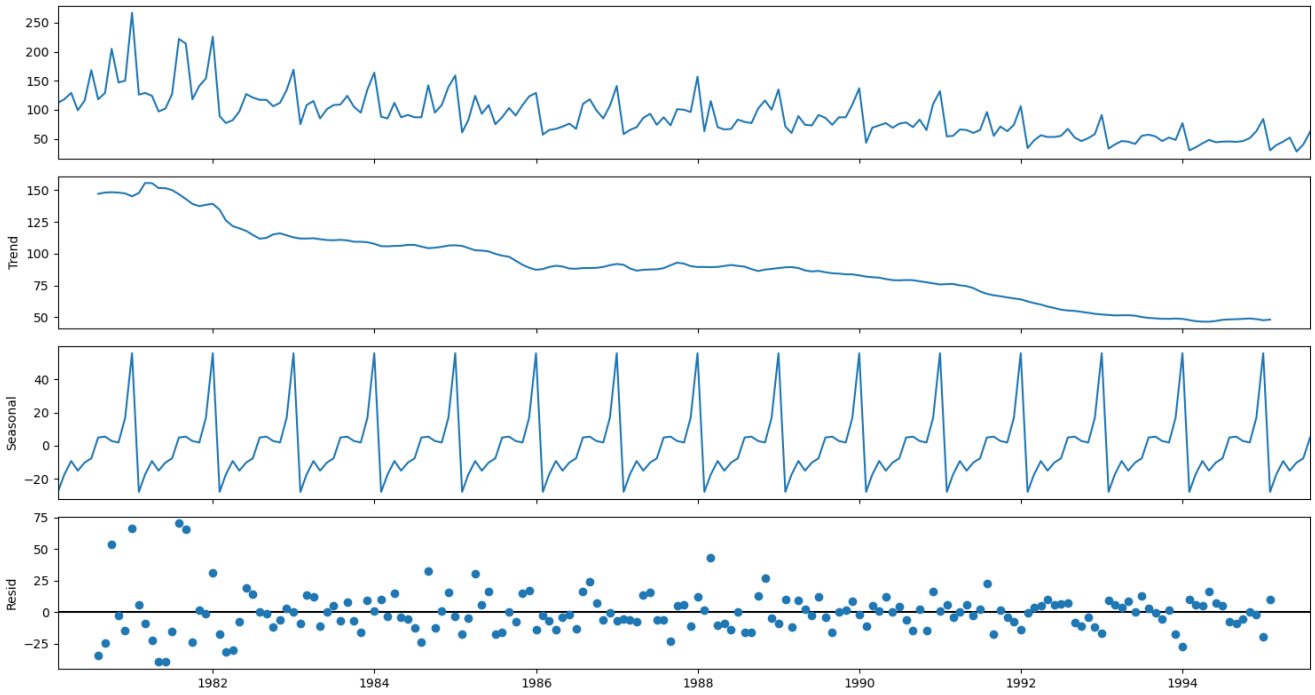
- We can notice a consistent decreasing trend in the sales with a significant drop in early 1990's

**Decomposition-**

- Decomposing the series into systematic component and irregular component
- **Systematic components**- Trend, seasonality which are interpretable and can be estimated
- **Irregular component-** error / noise associated with the series.
- Compares long term movement of series (Trend) and short-term movement (seasonality) to understand which has higher influence

1. **Additive model-**

   When seasonal variation is constant over time.

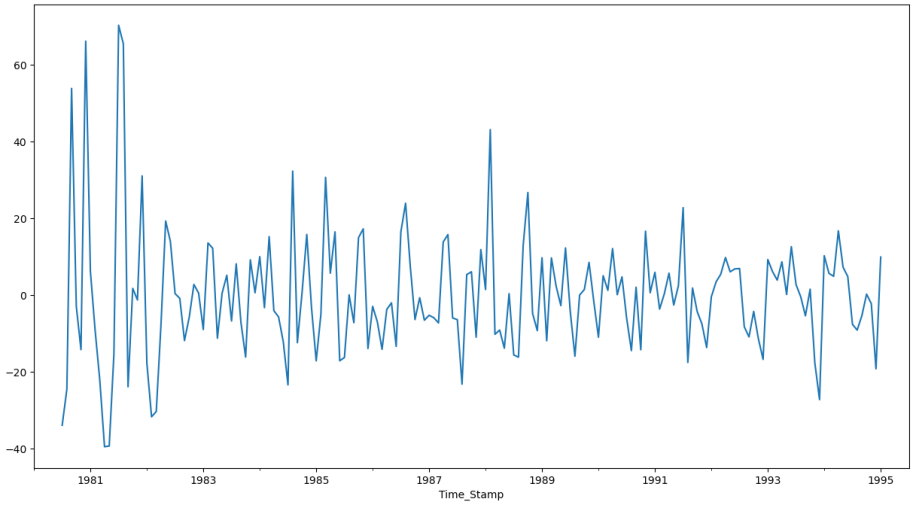



```
Trend
 Time_Stamp
1980-01-31       NaN
1980-02-29       NaN
1980-03-31       NaN
1980-04-30       NaN
1980-05-31       NaN
1980-06-30       NaN
1980-07-31    147.08
1980-08-31    148.12
1980-09-30    148.37
1980-10-31    148.08
1980-11-30    147.42
1980-12-31    145.12
Name: trend, dtype: float64
```

```
Seasonality
 Time_Stamp
1980-01-31   -27.90
1980-02-29   -17.43
1980-03-31    -9.28
1980-04-30   -15.09
1980-05-31   -10.19
1980-06-30    -7.67
1980-07-31     4.91
1980-08-31     5.43
1980-09-30     2.78
1980-10-31     1.88
1980-11-30    16.85
1980-12-31    55.72
Name: seasonal, dtype: float64
```

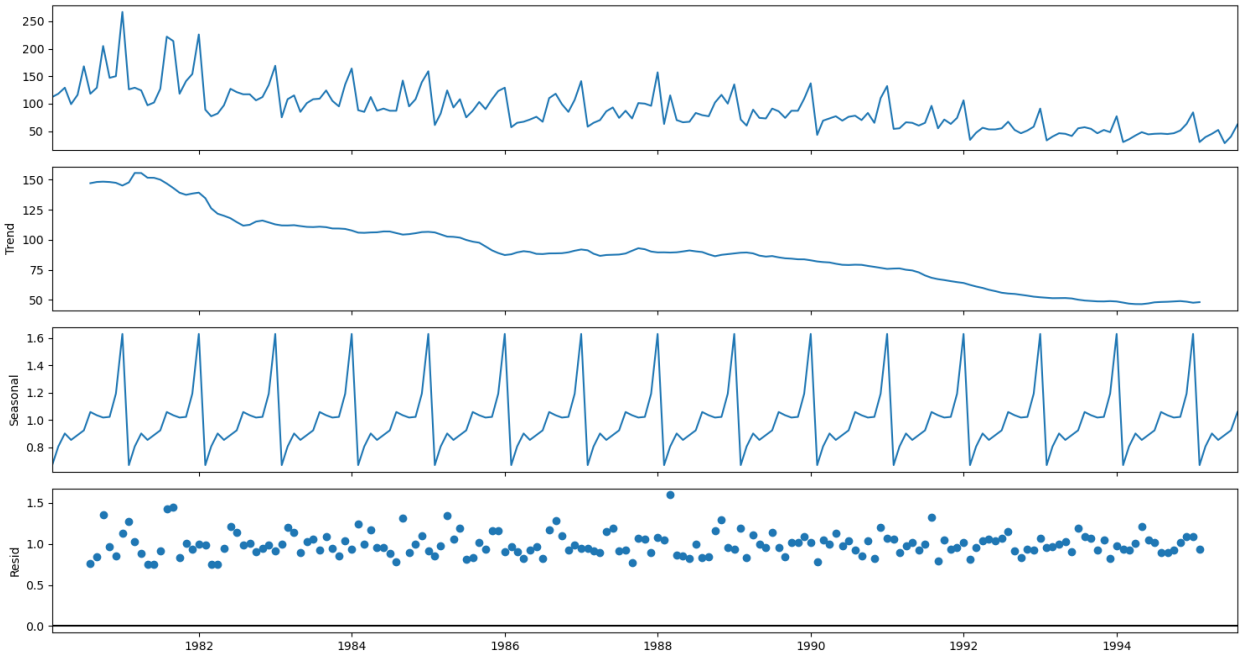
```
Residual
 Time_Stamp
1980-01-31      NaN
1980-02-29      NaN
1980-03-31      NaN
1980-04-30      NaN
1980-05-31      NaN
1980-06-30      NaN
1980-07-31   -33.99
1980-08-31   -24.56
1980-09-30    53.84
1980-10-31    -2.96
1980-11-30   -14.27
1980-12-31    66.16
Name: resid, dtype: float64
```

**Residual Plot-**



**2. Decompose the time series multiplicatively-**

When there is seasonal variation with time.

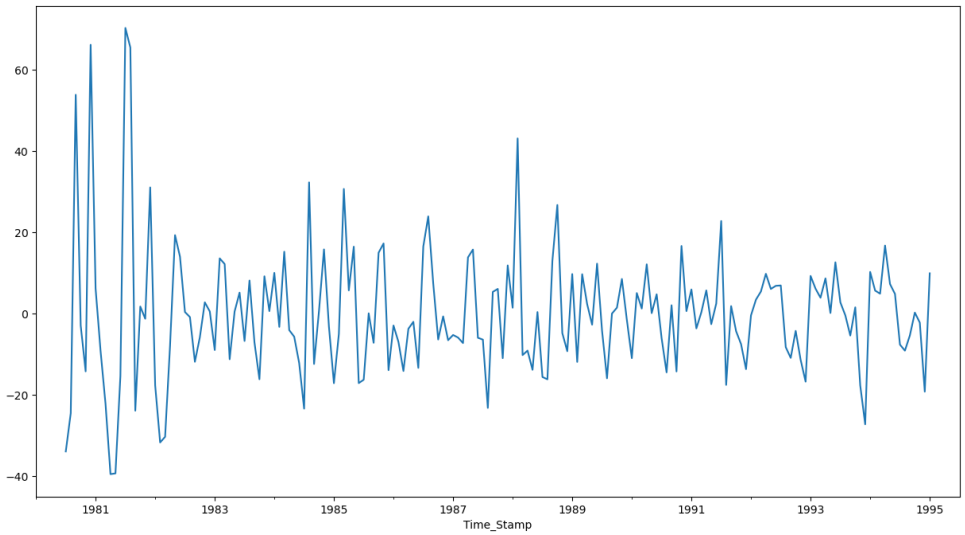



```
Trend
 Time_Stamp
1980-01-31       NaN
1980-02-29       NaN
1980-03-31       NaN
1980-04-30       NaN
1980-05-31       NaN
1980-06-30       NaN
1980-07-31    147.08
1980-08-31    148.12
1980-09-30    148.37
1980-10-31    148.08
1980-11-30    147.42
1980-12-31    145.12
Name: trend, dtype: float64
```

```
Seasonality
 Time_Stamp
1980-01-31    0.67
1980-02-29    0.81
1980-03-31    0.90
1980-04-30    0.85
1980-05-31    0.89
1980-06-30    0.92
1980-07-31    1.06
1980-08-31    1.03
1980-09-30    1.02
1980-10-31    1.02
1980-11-30    1.19
1980-12-31    1.63
Name: seasonal, dtype: float64
```

```
Residual
 Time_Stamp
1980-01-31     NaN
1980-02-29     NaN
1980-03-31     NaN
1980-04-30     NaN
1980-05-31     NaN
1980-06-30     NaN
1980-07-31    0.76
1980-08-31    0.84
1980-09-30    1.36
1980-10-31    0.97
1980-11-30    0.85
1980-12-31    1.13
Name: resid, dtype: float64
```

**Residual Plot-**



- On comparing the residual plots, we can observe that the error pattern looks the same. Hence, it seems like an additive model.

