Time Series Business Report

Sanjana K Venkatesh

13-08-2024

Table of Contents

List of figures

List of Tables

Context

As an analyst at ABC Estate Wines, we are presented with historical data encompassing the sales of different types of wines throughout the 20th century. These datasets originate from the same company but represent sales figures for distinct wine varieties. Our objective is to delve into the data, analyze trends, patterns, and factors influencing wine sales over the course of the century. By leveraging data analytics and forecasting techniques, we aim to gain actionable insights that can inform strategic decision-making and optimize sales strategies for the future.

Objective

The primary objective of this project is to analyze and forecast wine sales trends for the 20th century based on historical data provided by ABC Estate Wines. We aim to equip ABC Estate Wines with the necessary insights and foresight to enhance sales performance, capitalize on emerging market opportunities, and maintain a competitive edge in the wine industry.

Sample of the dataset

Rose.csv

	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

The dataset has 187 records with 2 rows.

Check the summary statistics

	count	mean	std	min	25%	50%	75%	max
Rose	185.0	90.394595	39.175344	28.0	63.0	86.0	112.0	267.0

Observation:

The Rose has a min of 28 and max of 267. With median as 89. With the range it looks like it has extreme datapoints so it will have outliers.

Check for null records

```
[10]:  1  df.isnull().sum()
```

```
Out[10]: Rose    2
         dtype: int64
```

There are only two missing values.

Null value treatment

```
Rose    0
dtype: int64
```

Check the duplicate records.

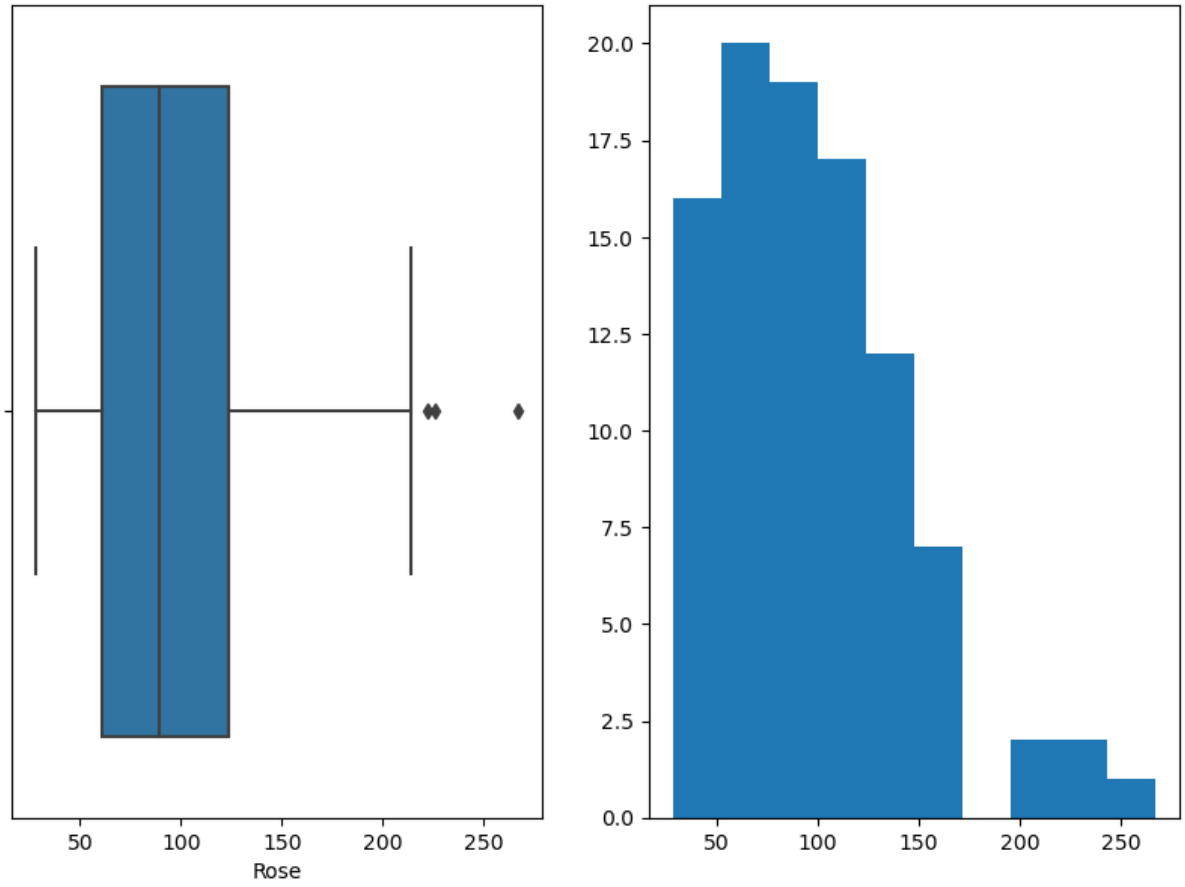
```
Out[14]: 91
```

There are total 91 duplicate records.

Treat the duplicate records by dropping them

Univariant analysis

Rose

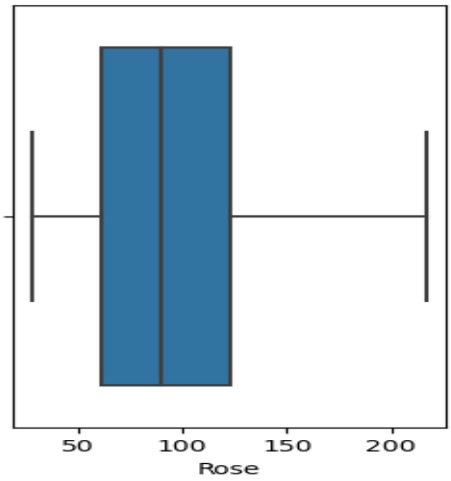


Observation

1. **Rose has extreme values we can see outliers. Its not evenly distributed. Will treat them with box plot technique.**

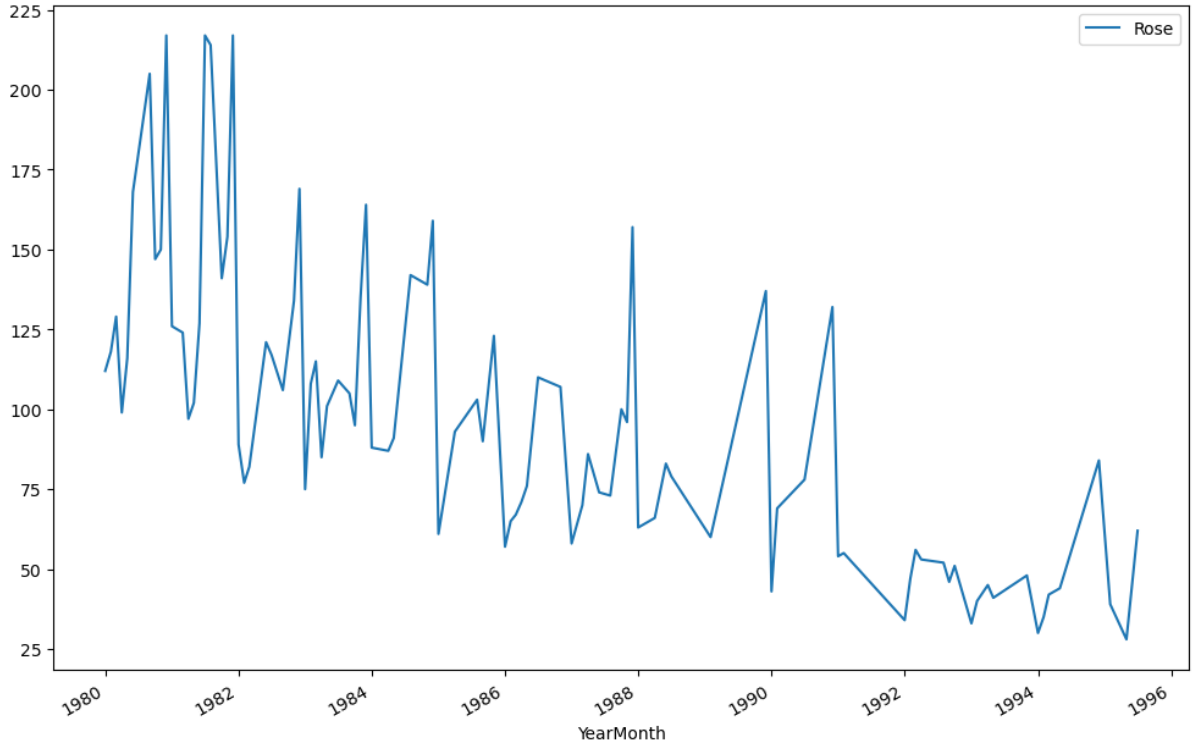
Outlier treatment with box plot technique

lower range -33.0 and upper range 217.0



Outlier datapoints are treated.

Plot the time series



We see an decreasing trend and seasonality which is not constant in nature.

Additive decomposition:

Additive decomposition is a method used in time series analysis to break down a time series into its individual components.

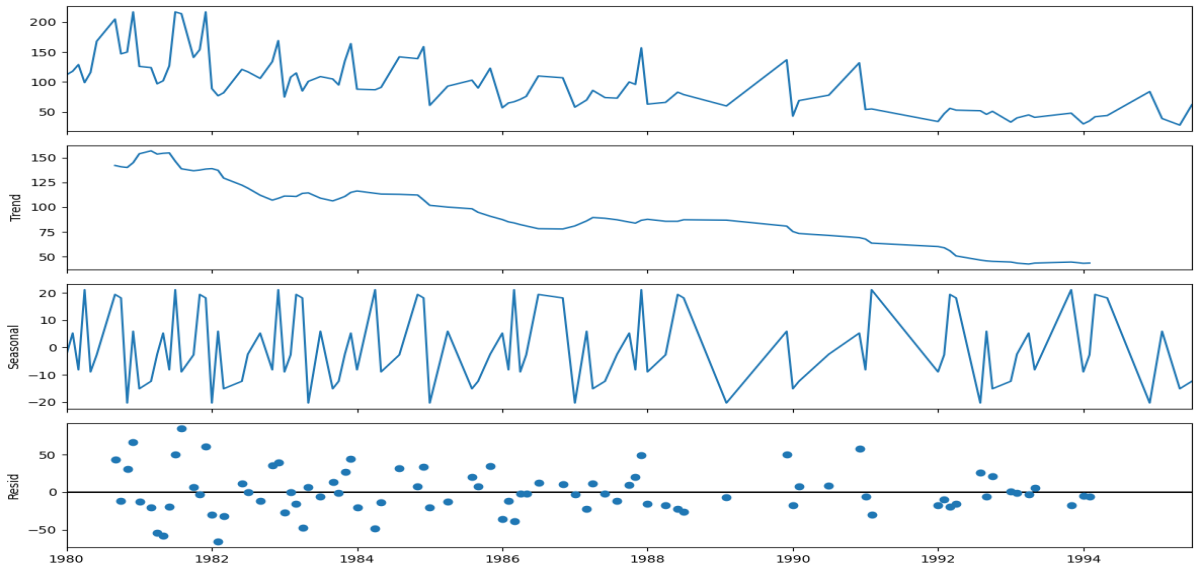
Components:

Trend: The direction of the time series data. It shows the pattern of increase or decrease over time.

Seasonality: The repeating short-term cycle of variations. It occurs due to seasonal flucation.

Cyclic: Fluctuations that are not fixed like seasonality but occur over longer periods.

Residual Component: The irregular or unpredictable variations that are not explained by the other components.



We see that the residuals are located around 0 from the plot of the residuals in the decomposition.