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

Training Data

| Time_Stamp | Rose |
|---|---|
| 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 |
| ... | ... |
| 1990-08-31 | 70.0 |
| 1990-09-30 | 83.0 |
| 1990-10-31 | 65.0 |
| 1990-11-30 | 110.0 |
| 1990-12-31 | 132.0 |

132 rows × 1 columns

**Test data-**

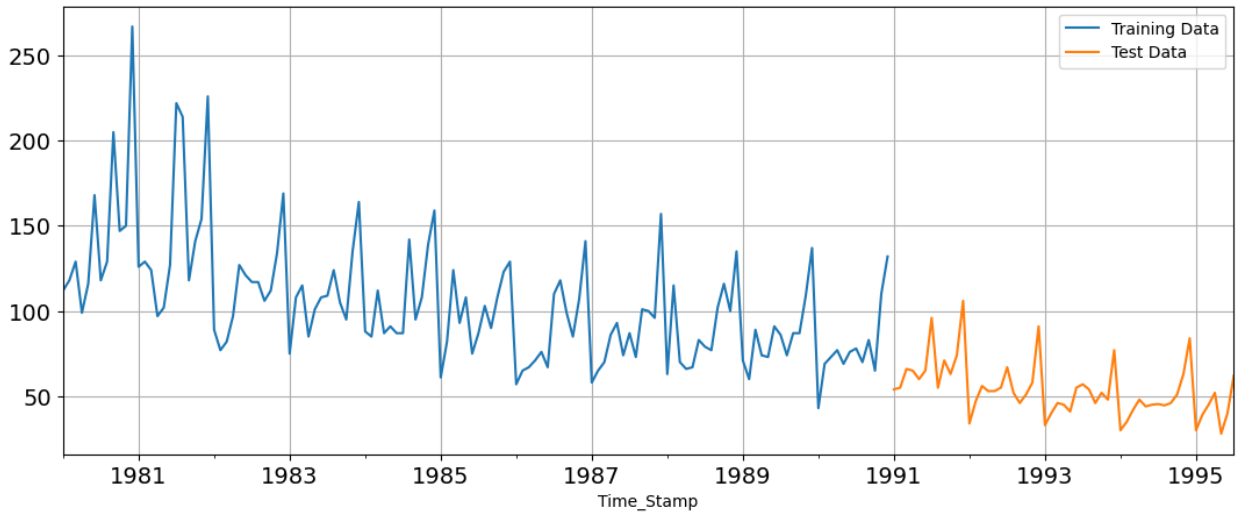
Head of Test Data

| Time_Stamp | Rose |
|---|---|
| 1991-01-31 | 54.0 |
| 1991-02-28 | 55.0 |
| 1991-03-31 | 66.0 |
| 1991-04-30 | 65.0 |
| 1991-05-31 | 60.0 |

Tail of Test Data

| Time_Stamp | Rose |
|---|---|
| 1995-03-31 | 45.0 |
| 1995-04-30 | 52.0 |
| 1995-05-31 | 28.0 |
| 1995-06-30 | 40.0 |
| 1995-07-31 | 62.0 |

**Plot –**




**3.4 Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE.**

**RMSE (Root Mean Squared Error)**

- It is a commonly used metric to evaluate the accuracy of a time series model.
- It measures the average magnitude of the errors between predicted and observed values.
- For time series models, RMSE is used to assess how well the model forecasts future values over time.
- A lower RMSE value indicates better model performance, as it denotes that the model's predictions are relatively closer to the actual values.

We are going to build models on the training data and evaluate their performance on test data based on the RMSE values.

**Model 1: Linear Regression**

Not a time series model, but we are breaking it into time series specific.
Train is from 1$^{st}$ - 132$^{nd}$ value
Test is from 133$^{rd}$ value.

```
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, 3
4, 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, 121, 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, 157,
158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187]
```

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.

**Train data-**

```
First few rows of Training Data
              Rose  time
Time_Stamp
1980-01-31  112.0     1
1980-02-29  118.0     2
1980-03-31  129.0     3
1980-04-30   99.0     4
1980-05-31  116.0     5

Last few rows of Training Data
              Rose  time
Time_Stamp
1990-08-31   70.0   128
1990-09-30   83.0   129
1990-10-31   65.0   130
1990-11-30  110.0   131
1990-12-31  132.0   132
```

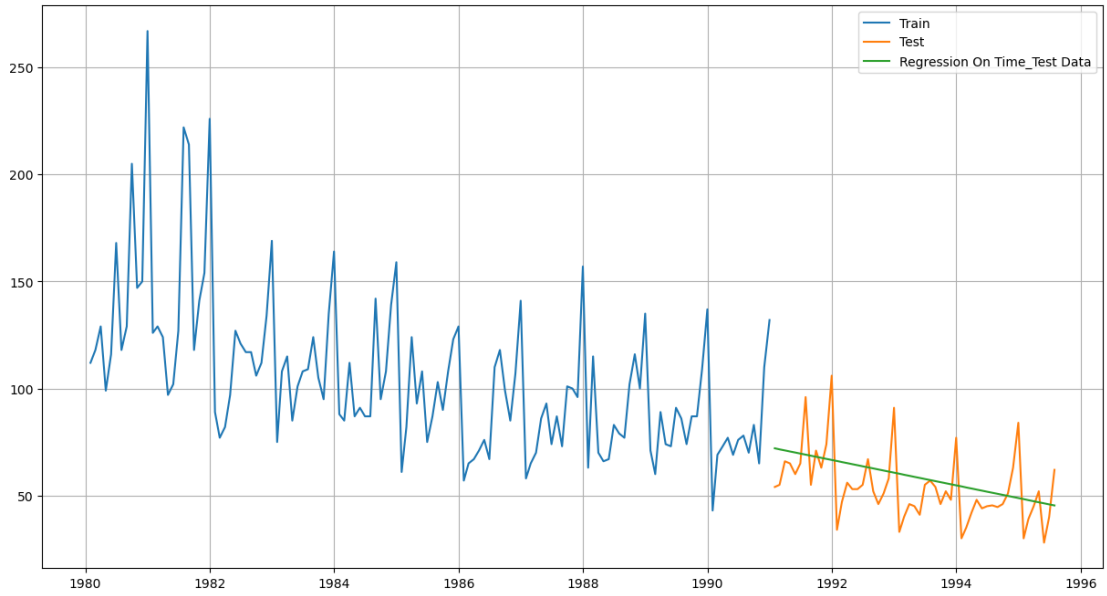
**Test data-**

```
First few rows of Test Data
             Rose  time
Time_Stamp
1991-01-31  54.0   133
1991-02-28  55.0   134
1991-03-31  66.0   135
1991-04-30  65.0   136
1991-05-31  60.0   137

Last few rows of Test Data
             Rose  time
Time_Stamp
1995-03-31  45.0   183
1995-04-30  52.0   184
1995-05-31  28.0   185
1995-06-30  40.0   186
1995-07-31  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.