Table on trends and seasonality - Additive

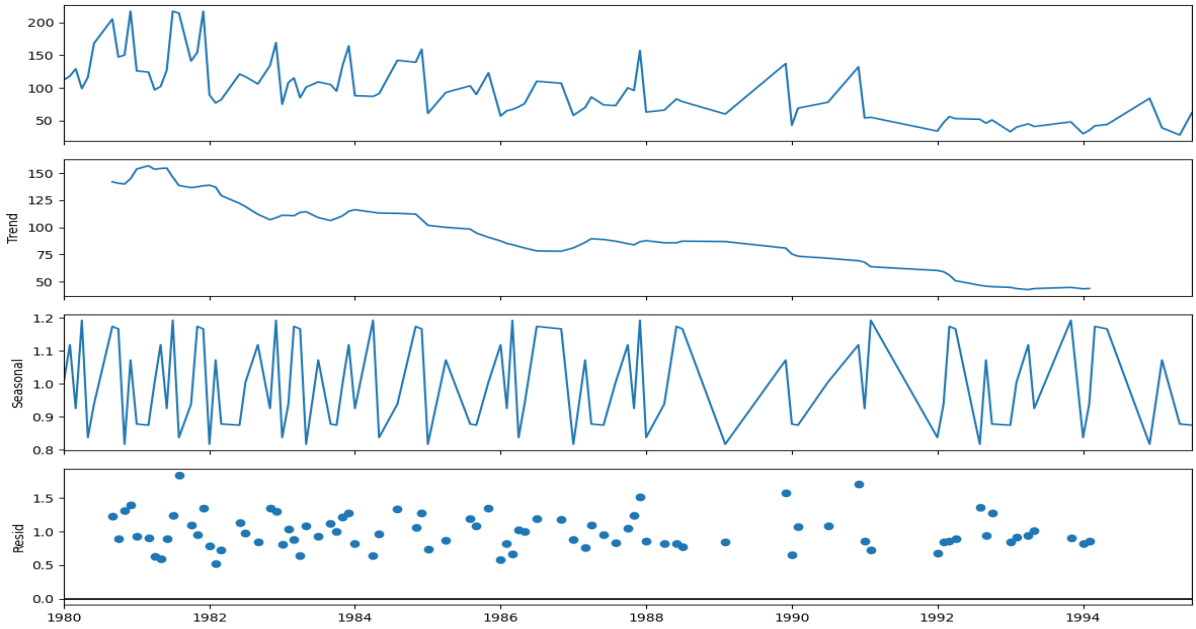
```
Trend
 YearMonth
1980-01-01           NaN
1980-02-01           NaN
1980-03-01           NaN
1980-04-01           NaN
1980-05-01           NaN
1980-06-01           NaN
1980-09-01    141.958333
1980-10-01    140.666667
1980-11-01    139.916667
1980-12-01    144.750000
1981-01-01    153.750000
1981-03-01    156.708333
Name: trend, dtype: float64

Seasonality
 YearMonth
1980-01-01     -2.457837
1980-02-01      5.268353
1980-03-01     -8.154266
1980-04-01     21.155258
1980-05-01     -8.916171
1980-06-01     -2.678075
1980-09-01     19.440972
1980-10-01     18.161210
1980-11-01    -20.309028
1980-12-01      5.946925
1981-01-01    -15.094742
1981-03-01    -12.362599
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01          NaN
1980-02-01          NaN
1980-03-01          NaN
1980-04-01          NaN
1980-05-01          NaN
1980-06-01          NaN
1980-09-01    43.600694
1980-10-01   -11.827877
1980-11-01    30.392361
1980-12-01    66.303075
1981-01-01   -12.655258
```

Multiplicative decomposition:

Multiplicative decomposition is a method used in time series analysis to break down a time series as the product of its components. It has the same component as additive ones.



For the multiplicative series, we see that a lot of residuals are located around 1. Thus Multiplicative Decomposition is the right way to decompose the time series.

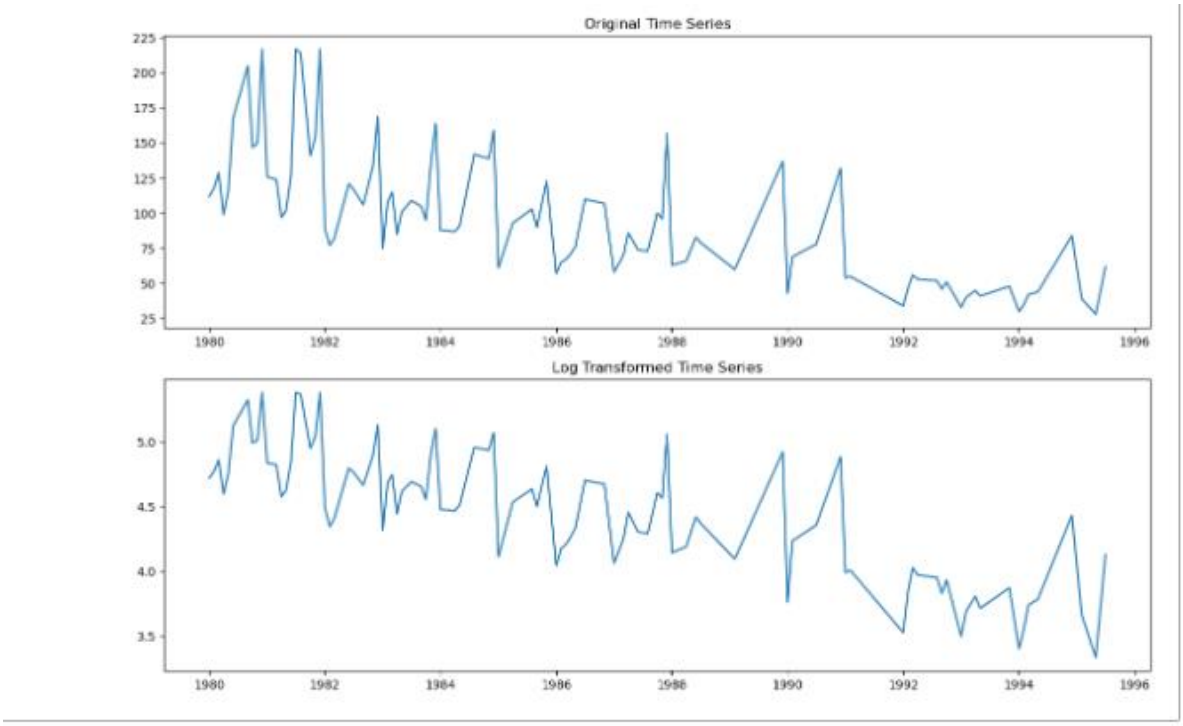
Table on trends and seasonality -Multiplicative

```
Trend
 YearMonth
1980-01-01           NaN
1980-02-01           NaN
1980-03-01           NaN
1980-04-01           NaN
1980-05-01           NaN
1980-06-01           NaN
1980-09-01    141.958333
1980-10-01    140.666667
1980-11-01    139.916667
1980-12-01    144.750000
1981-01-01    153.750000
1981-03-01    156.708333
Name: trend, dtype: float64

Seasonality
 YearMonth
1980-01-01    1.005084
1980-02-01    1.118391
1980-03-01    0.925328
1980-04-01    1.192655
1980-05-01    0.837201
1980-06-01    0.938609
1980-09-01    1.174200
1980-10-01    1.166623
1980-11-01    0.816985
1980-12-01    1.072100
1981-01-01    0.878124
1981-03-01    0.874700
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01         NaN
1980-02-01         NaN
1980-03-01         NaN
1980-04-01         NaN
1980-05-01         NaN
1980-06-01         NaN
1980-09-01    1.229846
1980-10-01    0.895768
1980-11-01    1.312224
1980-12-01    1.398318
1981-01-01    0.933253
1981-03-01    0.904629
Name: resid, dtype: float64
```

Log Transformation: comparison of original and transformed



Model building:

Steps:

1) Maintaining the order or sequencing is important in TSF. While choosing data for train or test, we need to ensure that sequencing is maintained.

2) We choose initial 80% of the data points as 'train' and rest as 'test'. We can choose major part of the initial period as 'train' and rest as 'test'.

Model 1: Linear Regression

Steps:

1) First create a predictor (x) for the time series data which can be just incremental numbers for entire time period i.e. for both 'train' and 'test'

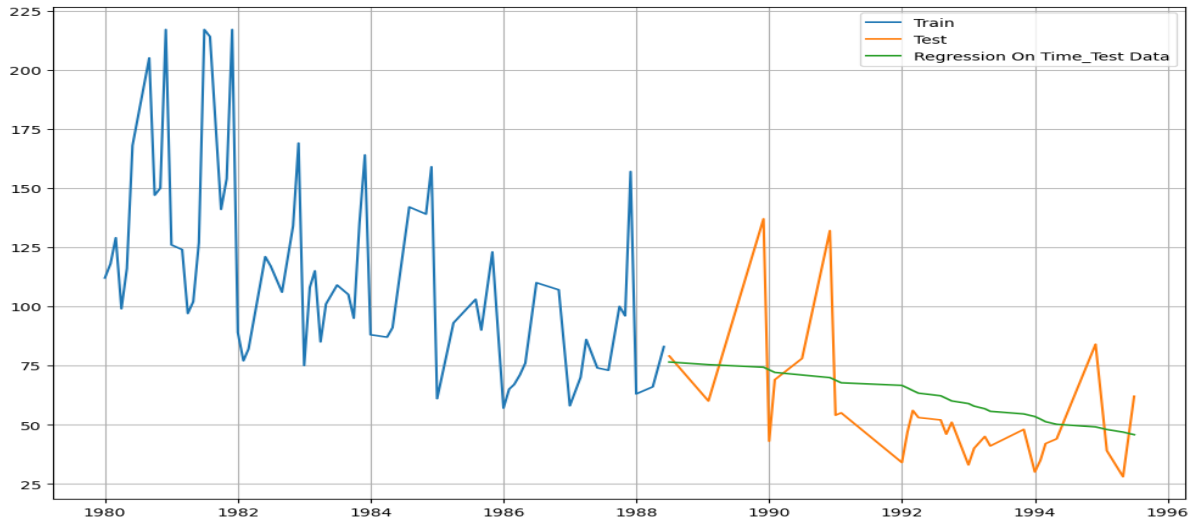
Note: If 'train' has 80 data points and test has 20 data points, x for train will range from 1 to 80 and x for 'test' will range from 81 to 100

2) Apply LinearRegression (y=f(x)) on the 'train' data and build a model and fit the model.

```
▾ LinearRegression
LinearRegression()
```

3) Use the model to predict 'y' for the test period and get the required forecast.

Plot for Regression On Time_Test Data



Model Evaluation:

Use the Root mean square error method to calculate the accuracy.

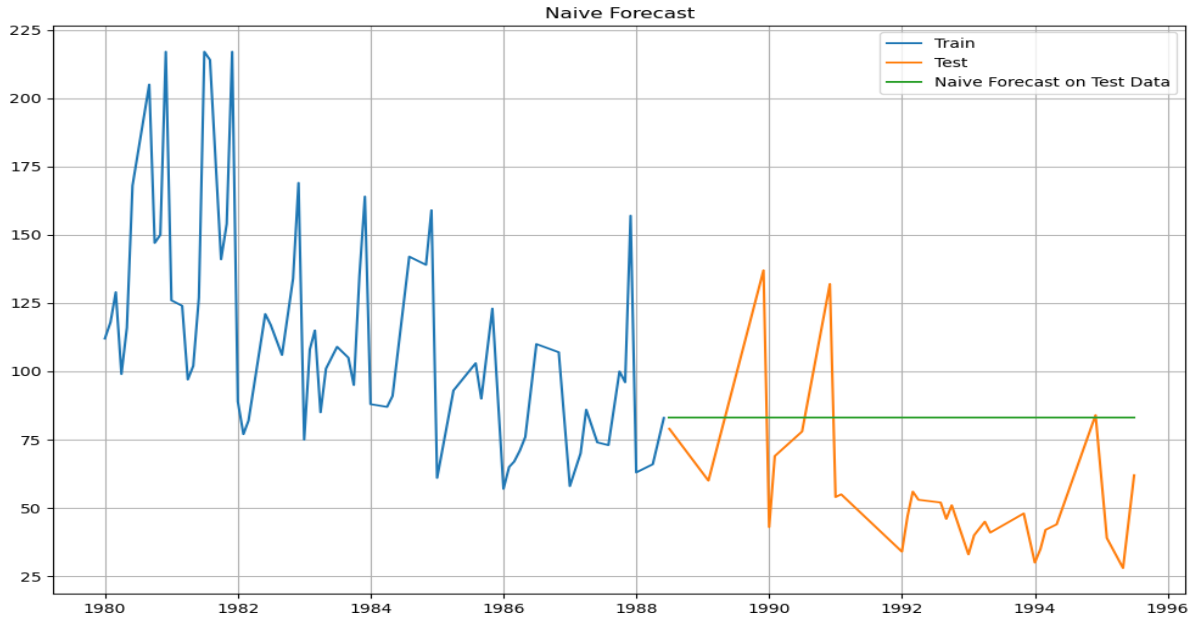
	Test RMSE
RegressionOnTime	23.237925

Model 2: Naive Approach

The prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.¶

Pick up the last value from the 'train' and use it as a forecast for future i.e. entire 'test' period.

Plot the graph for train and test data.



Model Evaluation:

Using RMSE Naïve approach gave 37.

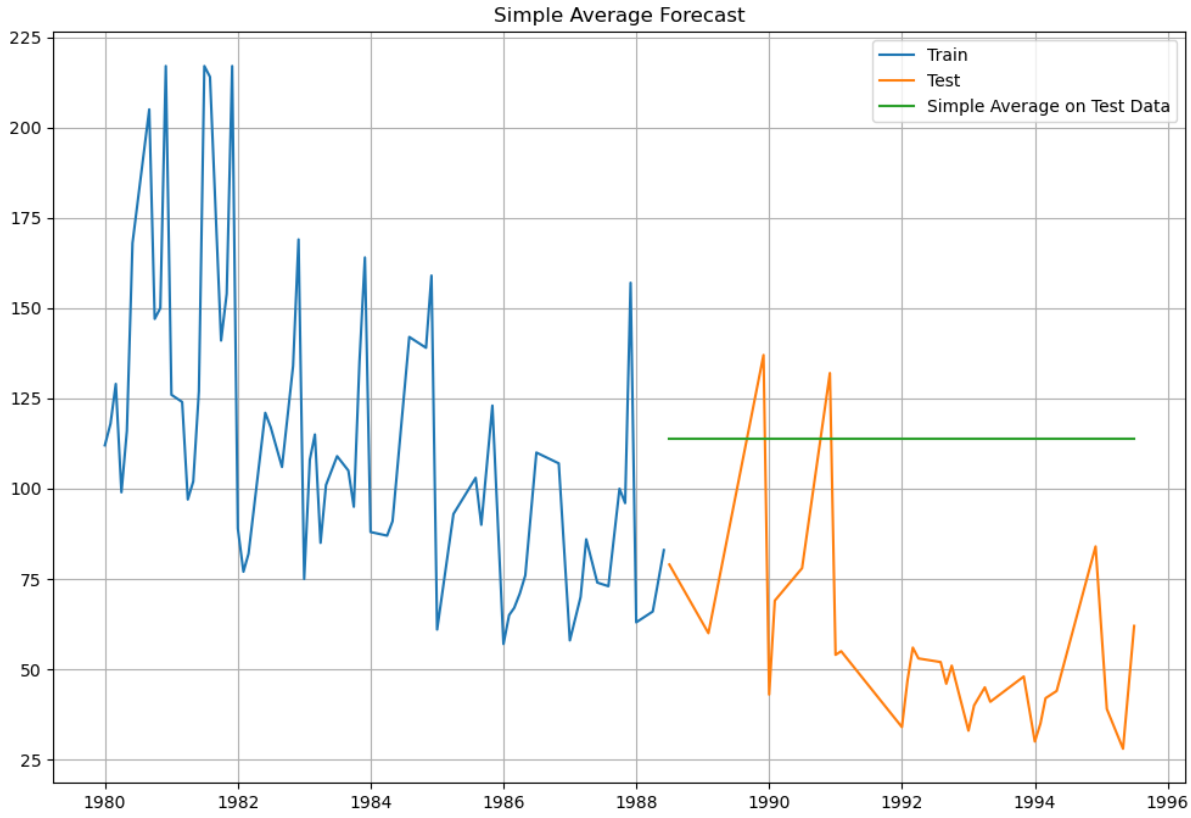
	Test RMSE
RegressionOnTime	23.237925
NaiveModel	37.364929

Method 3: Simple Average

We will forecast by using the average of the training values.¶

We are finding average of the value for entire 'train' period and use it as a forecast for future i.e. entire 'test' period.

Plot on Simple average



Model Evaluation:

The RMSE gave 63.

	Test RMSE
RegressionOnTime	23.237925
NaiveModel	37.364929
SimpleAverageModel	63.390407

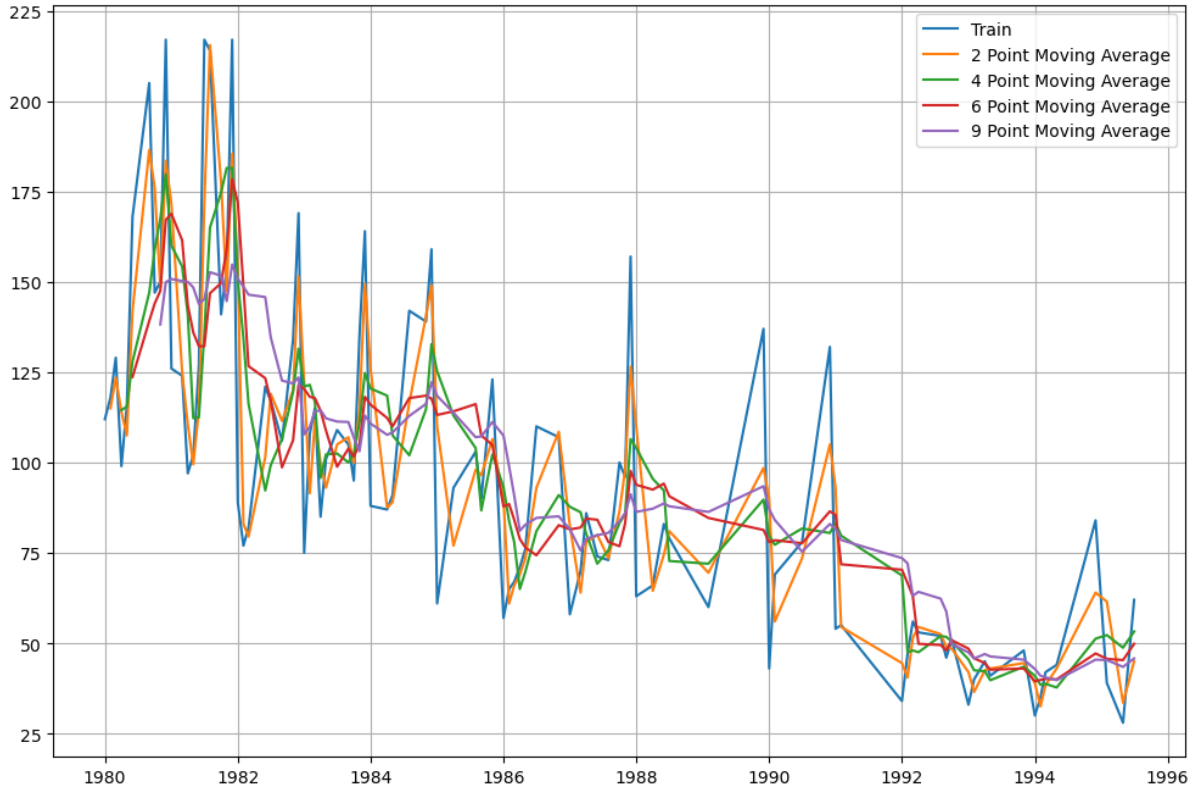
Method 4: Moving Average(MA)

Steps:

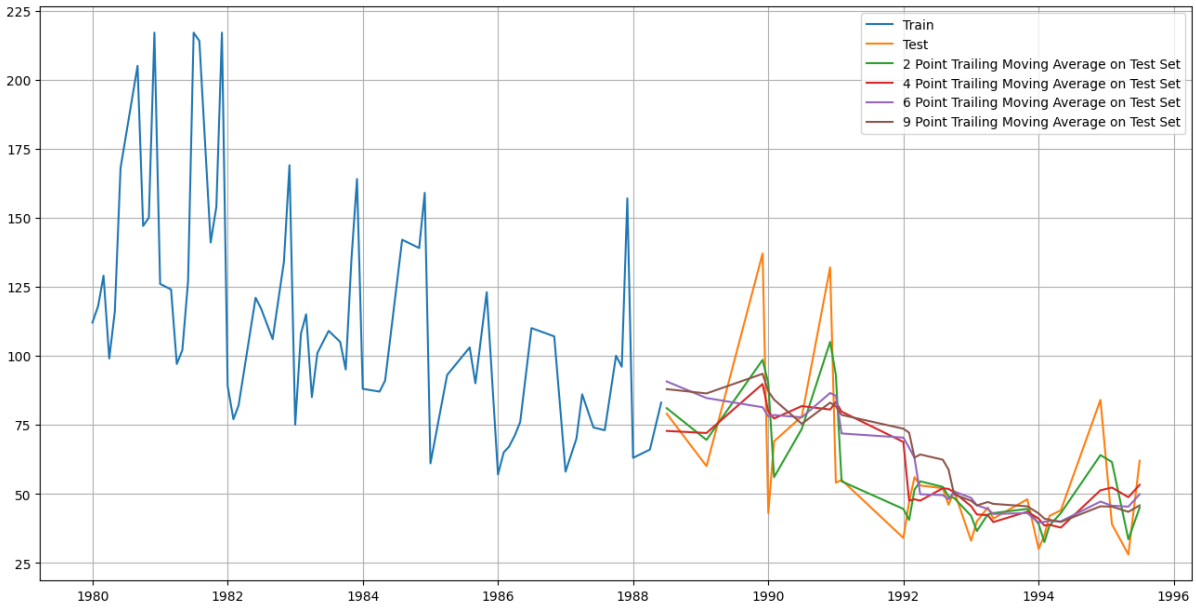
1) Based on the time series pattern and business understanding we choose a window to apply moving average.

2) For the chosen window, we apply moving average from the start of train period and get forecast for 1 additional time period beyond train.

3) We use rolling function in Python to implement Moving Average.



Plotting on both the Training and Test data



Model Evaluation

The RMSE score for MA is as follows. We could see for 2point trailing MA we have best score.