```
▾ LinearRegression
LinearRegression()
```

**Plotting train, test and predictions on test-**




Linear regression can handle trend to an extent but not seasonality.

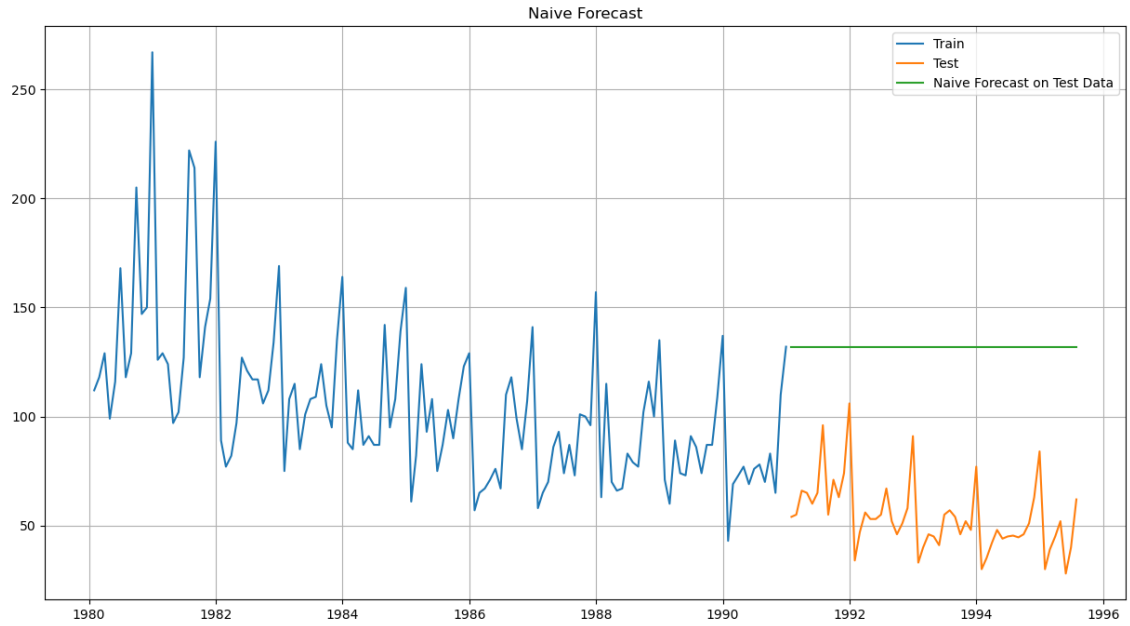
**Model Evaluation-**

For RegressionOnTime forecast on the Test Data, RMSE is **15.28**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 15.275687 |

**Model 2: Naive Approach:**

- Uses the last observed value
- Ignores trend, seasonality.




**Model evaluation-**

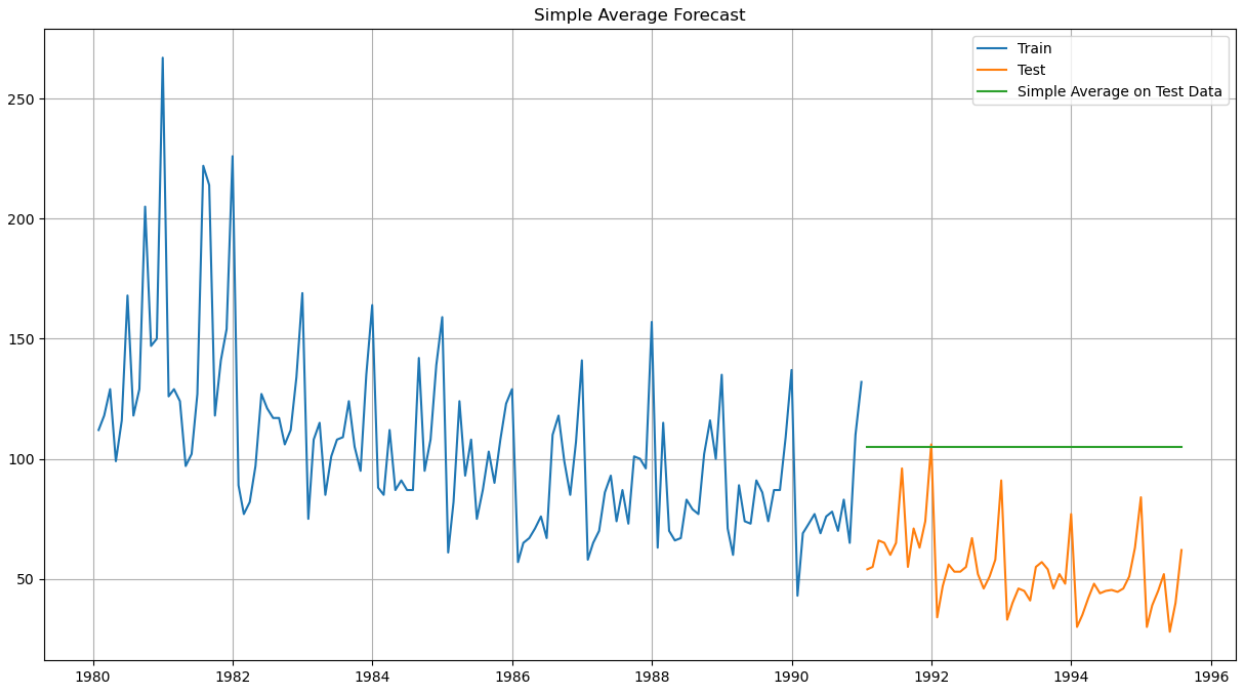
For RegressionOnTime forecast on the Test Data, RMSE is **79.739**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 15.275687 |
| **NaiveModel** | 79.738587 |

**Method 3: Simple Average**

For this particular simple average method, we will forecast by using the average of the training values.

| Time_Stamp | Rose | mean_forecast |
|---|---|---|
| **1991-01-31** | 54.0 | 104.939394 |
| **1991-02-28** | 55.0 | 104.939394 |
| **1991-03-31** | 66.0 | 104.939394 |
| **1991-04-30** | 65.0 | 104.939394 |
| **1991-05-31** | 60.0 | 104.939394 |




**Model evaluation-**

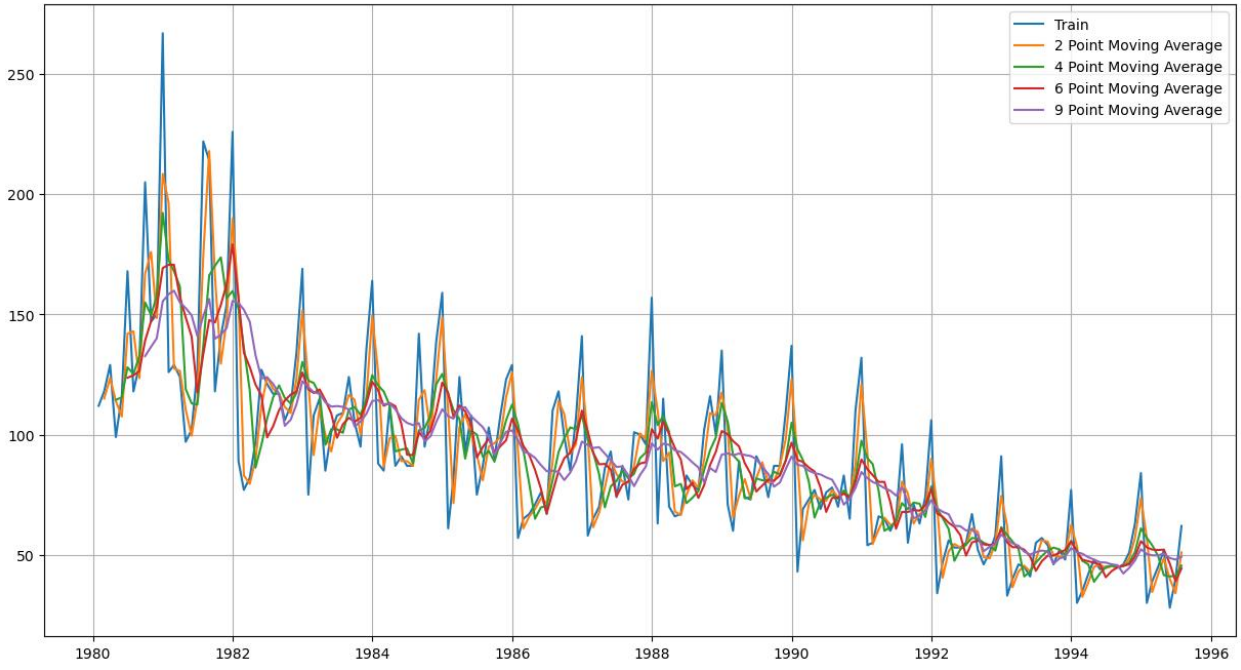
For Simple Average forecast on the Test Data, RMSE is **53.481**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 15.275687 |
| **NaiveModel** | 79.738587 |
| **SimpleAverageModel** | 53.480911 |

**Method 4: Moving Average(MA)-**

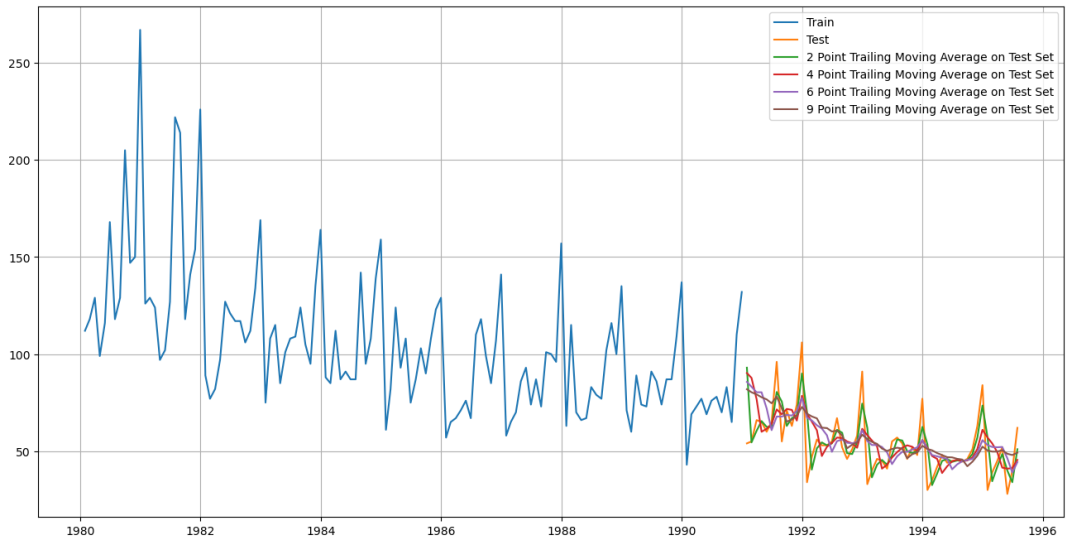
- For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.
- For Moving Average, we are going to average over the entire data.
- Here, we are considering 2 month, 4 months, 6 months and 9 months moving average

| Time_Stamp | Rose | Trailing_2 | Trailing_4 | Trailing_6 | Trailing_9 |
|---|---|---|---|---|---|
| **1980-01-31** | 112.0 | NaN | NaN | NaN | NaN |
| **1980-02-29** | 118.0 | 115.0 | NaN | NaN | NaN |
| **1980-03-31** | 129.0 | 123.5 | NaN | NaN | NaN |
| **1980-04-30** | 99.0 | 114.0 | 114.5 | NaN | NaN |
| **1980-05-31** | 116.0 | 107.5 | 115.5 | NaN | NaN |




- From the graph, we can observe that the 2 point is most closest to actuals.

**Let us split the data into train and test and plot this Time Series-**



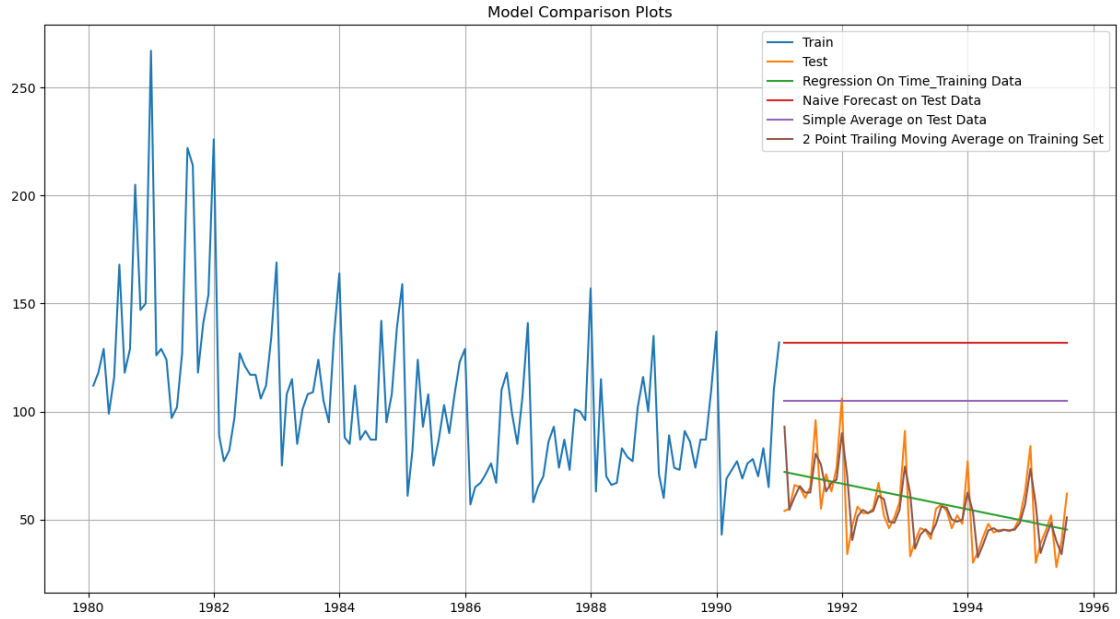

**Model Evaluation-**

For 2 point Moving Average Model forecast on the Training Data, RMSE is 11.530
For 4 point Moving Average Model forecast on the Training Data, RMSE is 14.457
For 6 point Moving Average Model forecast on the Training Data, RMSE is 14.571
For 9 point Moving Average Model forecast on the Training Data, RMSE is 14.731

- On comparing actuals and predictions, we can see least amount of error in 2 months as expected.

**Consolidated plots of all Models-**

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |




**Method 5: Exponential Smoothing methods-**

- Exponential smoothing methods consist of flattening time series data.
- Exponential smoothing averages or exponentially weighted moving averages consist of forecast based on previous periods data with exponentially declining influence on the older observations.
- Exponential smoothing methods consist of special case exponential moving with notation ETS (Error, Trend, Seasonality) where each can be none(N), additive (N), additive damped (Ad), Multiplicative (M) or multiplicative damped (Md).
- One or more parameters control how fast the weights decay.
- These parameters have values between 0 and 1.

**5.1 SES - ETS(A, N, N) - Simple Exponential Smoothing with additive errors-**

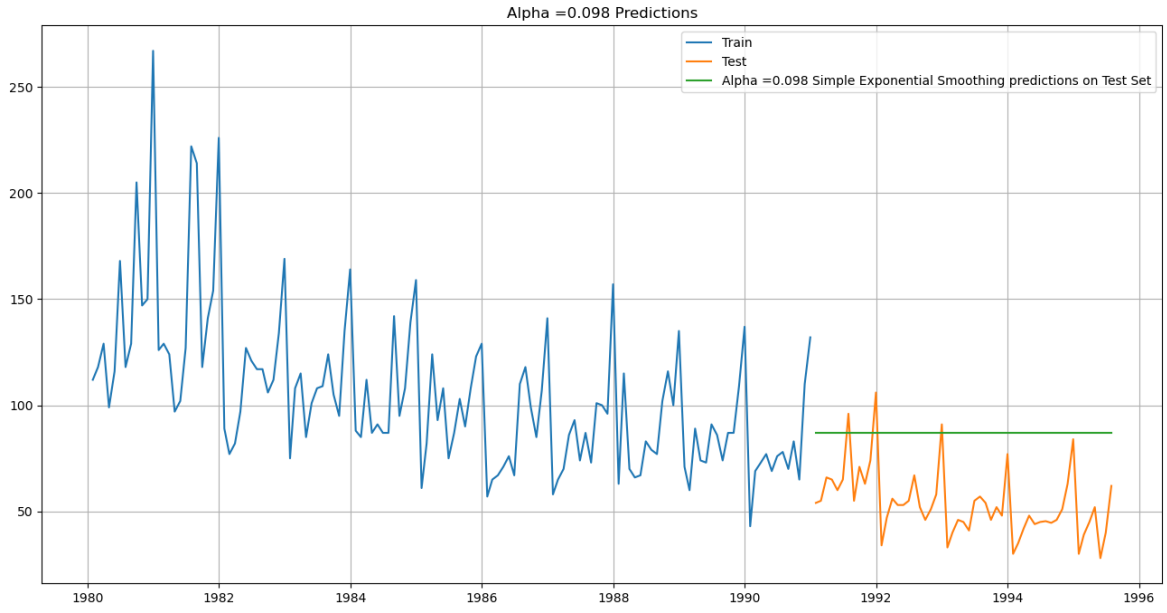
- The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing (SES).
- This method is suitable for forecasting data with no clear trend or seasonal pattern i.e. it only handles level.
- Parameter alpha is called the smoothing constant and its value lies between 0 and 1.
- Since the model uses only one smoothing constant, it is called Single Exponential Smoothing.
- SimpleExpSmoothing class must be instantiated and passed the training data.
- The fit() function is then called providing the fit configuration, the alpha value, smoothing level. If this is omitted or set to None, the model will automatically optimize the value.

**Parameters-**

```
{'smoothing_level': 0.09874983698117956,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 134.38702481818487,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- Trend and seasonality is not considered hence its nan.

| Time_Stamp | Rose | predict |
|---|---|---|
| 1991-01-31 | 54.0 | 87.104997 |
| 1991-02-28 | 55.0 | 87.104997 |
| 1991-03-31 | 66.0 | 87.104997 |
| 1991-04-30 | 65.0 | 87.104997 |
| 1991-05-31 | 60.0 | 87.104997 |



**Model Evaluation-**

For Alpha =0.098, Simple Exponential Smoothing Model forecast on the Test Data, RMSE is **36.817**

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |

- The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.
- Here, Alpha is more closer to 0. Hence, the data is falling beyond actuals.

**5.2 Double Exponential Smoothing (Holt's Model)**

- One of the drawbacks of the simple exponential smoothing is that the model does not do well in the presence of the trend.
- Applicable when data has Trend but no seasonality.
- Level is the local mean.
- This model is an extension of SES known as Double Exponential model which estimates two smoothing parameters-
    iii. One smoothing parameter α corresponds to the level series
    iv. A second smoothing parameter β corresponds to the trend series.