	Test RMSE
RegressionOnTime	23.237925
NaiveModel	37.364929
SimpleAverageModel	63.390407
2pointTrailingMovingAverage	16.460245
4pointTrailingMovingAverage	19.856435
6pointTrailingMovingAverage	20.815809
9pointTrailingMovingAverage	21.772118

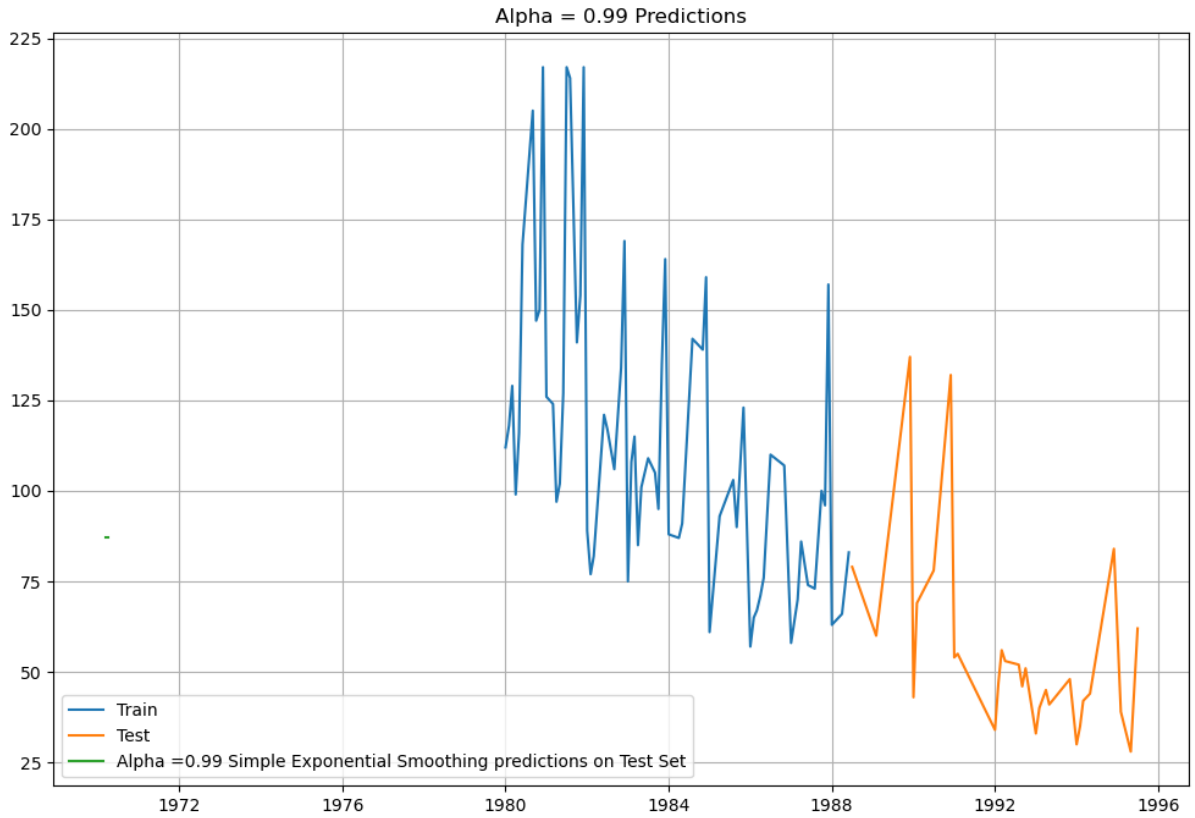
Exponential model

Simple exponential model building:

SES or one-parameter exponential smoothing is applicable to time series which do not contain either of trend or seasonality.

```
{'smoothing_level': 0.15045746473027144,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 133.50451788578772,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Plot the train and test and forecast values

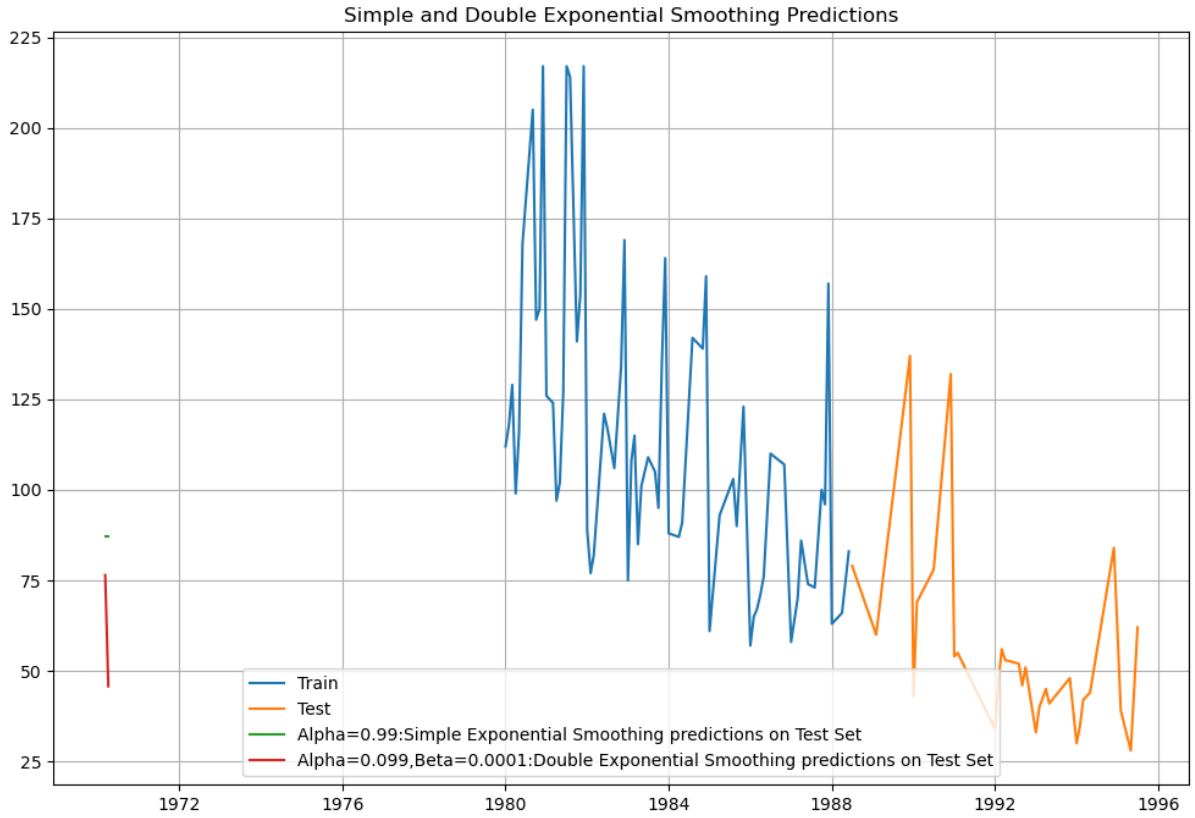


Model evaluation:

	Test RMSE
Alpha=0.99,SES	40.508153

Double exponential smoothing:

This method is applicable where trend is present in the data but no seasonality.



Model Evaluation:

	Test RMSE
Alpha=0.99,SES	40.508153
Alpha=1,Beta=0.0189:DES	23.237927

Inference

Here, we see that the Double Exponential Smoothing has actually done well when compared to the Simple Exponential Smoothing. This is because of the fact that the Double Exponential Smoothing model has picked up the trend component as well.

Triple Exponential smoothing:

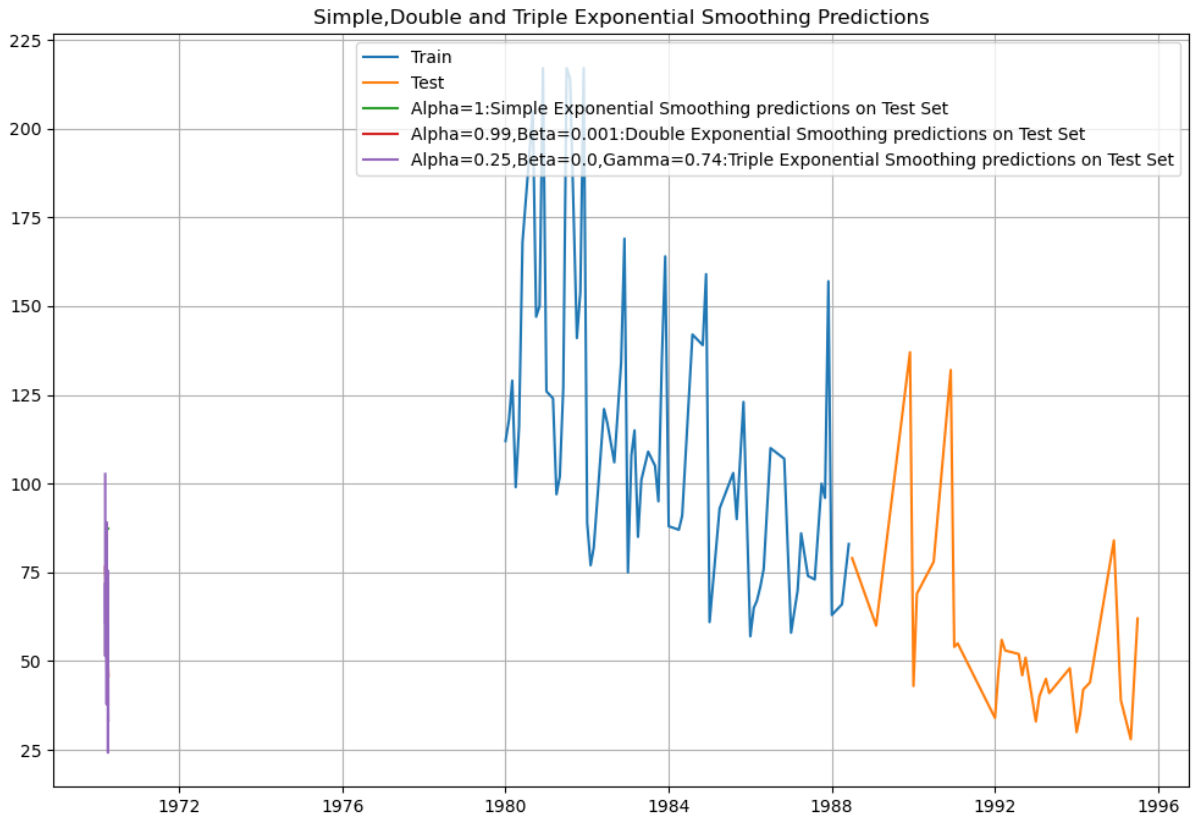
This is an extension of Holt's method when seasonality is found in the data.
This is also known as three parameters exponential or triple exponential because of the three smoothing parameters α, β and γ. This is a general method and a true multi-step ahead forecast.

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

The Holt-Winters ETS (A, A, A) method is a version of the Holt-Winters model that uses additive error, additive trend, and additive seasonality. It is used for time series forecasting when the data exhibits both trend and seasonality, and the seasonal variations are constant over time.

Components of ETS (A, A, A):

- Error (E): Additive error (A)
- Trend (T): Additive trend (A)
- Seasonality (S): Additive seasonality (A)



Model evaluation:

	Test RMSE
Alpha=0.99,SES	40.508153
Alpha=1,Beta=0.0189:DES	23.237927
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	28.165338

Holt-Winters - ETS(A, A, M) - Holt Winter's linear method

The **Holt-Winters ETS (A, A, M)** method is another variant of the Exponential Smoothing (ETS) model, where the error and trend are modelled additively, but the seasonality is modelled multiplicatively.

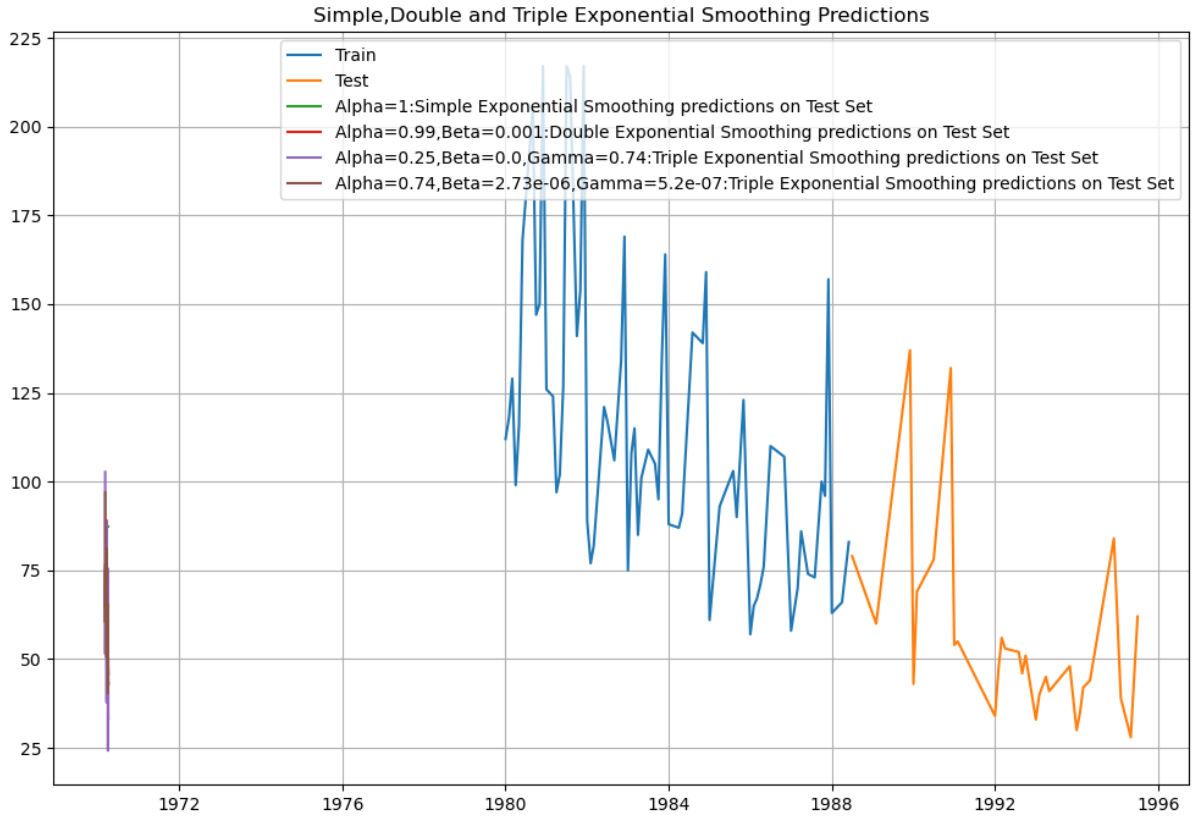
Components of ETS (A, A, M):

- **Error (E)**: Additive error (A)
- **Trend (T)**: Additive trend (A)
- **Seasonality (S)**: Multiplicative seasonality (M)

	Test RMSE
Alpha=0.99,SES	40.508153
Alpha=1,Beta=0.0189:DES	23.237927
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	28.165338
Alpha=0.74,Beta=2.73e-06,Gamma=5.2e-07,Gamma=0:TES	25.569568

Inference:

We see that the multiplicative seasonality model has done well when compared to the additive seasonality Triple Exponential Smoothing model. But still double soothing model is better.



Check for Stationarity

A stationary time series has statistical properties (mean, variance, autocovariance) that do not change over time.

Augmented Dickey-Fuller (ADF) Test:

The ADF test is one of the most widely used statistical tests for stationarity.

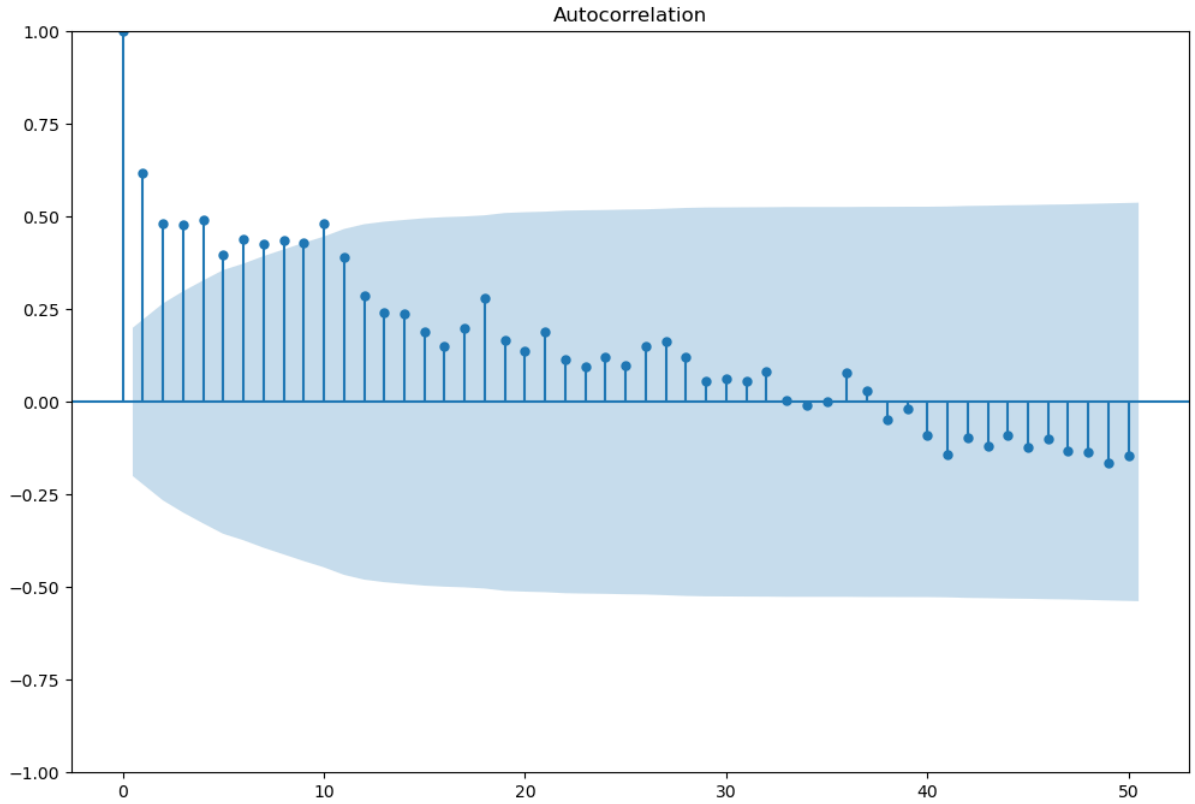
Steps to interpret ADF test:

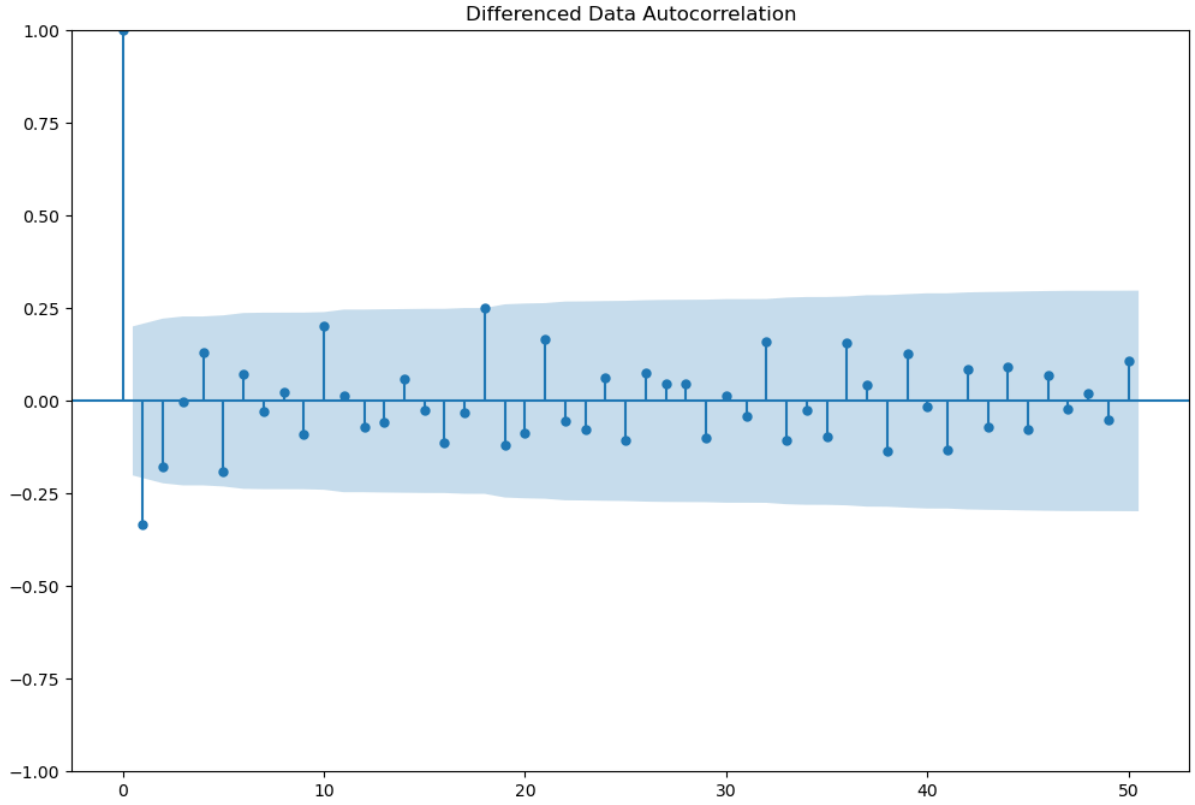
- Null hypothesis: The series has a unit root (non-stationary).
- Alternative hypothesis: The series is stationary.
- If the p-value is less than a chosen significance level (e.g., 0.05), the null hypothesis is rejected, and the series is considered stationary.

How to Make a Time Series Stationary (If It's Not)

- Differencing
- Log Transformation
- De-trending
- De-seasonalization

Plot the Autocorrelation function plots on the whole data.





ARIMA

ARIMA is suitable for non-seasonal data and focuses on trends and cycles.

Build an Automated version of an ARMA model for which the best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

Based on values of p,q,r the model is formed.

```
Some parameter combinations for the Model...
Model: (0, 0, 1)
Model: (0, 0, 2)
Model: (1, 0, 0)
Model: (1, 0, 1)
Model: (1, 0, 2)
Model: (2, 0, 0)
Model: (2, 0, 1)
Model: (2, 0, 2)
```

Fit the ARIMA model.

The Param and AIC in ascending order after fit.

	param	AIC
3	(1, 0, 0)	676.089535
5	(1, 0, 2)	676.358913
1	(0, 0, 1)	677.817596
4	(1, 0, 1)	678.079842
6	(2, 0, 0)	678.087268
8	(2, 0, 2)	678.284400
2	(0, 0, 2)	678.852847
7	(2, 0, 1)	679.932599
0	(0, 0, 0)	689.918948

The auto ARIMA summary table

```
                                SARIMAX Results
==========================================================================
===
Dep. Variable:                       Rose   No. Observations:
67
Model:                  ARIMA(1, 0, 0)   Log Likelihood                -335.
045
Date:                 Tue, 08 Oct 2024   AIC                            676.
090
Time:                         12:26:36   BIC                            682.
704
Sample:                             0   HQIC                           678.
707
                                  - 67
Covariance Type:                  opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025       0.9
75]
--------------------------------------------------------------------------
---
const        113.3622     10.888     10.412      0.000      92.022      134.
703
ar.L1          0.4570      0.120      3.816      0.000       0.222        0.
692
sigma2      1286.8631    211.429      6.087      0.000     872.470     1701.
256
==========================================================================
========
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):
2.37
Prob(Q):                              0.97   Prob(JB):
0.31
Heteroskedasticity (H):               0.43   Skew:
0.40
Prob(H) (two-sided):                  0.06   Kurtosis:
3.46
==========================================================================
========

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

Evaluation:

Based on mean squared error its 62.60

	RMSE
ARIMA(1,0,0)	62.608946

After changing the values of p,q,r the values looks different.