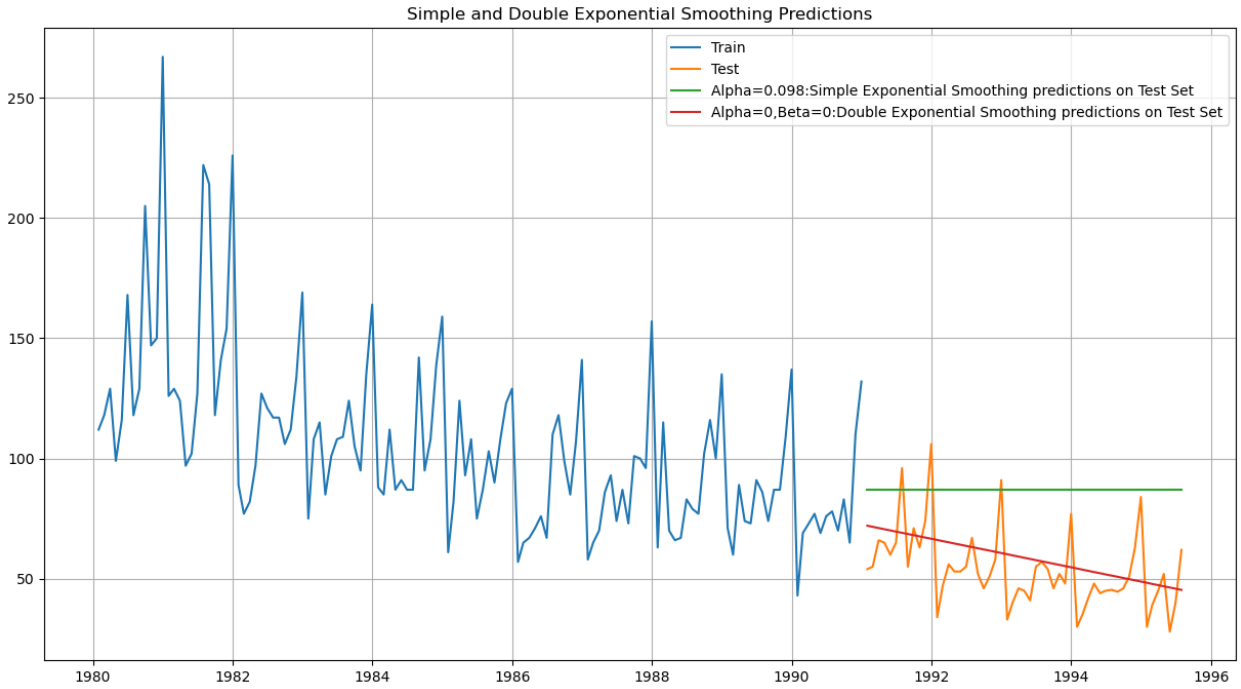
**Parameters-**

```
Holt model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 1.4901161193847656e-08, 'smoothing_trend': 1.6610391146660035e-10, 'smoothing_seasonal': nan, 'damping_tren
d': nan, 'initial_level': 137.81553690867275, 'initial_trend': -0.4943781897068274, 'initial_seasons': array([], dtype=float6
4), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

- Alpha is more closer to 0. Hence, the data is falling farther from actuals.
- Beta closer to 0 which means old past trends are relevant to the forecast.
- Seasonality is not considered hence its nan.

**Predictions-**

```
1991-01-31    72.063238
1991-02-28    71.568859
1991-03-31    71.074481
1991-04-30    70.580103
1991-05-31    70.085725
1991-06-30    69.591347
1991-07-31    69.096969
1991-08-31    68.602590
1991-09-30    68.108212
1991-10-31    67.613834
1991-11-30    67.119456
1991-12-31    66.625078
1992-01-31    66.130699
1992-02-29    65.636321
1992-03-31    65.141943
1992-04-30    64.647565
1992-05-31    64.153187
1992-06-30    63.658808
1992-07-31    63.164430
1992-08-31    62.670052
1992-09-30    62.175674
1992-10-31    61.681296
1992-11-30    61.186918
1992-12-31    60.692539
1993-01-31    60.198161
1993-02-28    59.703783
1993-03-31    59.209405
1993-04-30    58.715027
1993-05-31    58.220648
1993-06-30    57.726270
1993-07-31    57.231892
1993-08-31    56.737514
1993-09-30    56.243136
1993-10-31    55.748757
1993-11-30    55.254379
1993-12-31    54.760001
1994-01-31    54.265623
1994-02-28    53.771245
1994-03-31    53.276866
1994-04-30    52.782488
1994-05-31    52.288110
1994-06-30    51.793732
1994-07-31    51.299354
1994-08-31    50.804976
1994-09-30    50.310597
1994-10-31    49.816219
1994-11-30    49.321841
1994-12-31    48.827463
1995-01-31    48.333085
1995-02-28    47.838706
1995-03-31    47.344328
1995-04-30    46.849950
1995-05-31    46.355572
1995-06-30    45.861194
1995-07-31    45.366815
Freq: M, dtype: float64
```




- We see that the double exponential smoothing is picking up the trend component along with the level component as well.

**Model evaluation-**

DES RMSE: **15.27567518571869**

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |
| Alpha=0,Beta=0:DES | 15.275675 |

**5.3 Triple Exponential Smoothing (Holt - Winter's Model)-**

**Holt-Winters - ETS(A, A, A) - Holt Winter's linear method with additive errors**

- Three parameters $\alpha$, $\beta$ and $\gamma$ are estimated in this model. Level, Trend and Seasonality are accounted for in this model.

**Parameters-**

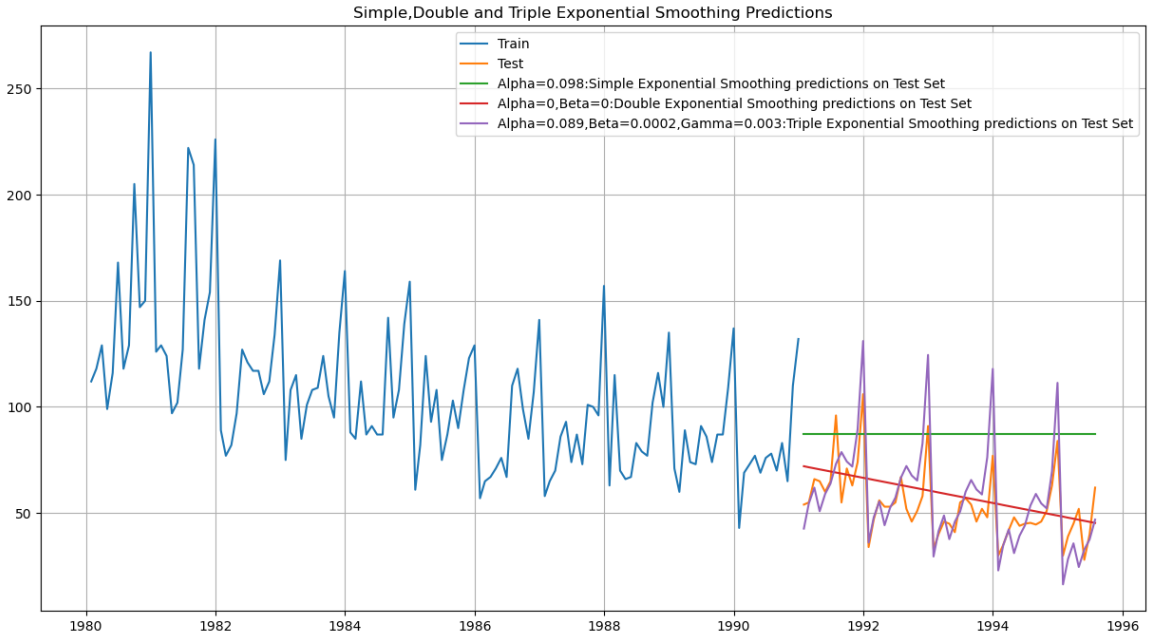
```
Holt Winters model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 0.08954054664605082, 'smoothing_trend': 0.0002400108693915795, 'smoothing_seasonal': 0.003466872515750747,
'damping_trend': nan, 'initial_level': 146.5570157826235, 'initial_trend': -0.547196983509005, 'initial_seasons': array([-31.17
478463, -18.74839869, -10.76961776, -21.36741017,
       -12.63775539,  -7.27430333,   2.61279801,   8.69603625,
         4.79381122,   2.96110122,  21.05738849,  63.18279918]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

- Alpha is more closer to 0. Hence, the data is falling farther from actuals.
- Beta closer to 0 which means old past trends are relevant to the forecast.
- Gamma closer to 0, old pasts seasons are relevant for forecast.

**Forecasting using this model for the duration of the test set-**

| | |
|---|---|
| 1991-01-31 | 42.684928 |
| 1991-02-28 | 54.564005 |
| 1991-03-31 | 61.995209 |
| 1991-04-30 | 50.852018 |
| 1991-05-31 | 59.034271 |
| 1991-06-30 | 63.850901 |
| 1991-07-31 | 73.190805 |
| 1991-08-31 | 78.724624 |
| 1991-09-30 | 74.276280 |
| 1991-10-31 | 71.895000 |
| 1991-11-30 | 89.444365 |
| 1991-12-31 | 131.042724 |
| 1992-01-31 | 36.119272 |
| 1992-02-29 | 47.998349 |
| 1992-03-31 | 55.429553 |
| 1992-04-30 | 44.286362 |
| 1992-05-31 | 52.468615 |
| 1992-06-30 | 57.285245 |
| 1992-07-31 | 66.625149 |
| 1992-08-31 | 72.158968 |
| 1992-09-30 | 67.710624 |
| 1992-10-31 | 65.329344 |
| 1992-11-30 | 82.878709 |
| 1992-12-31 | 124.477068 |
| 1993-01-31 | 29.553616 |
| 1993-02-28 | 41.432693 |
| 1993-03-31 | 48.863898 |
| 1993-04-30 | 37.720706 |
| 1993-05-31 | 45.902959 |
| 1993-06-30 | 50.719589 |
| 1993-07-31 | 60.059493 |
| 1993-08-31 | 65.593312 |
| 1993-09-30 | 61.144968 |
| 1993-10-31 | 58.763688 |
| 1993-11-30 | 76.313053 |
| 1993-12-31 | 117.911412 |

```
1994-01-31     22.987961
1994-02-28     34.867037
1994-03-31     42.298242
1994-04-30     31.155050
1994-05-31     39.337303
1994-06-30     44.153933
1994-07-31     53.493837

1994-08-31     59.027656
1994-09-30     54.579313
1994-10-31     52.198032
1994-11-30     69.747397
1994-12-31    111.345756
1995-01-31     16.422305
1995-02-28     28.301381
1995-03-31     35.732586
1995-04-30     24.589394
1995-05-31     32.771647
1995-06-30     37.588277
1995-07-31     46.928181
Freq: M, dtype: float64
```




**Model evaluation-**

TES RMSE: **14.267868116434915**

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |
| Alpha=0,Beta=0:DES | 15.275675 |
| Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive | 14.267868 |

**Holt-Winters - ETS(A, A, M) - Holt Winter's linear method with multiplicative seasonality-**

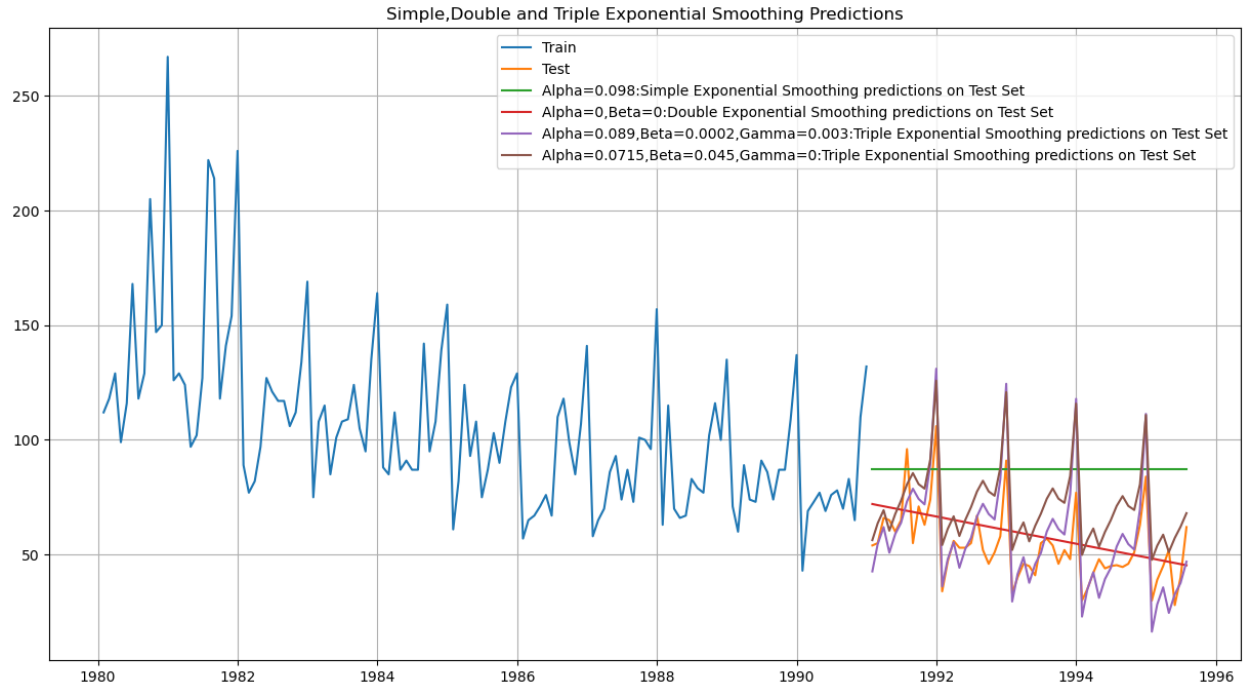
**Parameters-**

```
Holt Winters model Exponential Smoothing Estimated Parameters:

{'smoothing_level': 0.0715106306609405, 'smoothing_trend': 0.04529179757535142, 'smoothing_seasonal': 7.244325029450242e-05, 'd
amping_trend': nan, 'initial_level': 130.40839142502193, 'initial_trend': -0.77985743179386, 'initial_seasons': array([0.862189
96, 0.977675  , 1.0687727 , 0.93403881, 1.050625  ,
       1.14410977, 1.25836944, 1.33937772, 1.26778766, 1.24131254,
       1.44724625, 1.99553681]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

- Alpha is more closer to 0. Hence, the data is falling farther from actuals.
- Beta closer to 0  which means old past trends are relevant to the forecast.
- Gamma closer to 0, old pasts seasons are relevant for forecast.

**Predictions-**

```
1991-01-31     56.321655
1991-02-28     63.664690
1991-03-31     69.374024
1991-04-30     60.435528
1991-05-31     67.758341
1991-06-30     73.546478
1991-07-31     80.630117
1991-08-31     85.541323
1991-09-30     80.707713
1991-10-31     78.764555
1991-11-30     91.531230
1991-12-31    125.788433
1992-01-31     54.168902
1992-02-29     61.223492
1992-03-31     66.705377
1992-04-30     58.103246
1992-05-31     65.135026
1992-06-30     70.689855
1992-07-31     77.488188
1992-08-31     82.197159
1992-09-30     77.542202
1992-10-31     75.665128
1992-11-30     87.917577
1992-12-31    120.805914
1993-01-31     52.016149
1993-02-28     58.782294
1993-03-31     64.036730
1993-04-30     55.770964
1993-05-31     62.511711
1993-06-30     67.833232
1993-07-31     74.346259
1993-08-31     78.852995
1993-09-30     74.376691
1993-10-31     72.565700
1993-11-30     84.303925
1993-12-31    115.823395

1994-01-31     49.863396
1994-02-28     56.341097
1994-03-31     61.368082
1994-04-30     53.438682
1994-05-31     59.888397
1994-06-30     64.976609
1994-07-31     71.204330


1994-08-31     75.508831
1994-09-30     71.211180
1994-10-31     69.466273
1994-11-30     80.690272
1994-12-31    110.840875
1995-01-31     47.710643
1995-02-28     53.899899
1995-03-31     58.699435
1995-04-30     51.106400
1995-05-31     57.265082
1995-06-30     62.119986
1995-07-31     68.062402
Freq: M, dtype: float64
```




**Model accuracy-**

TES_am RMSE: **20.184415824578775**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 15.275687 |
| **NaiveModel** | 79.738587 |
| **SimpleAverageModel** | 53.480911 |
| **2pointTrailingMovingAverage** | 11.529756 |
| **4pointTrailingMovingAverage** | 14.456548 |
| **6pointTrailingMovingAverage** | 14.570933 |
| **9pointTrailingMovingAverage** | 14.731357 |
| **Alpha=0.098,SimpleExponentialSmoothing** | 36.816981 |
| **Alpha=0,Beta=0:DES** | 15.275675 |
| **Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive** | 14.267868 |
| **Alpha=0.0715,Beta=0.045,Gamma=0:TES Multiplicative** | 20.184416 |

**3.5 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.**
**Note: Stationarity should be checked at alpha = 0.05.**

- A Time Series is considered to be stationary when statistical properties such as the variance and (auto) correlation are constant over time.
- Stationary Time Series allows us to think of the statistical properties of the time series as not changing in time, which enables us to build appropriate statistical models for forecasting based on past data.
- Stationarity means that the autocorrelation of lag 'k' depends on k, but not on time t.

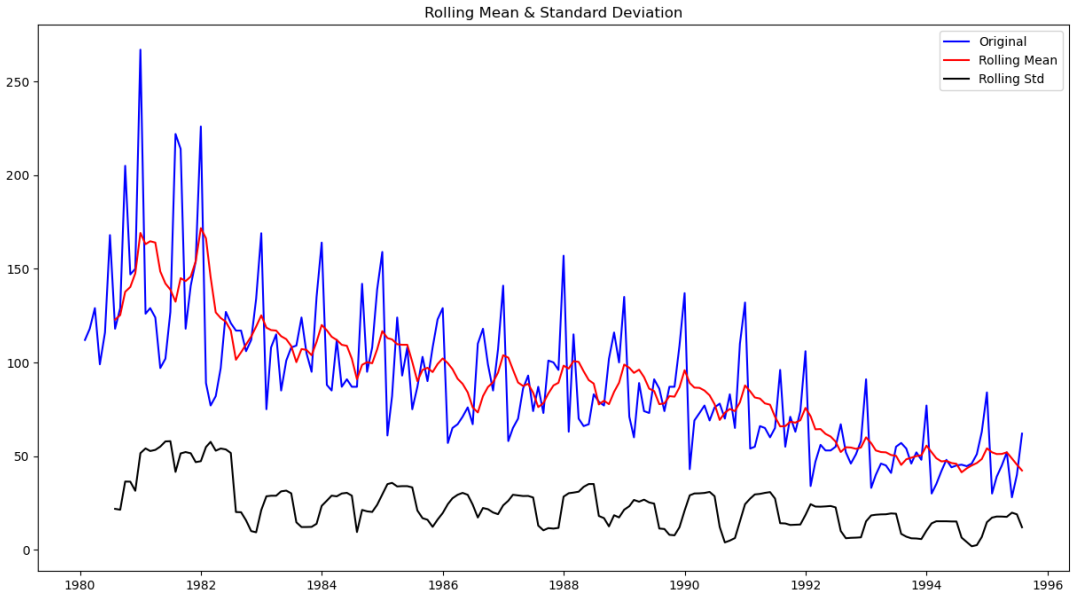
**How to check for Stationarity-**

**Dickey-Fuller Test -**

Dicky Fuller Test on the timeseries is run to check for stationarity of data.

**Null Hypothesis :** Time Series is non-stationary.
**Alternate Hypothesis :** Time Series is stationary.

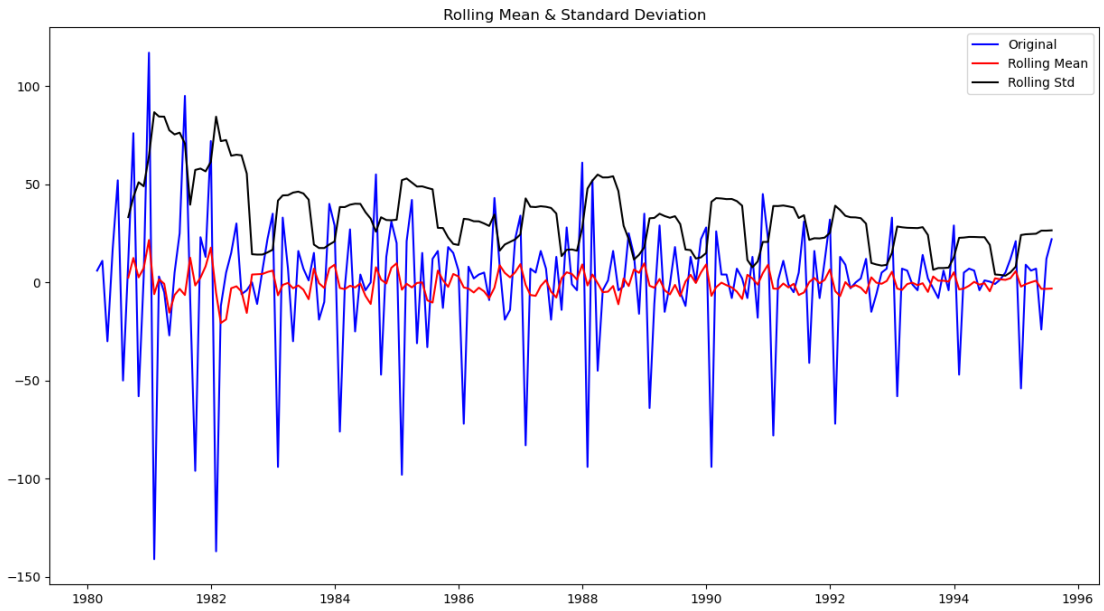
So, ideally if p-value < 0.05 then null hypothesis: TS is non-stationary is rejected else the TS is non-stationary is failed to be rejected .




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.873548
p-value                          0.344605
#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
```

- Here, p-value > 0.05. Hence, 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.

**Differencing-**




```
Results of Dickey-Fuller Test:
Test Statistic                -8.044173e+00
p-value                        1.813221e-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
```

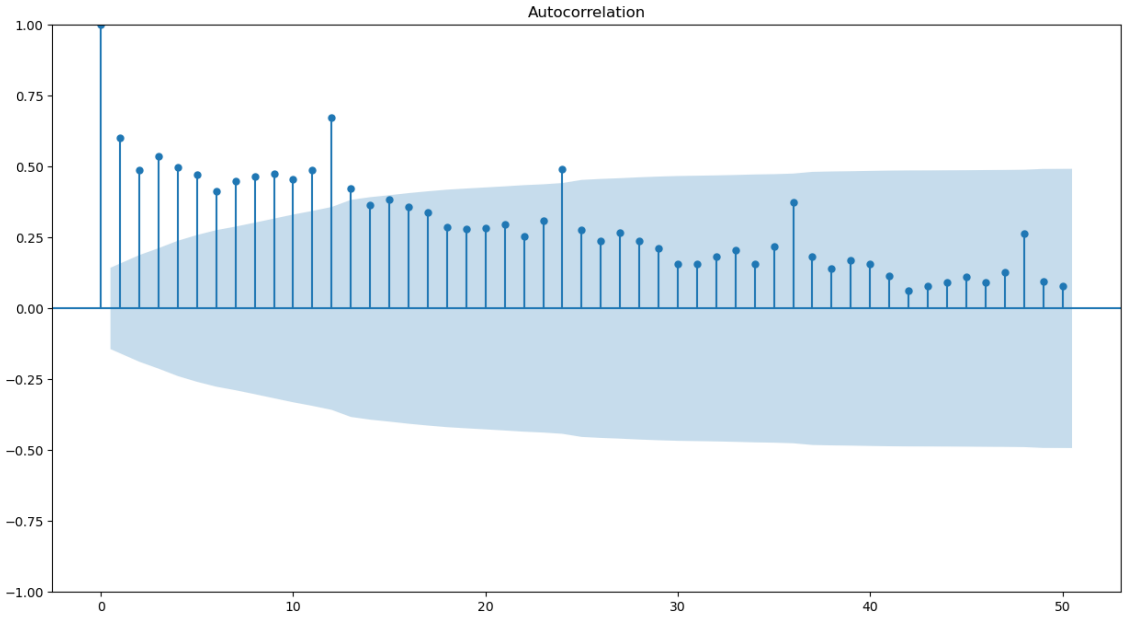
- We see that at alpha = 0.05 the Time Series is indeed stationary. (p-value <0.05)