:

	param	AIC
2	(0, 1, 2)	664.199131
4	(1, 1, 1)	665.306982
5	(1, 1, 2)	666.128886
7	(2, 1, 1)	666.194996
1	(0, 1, 1)	667.754368
8	(2, 1, 2)	668.054341
6	(2, 1, 0)	673.469588
3	(1, 1, 0)	680.056272
0	(0, 1, 0)	683.255294

```
                               SARIMAX Results
==========================================================================
===
Dep. Variable:                    Rose   No. Observations:
67
Model:                 ARIMA(0, 1, 2)   Log Likelihood                -329.
100
Date:                Tue, 08 Oct 2024   AIC                            664.
199
Time:                        12:26:37   BIC                            670.
768
Sample:                             0   HQIC                           666.
795
                                  - 67
Covariance Type:                  opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ma.L1         -0.5618      0.123     -4.566      0.000      -0.803      -0.
321
ma.L2         -0.2932      0.129     -2.278      0.023      -0.545      -0.
041
sigma2      1232.6594    228.523      5.394      0.000     784.764    1680.
555
==========================================================================
========
Ljung-Box (L1) (Q):                   0.06   Jarque-Bera (JB):
0.53
Prob(Q):                              0.81   Prob(JB):
0.77
Heteroskedasticity (H):               0.39   Skew:
0.20
Prob(H) (two-sided):                  0.03   Kurtosis:
3.19
==========================================================================
```

Evaluation:

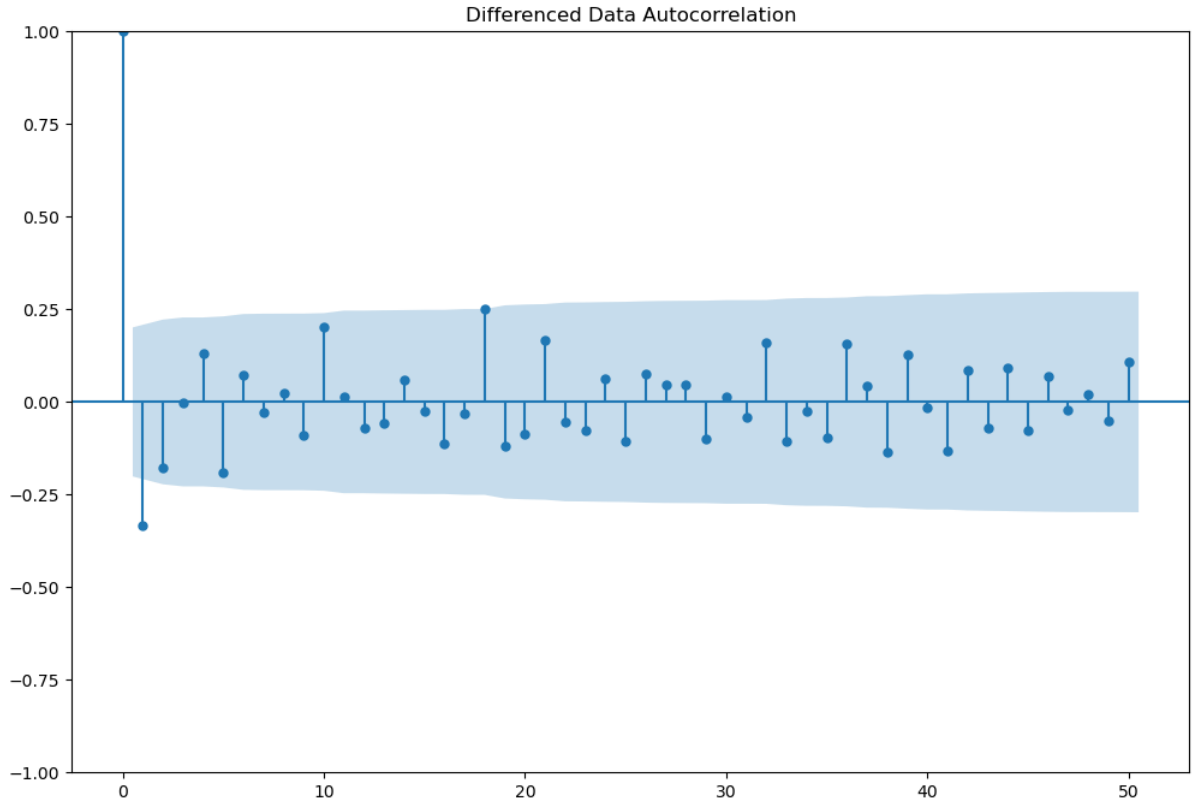
	RMSE
ARIMA(1,0,0)	62.608946
ARIMA(0,1,2)	41.484815

SARIMA:

SARIMA is an extension of ARIMA that incorporates seasonality, making it better for time series data with seasonal fluctuations.

Build an Automated version of a SARIMA model for which the best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC).

ACF plot for Differenced Data Autocorrelation



Based on p,q, d, D values the model.

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 6)
Model: (0, 1, 2)(0, 0, 2, 6)
Model: (1, 1, 0)(1, 0, 0, 6)
Model: (1, 1, 1)(1, 0, 1, 6)
Model: (1, 1, 2)(1, 0, 2, 6)
Model: (2, 1, 0)(2, 0, 0, 6)
Model: (2, 1, 1)(2, 0, 1, 6)
Model: (2, 1, 2)(2, 0, 2, 6)
```

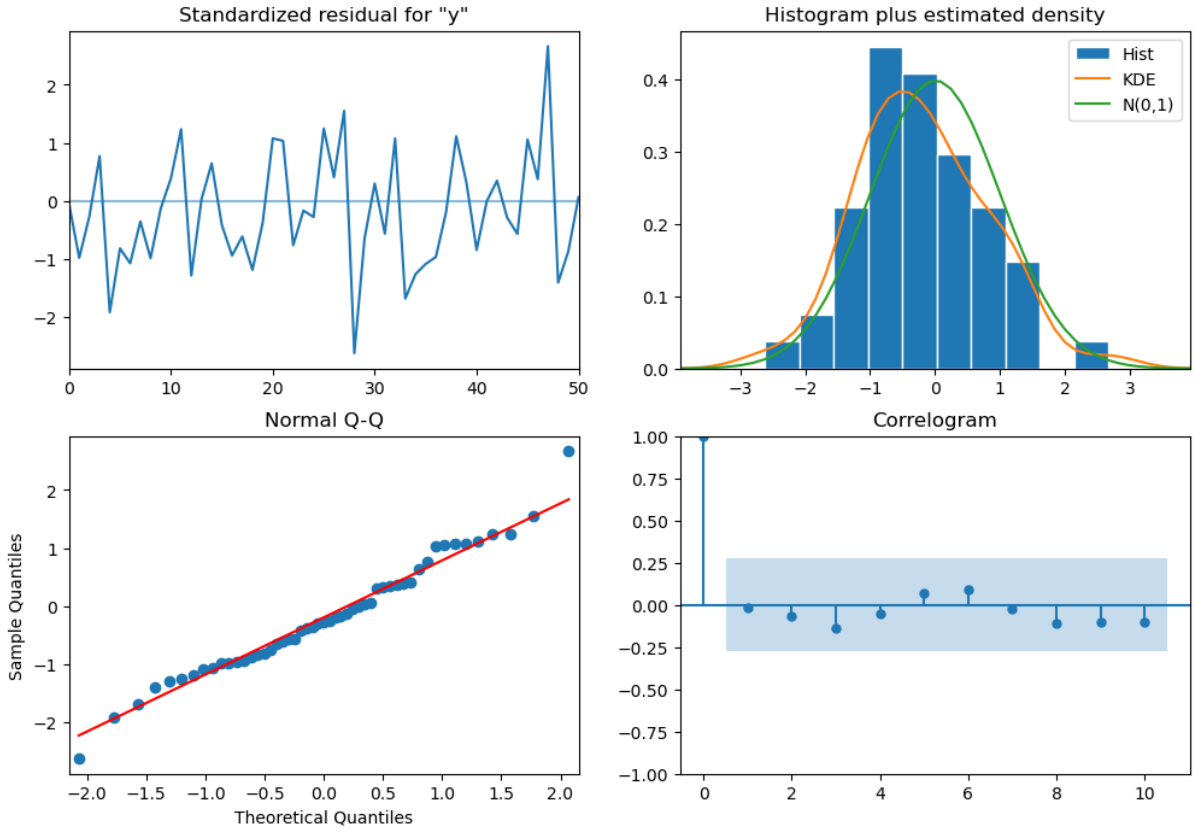
After the best parameter estimate:

	param	seasonal	AIC
53	(1, 1, 2)	(2, 0, 2, 6)	501.713832
80	(2, 1, 2)	(2, 0, 2, 6)	502.408993
77	(2, 1, 2)	(1, 0, 2, 6)	513.487865
20	(0, 1, 2)	(0, 0, 2, 6)	514.649980
50	(1, 1, 2)	(1, 0, 2, 6)	515.230421

The summary result

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:
67
Model:             SARIMAX(1, 1, 2)x(2, 0, 2, 6)   Log Likelihood
-242.857
Date:                           Tue, 08 Oct 2024   AIC
501.714
Time:                                   12:26:49   BIC
517.168
Sample:                                        0   HQIC
507.619
                                            - 67
Covariance Type:                              opg
==============================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
------------------------------------------------------------------------------
---
ar.L1         -0.0736      0.304     -0.242      0.809      -0.670       0.
523
ma.L1         -0.6227      0.352     -1.769      0.077      -1.313       0.
067
ma.L2         -0.1191      0.290     -0.410      0.682      -0.688       0.
450
ar.S.L6       -0.7321      0.103     -7.132      0.000      -0.933      -0.
531
ar.S.L12      -0.5879      0.107     -5.482      0.000      -0.798      -0.
378
ma.S.L6        0.8028      0.283      2.835      0.005       0.248       1.
358
ma.S.L12       0.7280      0.394      1.846      0.065      -0.045       1.
501
sigma2       667.3440    188.726      3.536      0.000     297.447    1037.
241
==============================================================================
========
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):
1.00
Prob(Q):                              0.91   Prob(JB):
0.61
Heteroskedasticity (H):               1.34   Skew:
0.31
Prob(H) (two-sided):                  0.56   Kurtosis:
3.31
```

The plot diagnostics



Evaluation:

The RMSE gives 42.30

	RMSE
ARIMA(1,0,0)	62.608946
ARIMA(0,1,2)	41.484815
SARIMA(1,1,2)(2, 0, 2, 6)	42.300763

Inference:

The Sarima model is giving optimum of RMSE 42. But the ARIMA with optimised p,q,d (0,1,2) its giving 41.

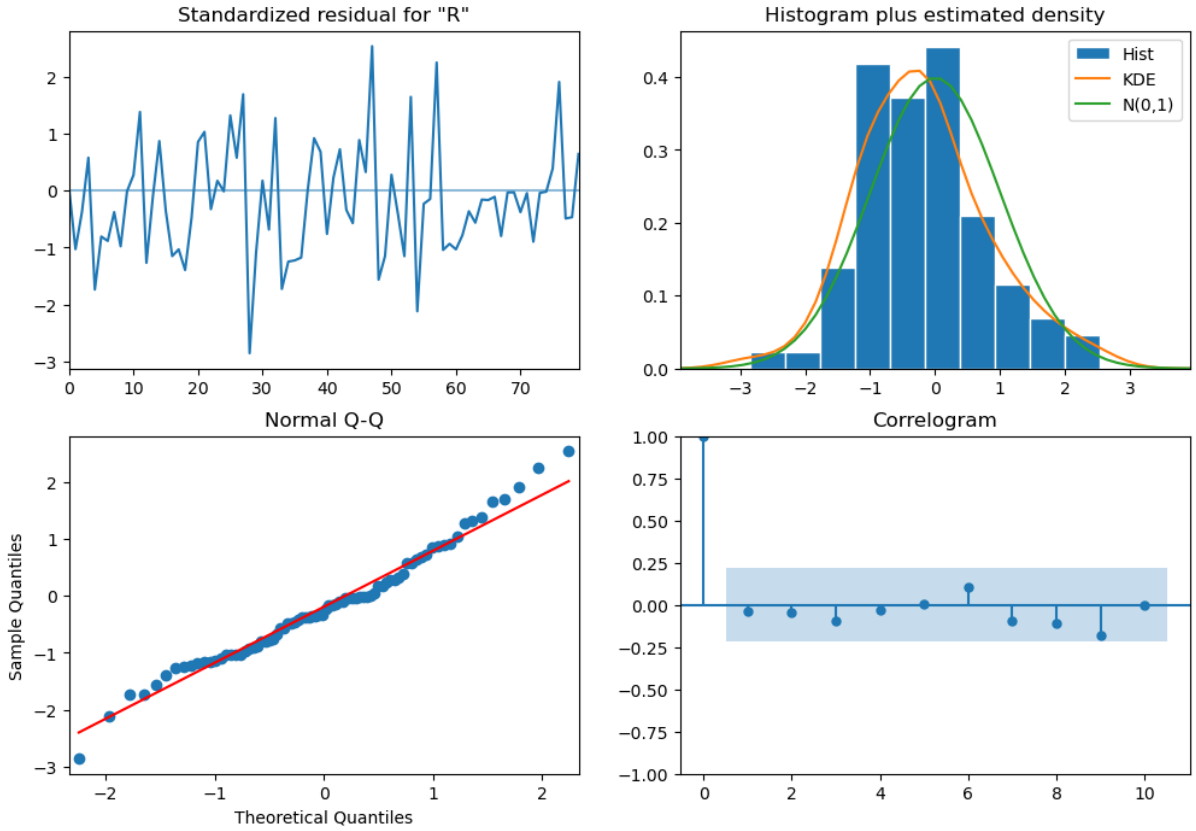
Building the most optimum model on the Full Data.