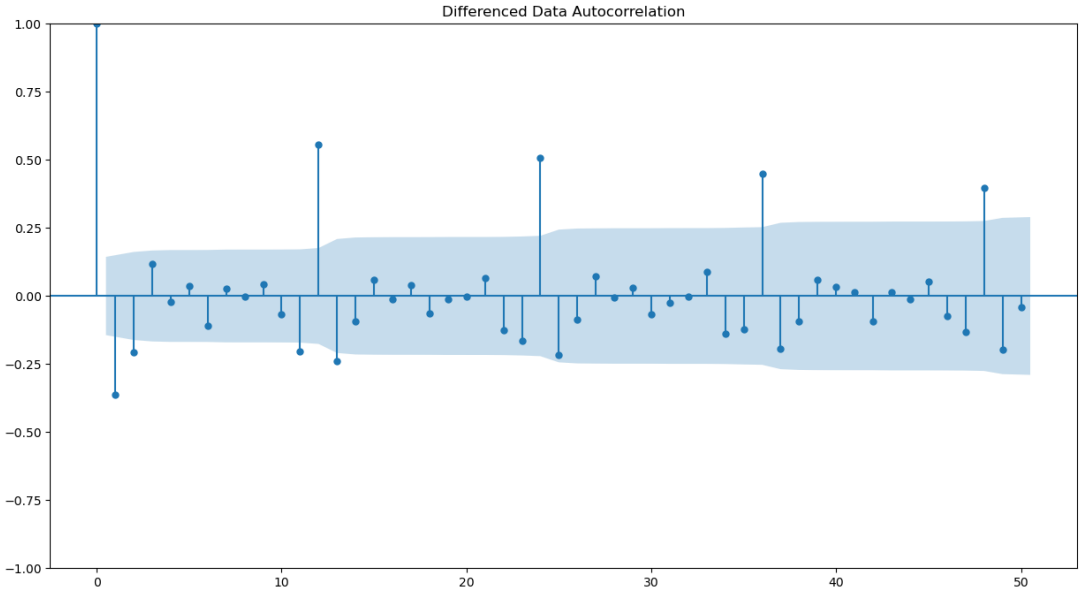
**Autocorrelation function plots on the whole data-**

- PACF (Partial Autocorrelation Function) Plot-
  - To determine value of ‘p’
  - Measures relationship after eliminating effect of lags

- ACF Plot-
  - To determine value of ‘q’
  - Measures how much Time series is correlated with itself at different lags

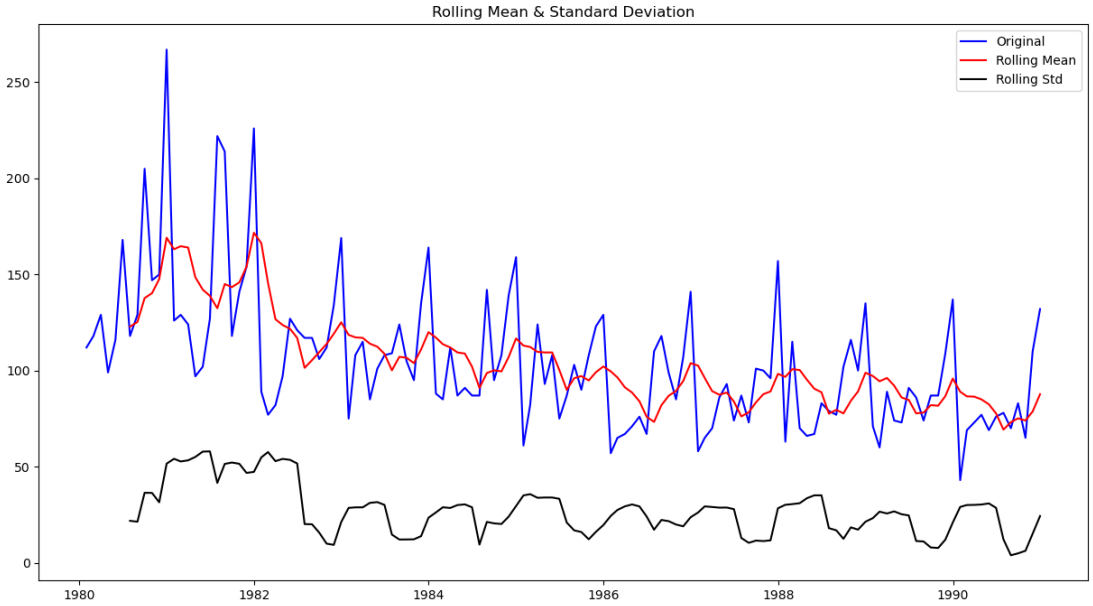
- Autocorrelation decreases as lag increases.
- The terms ‘Lag’ is the number of data points we are looking back.




- From the above plots, we can say that there seems to be a seasonality in the data.
- From ACF Plot, we can see that the seasonality repeats at 12 months and at 6 months, it is repeating in mirror image. Seasonality can be 6 or 12.
- From the PCF plot, the manual analysis of Q value= 2

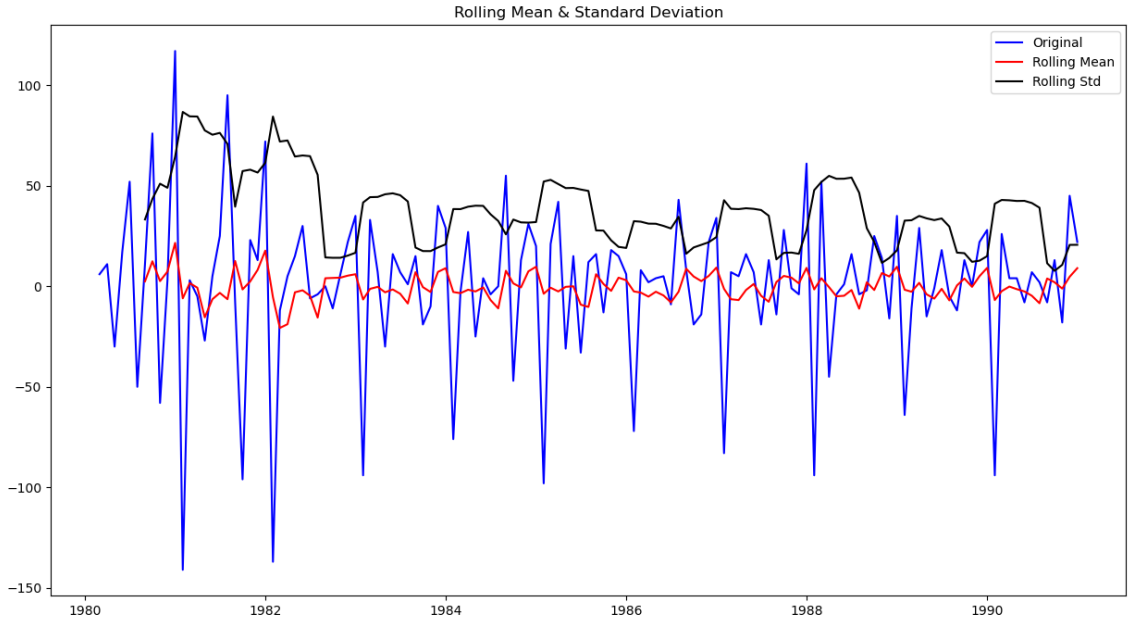
**Check for stationarity of the Training Data Time Series-**




```
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
```

- We see that the series is non-stationary at alpha = 0.05 as p-value (0.219476) > 0.05. Hence, null hypothesis is failed to be rejected.

**Taking a difference of order 1-**




```
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
```

- p-value is < 0.05. Hence TS is stationary.

### 3.6 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.

#### 3.6.1 ARIMA:

- Auto Regressive Integrated Moving Average is a way of modeling time series data for forecasting or predicting future data points
- Improving AR Models by making Time Series stationary through Moving Average Forecasts
- ARIMA models consist of 3 components:-
    - AR model: The data is modeled based on past observations.
    - Integrated component: Whether the data needs to be differenced/transformed.
    - MA model: Previous forecast errors are incorporated into the model.

**ARIMA Model building to estimate best p, d, q parameters using Lowest AIC Approach -**

- AIC- Akaike Information Criteria
- Lowest AIC value compared among different orders of 'p' is considered
- Lower the AIC, better the 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)
```

- d is constant with value of 1. Only 'p' and 'q' varies.

**Calculating AIC value for different parameters-**

```
ARIMA(0, 1, 0) - AIC:1333.1546729124348
ARIMA(0, 1, 1) - AIC:1282.3098319748312
ARIMA(0, 1, 2) - AIC:1279.6715288535818
ARIMA(1, 1, 0) - AIC:1317.3503105381492
ARIMA(1, 1, 1) - AIC:1280.5742295380064
ARIMA(1, 1, 2) - AIC:1279.8707234231902
ARIMA(2, 1, 0) - AIC:1298.611034160493
ARIMA(2, 1, 1) - AIC:1281.507862186858
ARIMA(2, 1, 2) - AIC:1281.8707222264393
```

**Sorting them based on AIC value-**

| | param | AIC |
|---|---|---|
| **2** | (0, 1, 2) | 1279.671529 |
| **5** | (1, 1, 2) | 1279.870723 |
| **4** | (1, 1, 1) | 1280.574230 |
| **7** | (2, 1, 1) | 1281.507862 |
| **8** | (2, 1, 2) | 1281.870722 |
| **1** | (0, 1, 1) | 1282.309832 |
| **6** | (2, 1, 0) | 1298.611034 |
| **3** | (1, 1, 0) | 1317.350311 |
| **0** | (0, 1, 0) | 1333.154673 |

**Building ARIMA model with parameters considered best with lowest AIC value of 1279.672 i.e. (0,1,2) p=0, d=1, q=2**

SARIMAX Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Rose | No. Observations: | 132 |
| Model: | ARIMA(0, 1, 2) | Log Likelihood | -636.836 |
| Date: | Sat, 27 Jan 2024 | AIC | 1279.672 |
| Time: | 18:38:53 | BIC | 1288.297 |
| Sample: | 01-31-1980 | HQIC | 1283.176 |
| | - 12-31-1990 | | |
| Covariance Type: | opg | | |

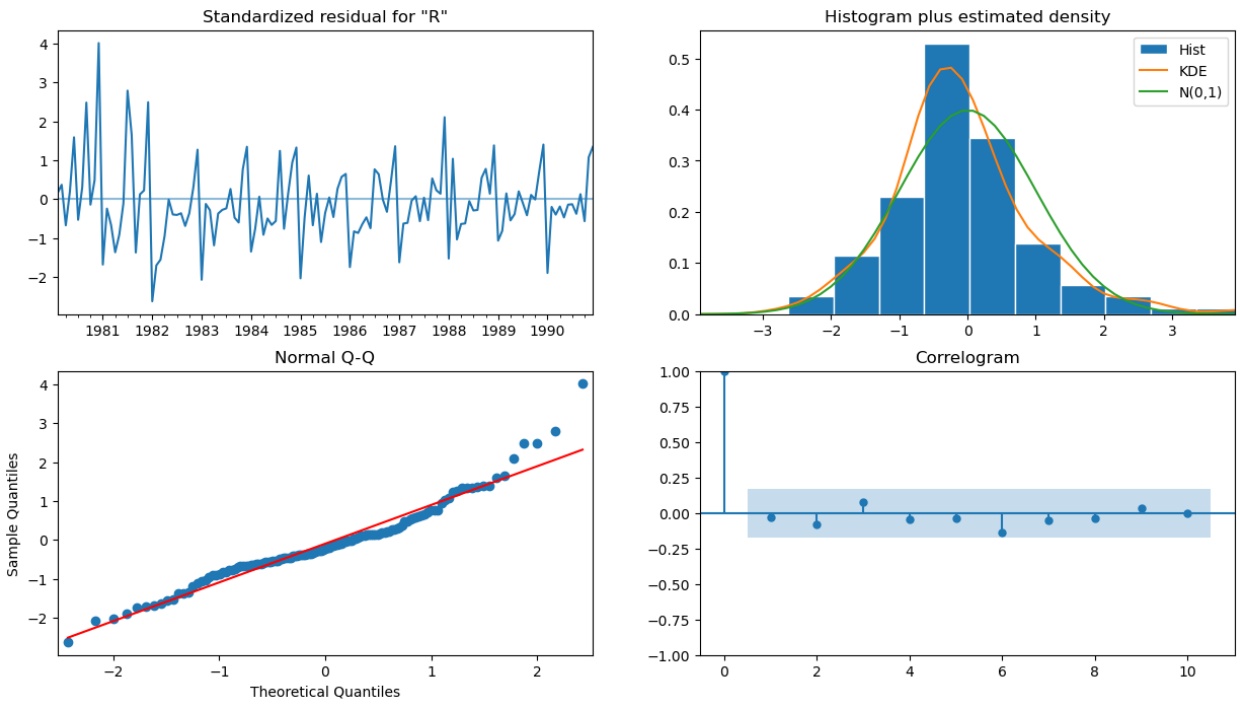
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ma.L1 | -0.6970 | 0.072 | -9.689 | 0.000 | -0.838 | -0.556 |
| ma.L2 | -0.2042 | 0.073 | -2.794 | 0.005 | -0.347 | -0.061 |
| sigma2 | 965.8407 | 88.305 | 10.938 | 0.000 | 792.766 | 1138.915 |

| | | | |
|---|---|---|---|
| Ljung-Box (L1) (Q): | 0.14 | Jarque-Bera (JB): | 39.24 |
| Prob(Q): | 0.71 | Prob(JB): | 0.00 |
| Heteroskedasticity (H): | 0.36 | Skew: | 0.82 |
| Prob(H) (two-sided): | 0.00 | Kurtosis: | 5.13 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

- It includes a first-order differencing (d=1) to achieve stationarity and has two moving average terms MA1 and MA2.
- Prob(Q)= 0.71 > 0.05, hence residuals are independent i.e no significant autocorrelation in the residuals
- Prob(JB)=0 indicates that the residuals do not follow a normal distribution
- Skewness= 0.82 which indicates a positive right skewness
- Kurtosis of 5.13 indicates presence of outliers or extreme values.

**Diagnostics Plot-**




**4 plots in the residuals diagnostic plots tell us** :

- **Standardized residuals plot:** The top left plot shows 1-step-ahead standardized residuals. If model is working correctly, then no pattern should be obvious in the residuals.
- **Histogram plus estimated density plot:** This plot shows the distribution of the residuals. The orange line shows a smoothed version of this histogram, and the green line shows a normal distribution. If the model is good these two lines should be the same. Here there are small differences between them, which indicate that our model is doing just well enough.
- **Normal Q-Q plot:** The Q-Q plot compare the distribution of residuals to normal distribution. If the distribution of the residuals is normal, then all the points should lie along the red line, except for some values at the end, which is exactly happening in this case.
- **Correlogram plot:** The correlogram plot is the ACF plot of the residuals rather than the data. 95% of the correlations for lag >0 should not be significant (within the blue shades). If there is a significant correlation in the residuals, it means that there is information in the data that was not captured by the model, which is clearly not in this case.

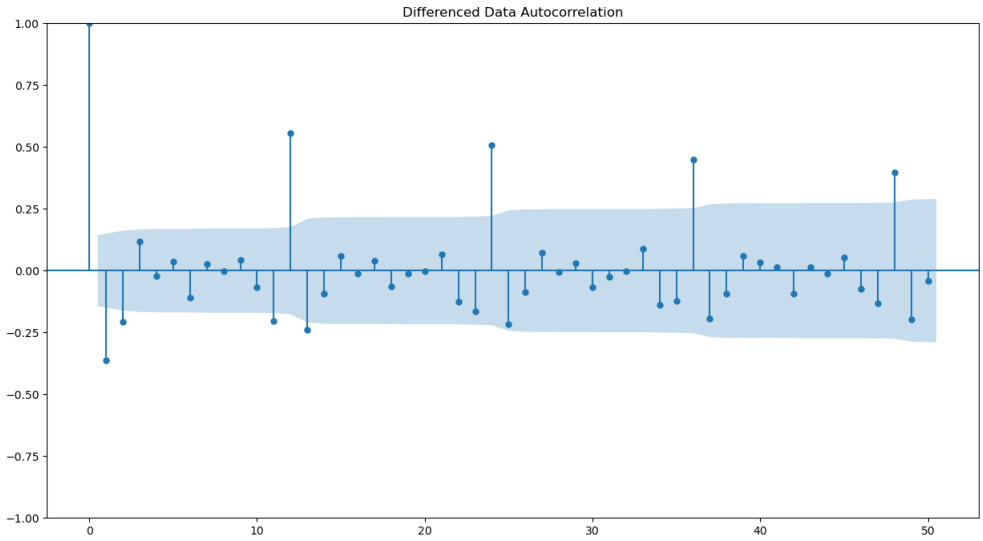
**Predict on the Test Set using this model and evaluate the model-**

RMSE: **37.32712717024891**

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |
| Alpha=0,Beta=0:DES | 15.275675 |
| Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive | 14.267868 |
| Alpha=0.0715,Beta=0.045,Gamma=0:TES Multiplicative | 20.184416 |
| ARIMA(0,1,2) | 37.327127 |

### 3.6.2 SARIMA Model-

- The ARIMA models can be extended/improved to handle seasonal components of a data series.
- The seasonal autoregressive moving average model is given by SARIMA (p, d, q)(P, D, Q)F
- The model consists of:
    - Autoregressive and moving average components (p, q)
    - Seasonal autoregressive and moving average components (P, Q)
    - The ordinary and seasonal difference components of order ‘d’ and ‘D’
    - Seasonal frequency ‘F’
- The value for the parameters (p,d,q) and (P, D, Q) can be decided by comparing different values for each and taking the lowest AIC value for the model build.
- The value for F can be consolidated by ACF plot




- We see that there can be a seasonality of 6 as well as 12.

**Setting the seasonality as 12 to estimate parameters using auto SARIMA model-**

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

**Calculating AIC values for different parameters-**

| | param | seasonal | AIC |
|---|---|---|---|
| 0 | (0, 1, 0) | (0, 0, 0, 12) | 1323.965788 |
| 1 | (0, 1, 0) | (0, 0, 1, 12) | 1145.423083 |
| 2 | (0, 1, 0) | (0, 0, 2, 12) | 976.437530 |
| 3 | (0, 1, 0) | (1, 0, 0, 12) | 1139.921739 |
| 4 | (0, 1, 0) | (1, 0, 1, 12) | 1116.020787 |
| .. | ... | ... | ... |
| 76 | (2, 1, 2) | (1, 0, 1, 12) | 1044.190935 |
| 77 | (2, 1, 2) | (1, 0, 2, 12) | 907.666149 |
| 78 | (2, 1, 2) | (2, 0, 0, 12) | 897.346444 |
| 79 | (2, 1, 2) | (2, 0, 1, 12) | 898.378189 |
| 80 | (2, 1, 2) | (2, 0, 2, 12) | 890.668798 |

[81 rows x 3 columns]

**Sorting based on AIC values-**

| | param | seasonal | AIC |
|---|---|---|---|
| **26** | (0, 1, 2) | (2, 0, 2, 12) | 887.937509 |
| **80** | (2, 1, 2) | (2, 0, 2, 12) | 890.668798 |
| **69** | (2, 1, 1) | (2, 0, 0, 12) | 896.518161 |
| **53** | (1, 1, 2) | (2, 0, 2, 12) | 896.686895 |
| **78** | (2, 1, 2) | (2, 0, 0, 12) | 897.346444 |

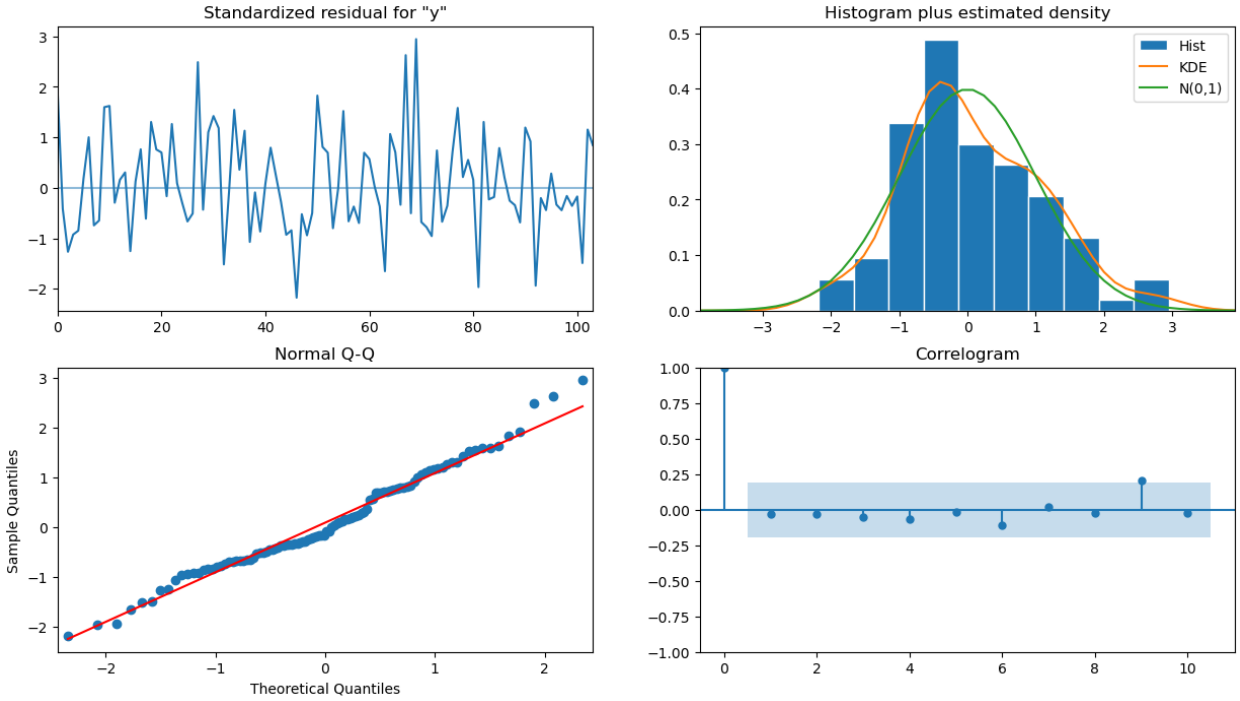
Here,
p = non-seasonal AR order = 0,
d = non-seasonal differencing = 1,
q = non-seasonal MA order = 2,
P = seasonal AR order = 2,
D = seasonal differencing = 0,
Q = seasonal MA order = 2,
S = time span of repeating seasonal pattern = 12

**Building SARIMA model with parameters considered best with lowest AIC value of 887.938 i.e. (p,d,q) (P,D,Q,F): (0,1,2) (2,0,2,12)-**

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:                  132
Model:             SARIMAX(0, 1, 2)x(2, 0, 2, 12)   Log Likelihood                -436.969
Date:                            Sat, 27 Jan 2024   AIC                            887.938
Time:                                    18:41:15   BIC                            906.448
Sample:                                         0   HQIC                           895.437
                                              - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8427    189.817     -0.004      0.996    -372.877     371.192
ma.L2         -0.1573     29.821     -0.005      0.996     -58.606      58.291
ar.S.L12       0.3467      0.079      4.375      0.000       0.191       0.502
ar.S.L24       0.3023      0.076      3.996      0.000       0.154       0.451
ma.S.L12       0.0767      0.133      0.577      0.564      -0.184       0.337
ma.S.L24      -0.0726      0.146     -0.498      0.618      -0.358       0.213
sigma2       251.3137   4.77e+04      0.005      0.996   -9.33e+04    9.38e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.10   Jarque-Bera (JB):                 2.33
Prob(Q):                              0.75   Prob(JB):                         0.31
Heteroskedasticity (H):               0.88   Skew:                             0.37
Prob(H) (two-sided):                  0.70   Kurtosis:                         3.03
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

- We got L24 as frequency of time series is Monthly and seasonality is intra year. And we have given (2,0,2).
- Most contributing to forecast is ma.L1 as coefficient is high. However, higher the p values, lesser the contribution. Hence, coefficients for ma.L1, ma.L2 are not significant as their p-values are of greater value.
- Prob(Q), a high p-value (close to 1) suggests that there is no evidence of significant autocorrelation in the residuals.
- Prob(JB) is 0.31, residuals are not normally distributed.
- Skewness is 0.37 and kurtosis is 3.03. A skewness around 0 and kurtosis around 3 are typical for a normal distribution.
- Heteroskedasticity of 0.88 indicates the variance of the residuals over time.

**Diagnostics Plot:**




- Lag is not beyond Confidence region so correlation is less. We have considered significant zones appropriately i.e. learnt from errors.
- Errors seem normally distributed from the plot.

**Predict on the Test Set using this model and evaluate the model**-

| y | mean | mean_se | mean_ci_lower | mean_ci_upper |
|---|---|---|---|---|
| **0** | 62.867263 | 15.928501 | 31.647976 | 94.086551 |
| **1** | 70.541190 | 16.147659 | 38.892361 | 102.190019 |