Using sm.tsa.statespace.SARIMAX for building the best model.

Summary result

```
                                SARIMAX Results
==========================================================================
==============
Dep. Variable:                              Rose   No. Observations:
96
Model:             SARIMAX(1, 1, 2)x(2, 0, 2, 6)   Log Likelihood
-375.438
Date:                           Tue, 08 Oct 2024   AIC
766.877
Time:                                   12:26:50   BIC
785.933
Sample:                                        0   HQIC
774.517
                                             - 96
Covariance Type:                             opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ar.L1         -0.0834      0.273     -0.306      0.760      -0.618       0.
451
ma.L1         -0.6860      0.281     -2.443      0.015      -1.237      -0.
136
ma.L2         -0.0521      0.243     -0.214      0.830      -0.529       0.
424
ar.S.L6       -0.8009      0.073    -10.900      0.000      -0.945      -0.
657
ar.S.L12      -0.6257      0.057    -10.895      0.000      -0.738      -0.
513
ma.S.L6        0.9848      0.150      6.569      0.000       0.691       1.
279
ma.S.L12       0.7691      0.222      3.461      0.001       0.334       1.
205
sigma2       602.2853    119.266      5.050      0.000     368.528     836.
042
==========================================================================
========
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):
2.31
Prob(Q):                              0.72   Prob(JB):
0.32
Heteroskedasticity (H):               1.13   Skew:
0.36
Prob(H) (two-sided):                  0.75   Kurtosis:
3.40
```

Plot for full data

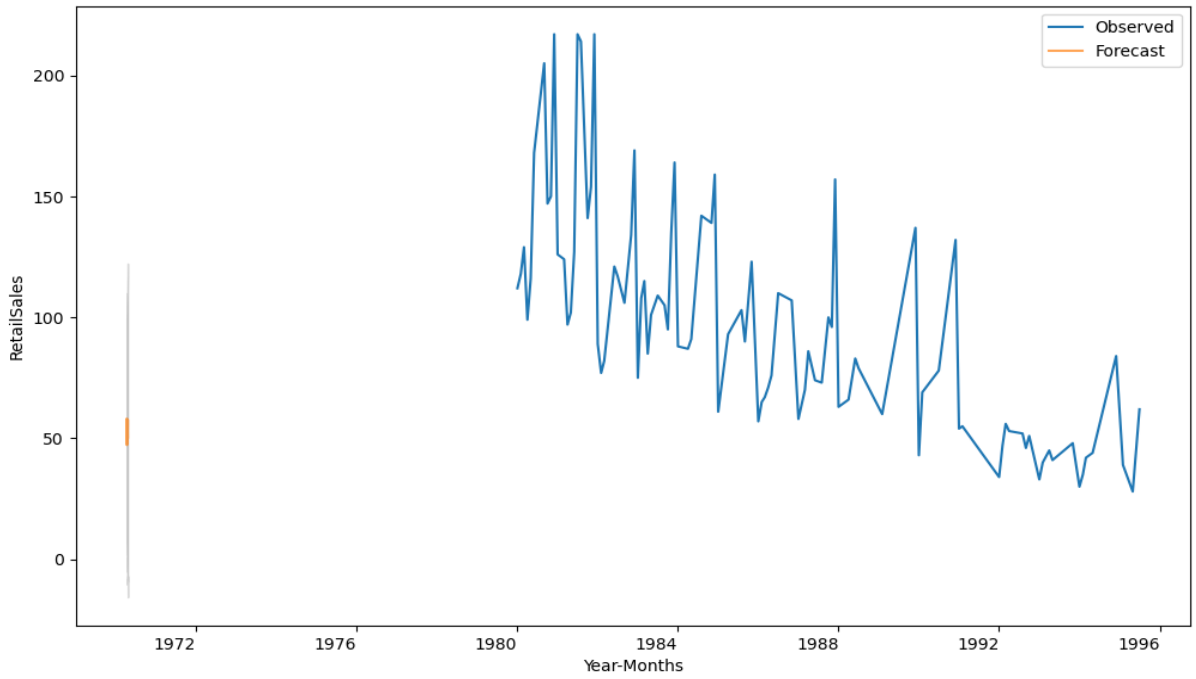


Evaluate the model on the whole and predict 12 months into the future

	Rose	mean	mean_se	mean_ci_lower	mean_ci_upper
96		50.591524	24.674697	2.230007	98.953042
97		55.514512	25.321015	5.886235	105.142789
98		58.133153	25.982504	7.208381	109.057924
99		47.253644	26.653380	-4.986021	99.493308
100		49.414822	27.309863	-4.111526	102.941170

The RMSE for full data is 33. It's the best version we got so far.

The plot on full data



Inference:

The forecast for next 12 months gave optimum value of 33. It’s the best version we got so far.

Context

As an analyst at ABC Estate Wines, we are presented with historical data encompassing the sales of different types of wines throughout the 20th century. These datasets originate from the same company but represent sales figures for distinct wine varieties. Our objective is to delve into the data, analyze trends, patterns, and factors influencing wine sales over the course of the century. By leveraging data analytics and forecasting techniques, we aim to gain actionable insights that can inform strategic decision-making and optimize sales strategies for the future.

Objective

The primary objective of this project is to analyze and forecast wine sales trends for the 20th century based on historical data provided by ABC Estate Wines. We aim to equip ABC Estate Wines with the necessary insights and foresight to enhance sales performance, capitalize on emerging market opportunities, and maintain a competitive edge in the wine industry.

Sample of the dataset

Sparkling.csv

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

The dataset has 187 records with 2 rows.

Check the summary statistics

	count	mean	std	min	25%	50%	75%	max
Sparkling	187.0	2402.417112	1295.11154	1070.0	1605.0	1874.0	2549.0	7242.0

Observation:

The Sparkling has a min of 107- and max of 7242. With median as 1874.

Check for null records

```
1 df.isnull().sum()
```

```
Sparkling    0
dtype: int64
```

There are 0 null values.

Check the duplicate records.

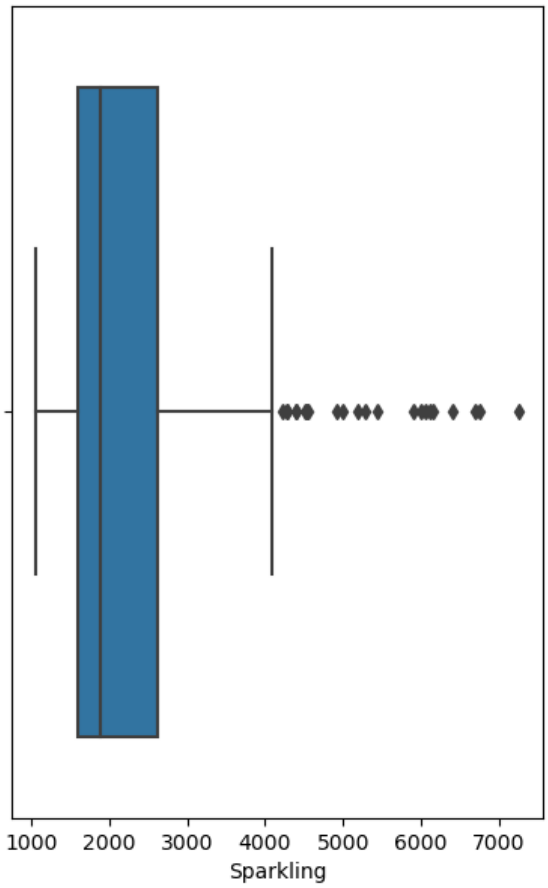
```
1 df.duplicated().sum()
```

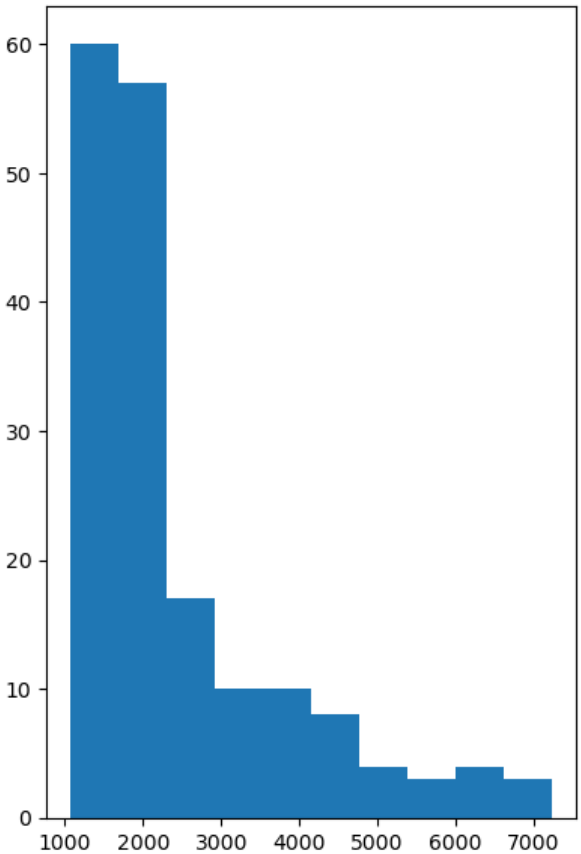
```
11
```

Treat the duplicate records by dropping them.

Univariant analysis

Sparkling





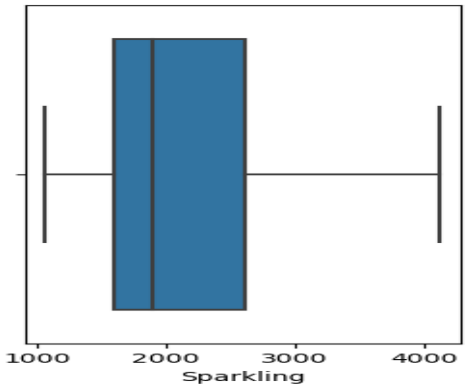
Observation:

Sparkling has extreme values we can see outliers. Its not evenly distributed. Will treat them with box plot technique.