| **2** | 77.356411 | 16.147656 | 45.707586 | 109.005235 |
| **3** | 76.208814 | 16.147656 | 44.559989 | 107.857638 |
| **4** | 72.747398 | 16.147656 | 41.098573 | 104.396222 |

**Model evaluation:**

RMSE: **26.94846747578089**

| | Test RMSE |
|---|---|
| **RegressionOnTime** | 15.275687 |
| **NaiveModel** | 79.738587 |
| **SimpleAverageModel** | 53.480911 |
| **2pointTrailingMovingAverage** | 11.529756 |
| **4pointTrailingMovingAverage** | 14.456548 |
| **6pointTrailingMovingAverage** | 14.570933 |
| **9pointTrailingMovingAverage** | 14.731357 |
| **Alpha=0.098,SimpleExponentialSmoothing** | 36.816981 |
| **Alpha=0,Beta=0:DES** | 15.275675 |
| **Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive** | 14.267868 |
| **Alpha=0.0715,Beta=0.045,Gamma=0:TES Multiplicative** | 20.184416 |
| **ARIMA(0,1,2)** | 37.327127 |
| **SARIMA(0,1,2)(2,0,2,12)** | 26.948467 |

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

| | Test RMSE |
|---|---|
| RegressionOnTime | 15.275687 |
| NaiveModel | 79.738587 |
| SimpleAverageModel | 53.480911 |
| 2pointTrailingMovingAverage | 11.529756 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |
| Alpha=0,Beta=0:DES | 15.275675 |
| Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive | 14.267868 |
| Alpha=0.0715,Beta=0.045,Gamma=0:TES Multiplicative | 20.184416 |
| ARIMA(0,1,2) | 37.327127 |
| SARIMA(0,1,2)(2,0,2,12) | 26.948467 |

- 2-point Trailing Moving Average:
  Test RMSE: 11.53
  The 2-point trailing moving average model provides a relatively low RMSE, indicating that it performs well in predicting values. This model considers the average of the last two observations.

- TES Additive (Alpha=0.089, Beta=0.0002, Gamma=0.003):
  Test RMSE: 14.27
  The TES Additive model with specified alpha, beta, and gamma values also has a low RMSE suggesting good predictive performance. This model captures trend, seasonality, and error components.

- 4-point Trailing Moving Average:
  Test RMSE: 14.46
  The 4-point trailing moving average model provides a reasonably low RMSE, indicating good performance. It considers the average of the last four observations.

- 6-point Trailing Moving Average:
  Test RMSE: 14.57

Similar to the 4-point moving average, the 6-point trailing moving average model has a relatively low RMSE, suggesting effective prediction by considering the average of the last six observations.

- 9-point Trailing Moving Average:
  Test RMSE: 14.73
  The 9-point trailing moving average model still performs well with a reasonably low RMSE, considering the average of the last 9 observations.

- DES (Alpha=0, Beta=0):
  Test RMSE: 15.28
  The Double Exponential Smoothing (DES) model with alpha and beta values both set to 0 has a slightly higher RMSE. This indicates that the model's performance may be impacted by not considering trend and seasonality.

- Regression on Time:
  Test RMSE: 15.28
  The regression model on time has a similar RMSE to the DES model suggesting that both models have similar predictive performance.

- TES Multiplicative (Alpha=0.0715, Beta=0.045, Gamma=0):
  Test RMSE: 20.18
  The TES Multiplicative model with specific alpha, beta, and gamma values has a higher RMSE compared to the TES Additive model. It suggests that, the additive approach performs better in this case.

- SARIMA(0,1,2)(2,0,2,12):
  Test RMSE: 26.95
  The Seasonal Autoregressive Integrated Moving Average (SARIMA) model with specified parameters has a higher RMSE, indicating that it may not be as accurate in predicting the values.

- Simple Exponential Smoothing (Alpha=0.098):
  Test RMSE: 36.82
  The simple exponential smoothing model with a specific alpha value has a higher RMSE, suggesting that its predictive accuracy is lower compared to other models.

- ARIMA(0,1,2):
  Test RMSE: 37.33
  The Autoregressive Integrated Moving Average (ARIMA) model with specific orders has a higher RMSE, indicating that its performance may be limited in capturing the underlying patterns in the data.

- Simple Average Model:
  Test RMSE: 53.48
  The simple average model has a higher RMSE, suggesting that it may not be effective in predicting the 'rose' variable.

- Naive Model:
  Test RMSE: 79.74
  The naive model, which predicts future values based on the most recent observation, has the highest RMSE, indicating its limited ability to capture the underlying patterns in the data.

**3.8 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.**

| | Test RMSE |
|---|---|
| 2pointTrailingMovingAverage | 11.529756 |
| Alpha=0.089,Beta=0.0002,Gamma=0.003:TES Additive | 14.267868 |
| 4pointTrailingMovingAverage | 14.456548 |
| 6pointTrailingMovingAverage | 14.570933 |
| 9pointTrailingMovingAverage | 14.731357 |
| Alpha=0,Beta=0:DES | 15.275675 |
| RegressionOnTime | 15.275687 |
| Alpha=0.0715,Beta=0.045,Gamma=0:TES Multiplicative | 20.184416 |
| SARIMA(0,1,2)(2,0,2,12) | 26.948467 |
| Alpha=0.098,SimpleExponentialSmoothing | 36.816981 |
| ARIMA(0,1,2) | 37.327127 |
| SimpleAverageModel | 53.480911 |
| NaiveModel | 79.738587 |

- On comparing the RMSE values, the 2-point Trailing Moving Average model seems the most optimum one.

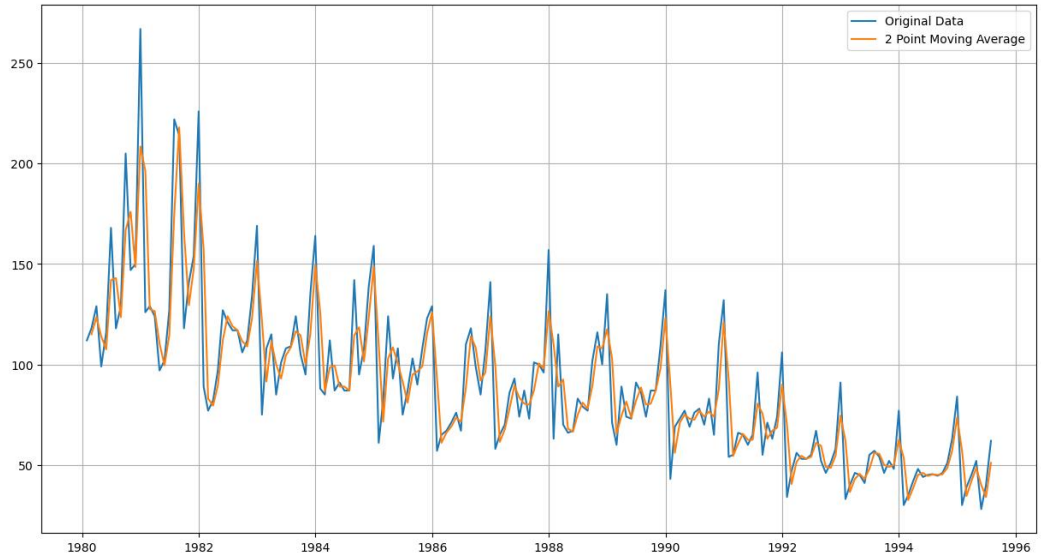
**Displaying head of moving average data frame**

| | 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 |

**Displaying head of moving average data frame along with their trailing values-**

| Time_Stamp | Rose | Trailing_2 |
|---|---|---|
| **1980-01-31** | 112.0 | NaN |
| **1980-02-29** | 118.0 | 115.0 |
| **1980-03-31** | 129.0 | 123.5 |
| **1980-04-30** | 99.0 | 114.0 |
| **1980-05-31** | 116.0 | 107.5 |

**Plotting on the whole data-**

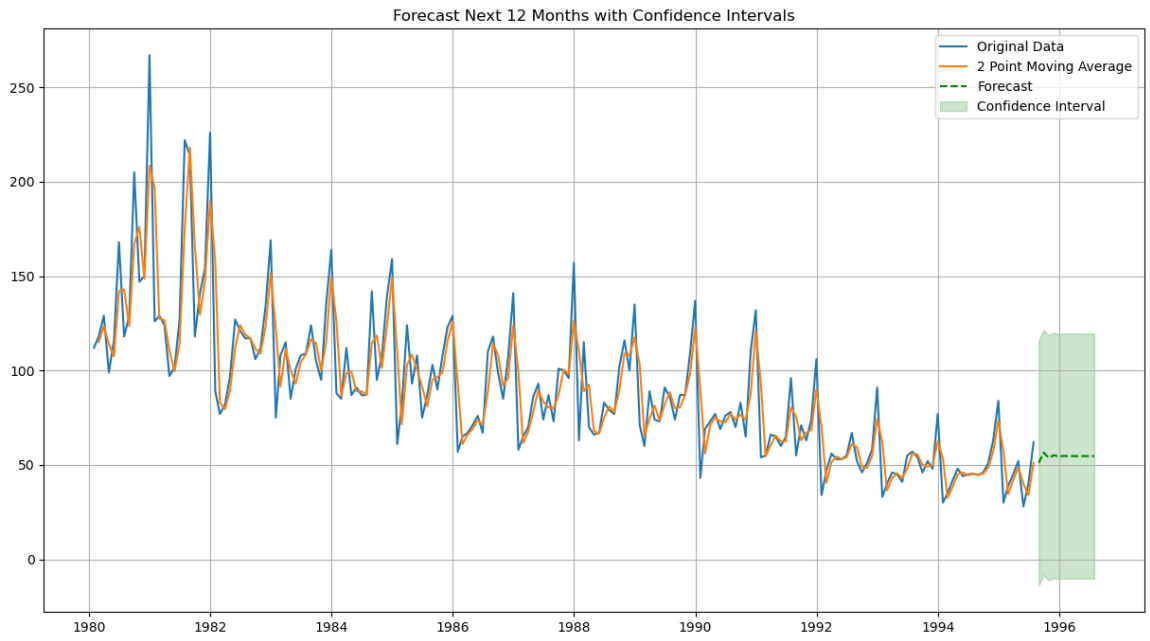



Calculating 2-point average for last 2 data points. Creating a forecast list, iterating the calculation and appending the values for next 12 months-.

**Forecast dataframe along with their confidence intervals-**

| | Forecast | Lower_CI | Upper_CI |
|---|---|---|---|
| **1995-08-31** | 51.000000 | -13.901285 | 115.901285 |
| **1995-09-30** | 56.500000 | -8.401285 | 121.401285 |
| **1995-10-31** | 53.750000 | -11.151285 | 118.651285 |
| **1995-11-30** | 55.125000 | -9.776285 | 120.026285 |
| **1995-12-31** | 54.437500 | -10.463785 | 119.338785 |
| **1996-01-31** | 54.781250 | -10.120035 | 119.682535 |
| **1996-02-29** | 54.609375 | -10.291910 | 119.510660 |
| **1996-03-31** | 54.695312 | -10.205972 | 119.596597 |
| **1996-04-30** | 54.652344 | -10.248941 | 119.553629 |
| **1996-05-31** | 54.673828 | -10.227457 | 119.575113 |
| **1996-06-30** | 54.663086 | -10.238199 | 119.564371 |
| **1996-07-31** | 54.668457 | -10.232828 | 119.569742 |

**Plotting the forecast for next 12 months along with the original data-**




**Forecasted values-**

Forecasted Values_Rose.csv

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

- We have compared various forecasting models. It's essential to choose a forecasting model that best fits the data. In this case, the 2-point Trailing Moving Average model outperforms other models in terms of accuracy.
- The forecasted values along with lower and upper confidence intervals, provided a range of expected outcomes for future months.
- The model forecasts sale of around 653 units of Rose wine sales on an average in next 12 months (Aug 1995- July 1996). And approximately 54 units per month.
- We can see a clear decline in trend since 1981. The business will need to investigate on the low demand in the market.
- However, the sale boosts in the month of November and December every year. So, we need to be prepared for the demand during the festivities.
- The average Rose wine sales value is 89.909, indicating the typical sales level. However, significant fluctuations and outliers are noticeable.
- Recognizing the inconsistent years (e.g., 1980, 1982, 1995) and identifying the most consistent year (1985) is crucial for planning and strategy. They need to be investigated for potential causes.

**Business Strategy:**

- The data shows a decreasing trend with a seasonal pattern. Understanding these patterns can aid in demand forecasting.
- The seasonal trends need to be considered in production and marketing strategies.
- The overall declining trend in Rose sales, especially in the early 1990s, requires a deeper analysis to understand the underlying factors. The company should focus on addressing the concerns and take measures to identify the potential growth opportunities.
- Strategies to manage inventory, production, and marketing efforts need to be aligned with the observed seasonal patterns.
- Continuous monitoring of market trends and conditions, consumer preferences, and external factors impacting sales is essential for decision-making.

**4. Dataset:**

Sparkling Dataset : Sparkling.csv

Rose Dataset : Rose.csv

# THE END.