Outlier treatment with box plot technique

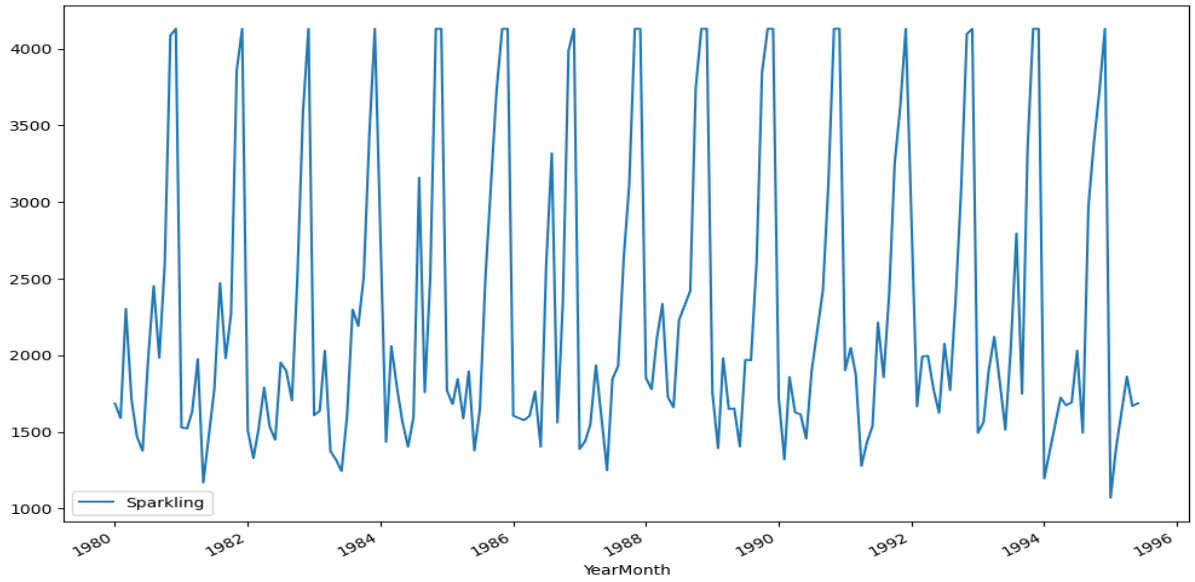
lower range -33.0 and upper range 217.0

lower range 91.125 and upper range 4130.125



Outlier datapoints are treated.

Plot the time series



We see an stable trend and seasonality which is not constant in nature.

Additive decomposition:

Additive decomposition is a method used in time series analysis to break down a time series into its individual components.

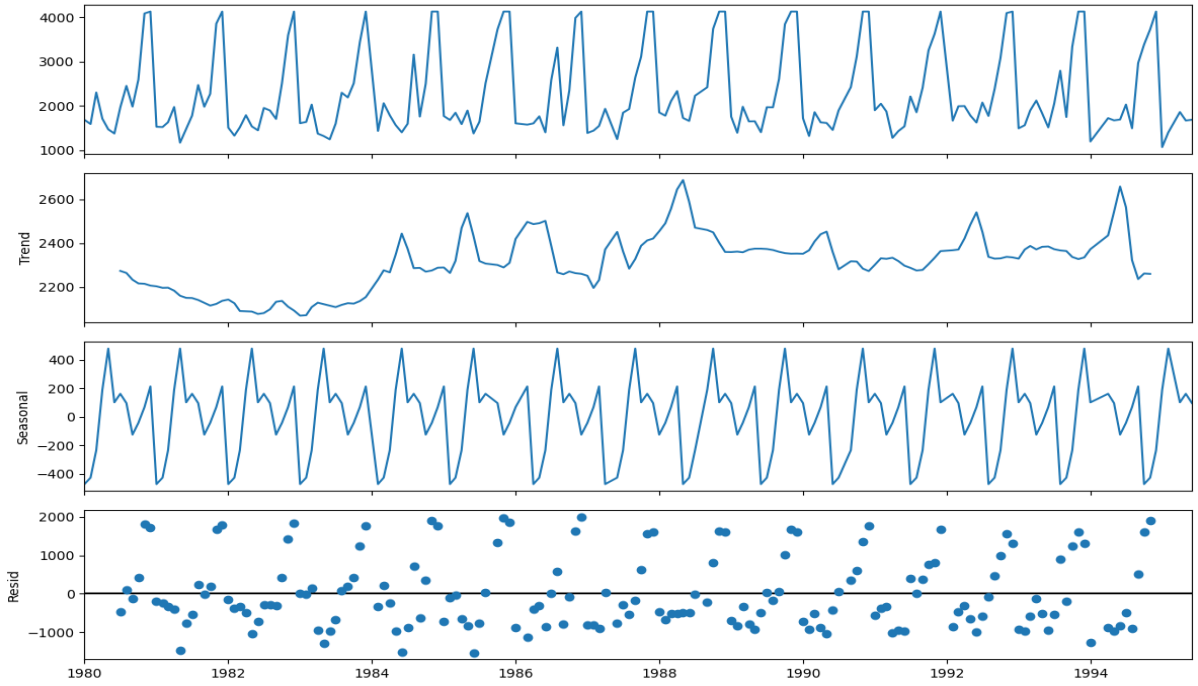
Components:

Trend: The direction of the time series data. It shows the pattern of increase or decrease over time.

Seasonality: The repeating short-term cycle of variations. It occurs due to seasonal flucation.

Cyclic: Fluctuations that are not fixed like seasonality but occur over longer periods.

Residual Component: The irregular or unpredictable variations that are not explained by the other components.



We see that the residuals are located around 0 from the plot of the residuals in the decomposition.

Table on trends and seasonality - Additive

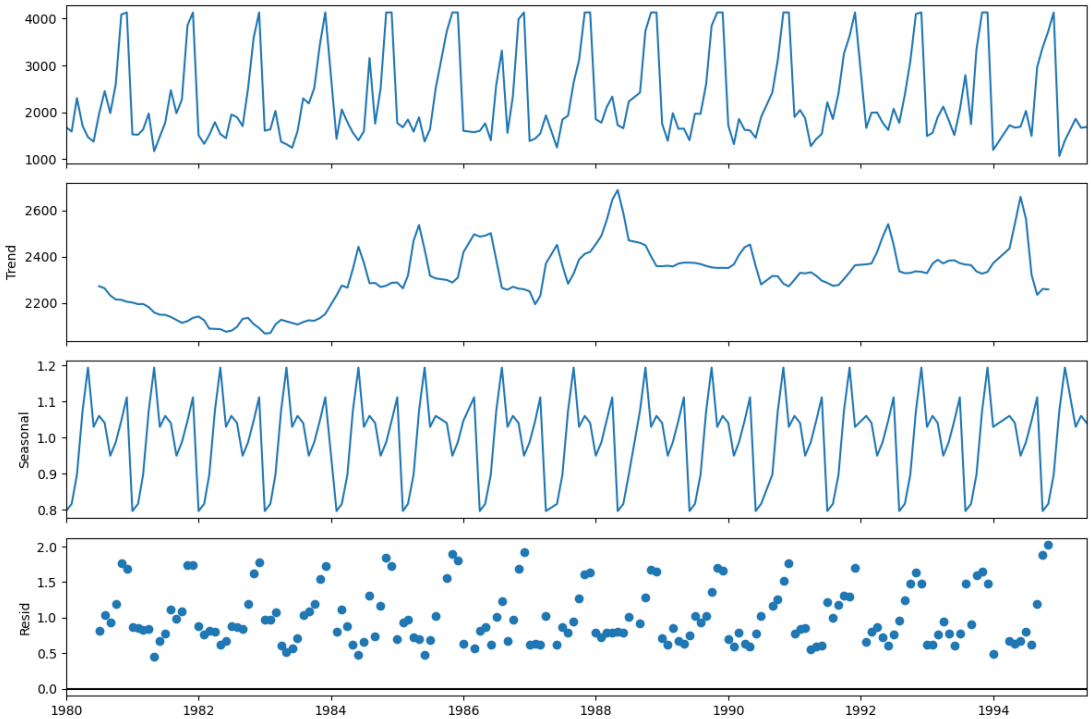
```
Trend
 YearMonth
1980-01-01              NaN
1980-02-01              NaN
1980-03-01              NaN
1980-04-01              NaN
1980-05-01              NaN
1980-06-01              NaN
1980-07-01      2273.260417
1980-08-01      2263.927083
1980-09-01      2233.135417
1980-10-01      2216.177083
1980-11-01      2214.635417
1980-12-01      2206.385417
Name: trend, dtype: float64

Seasonality
 YearMonth
1980-01-01     -472.986433
1980-02-01     -427.614409
1980-03-01     -234.927367
1980-04-01      188.979684
1980-05-01      478.117505
1980-06-01      100.544187
1980-07-01      160.966692
1980-08-01       95.600621
1980-09-01     -125.831299
1980-10-01      -43.069766
1980-11-01       67.680234
1980-12-01      212.540353
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01              NaN
1980-02-01              NaN
1980-03-01              NaN
1980-04-01              NaN
1980-05-01              NaN
1980-06-01              NaN
1980-07-01      -468.227109
1980-08-01        93.472296
1980-09-01      -123.304118
1980-10-01       422.892683
1980-11-01      1804.684350
1980-12-01      1711.199231
Name: resid, dtype: float64
```

Multiplicative decomposition:

Multiplicative decomposition is a method used in time series analysis to break down a time series as the product of its components. It has the same component as additive ones.



For the multiplicative series, we see that a lot of residuals are located around 1. Thus Multiplicative Decomposition is the right way to decompose the time series.

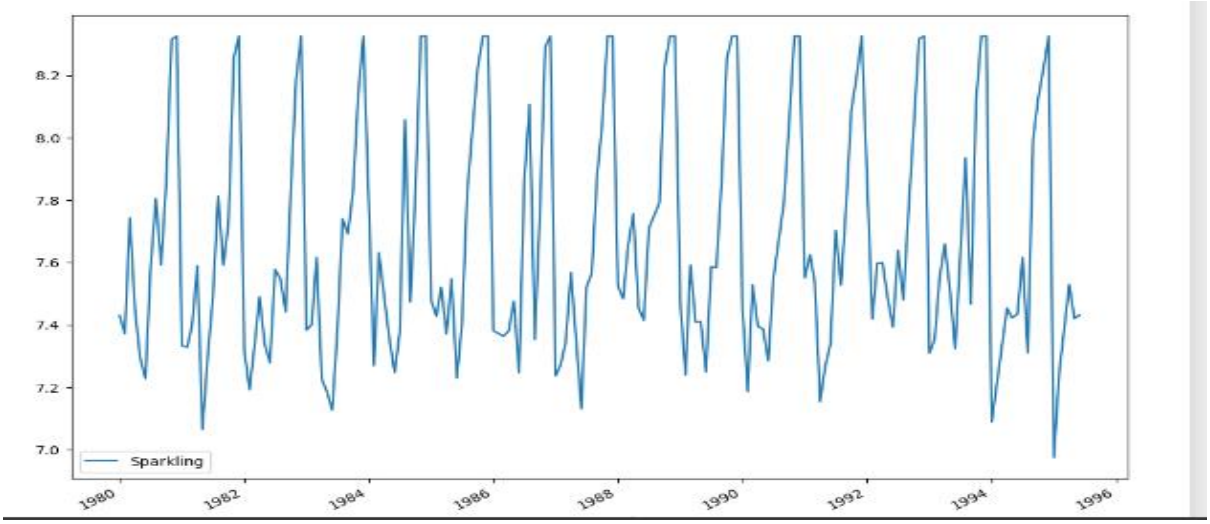
Table on trends and seasonality -Multiplicative

```
Trend
 YearMonth
1980-01-01             NaN
1980-02-01             NaN
1980-03-01             NaN
1980-04-01             NaN
1980-05-01             NaN
1980-06-01             NaN
1980-07-01     2273.260417
1980-08-01     2263.927083
1980-09-01     2233.135417
1980-10-01     2216.177083
1980-11-01     2214.635417
1980-12-01     2206.385417
Name: trend, dtype: float64

Seasonality
 YearMonth
1980-01-01     0.796340
1980-02-01     0.816153
1980-03-01     0.896984
1980-04-01     1.073721
1980-05-01     1.193584
1980-06-01     1.029294
1980-07-01     1.059849
1980-08-01     1.040105
1980-09-01     0.949441
1980-10-01     0.986770
1980-11-01     1.046577
1980-12-01     1.111182
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01          NaN
1980-02-01          NaN
1980-03-01          NaN
1980-04-01          NaN
1980-05-01          NaN
1980-06-01          NaN
1980-07-01     0.816000
1980-08-01     1.041736
1980-09-01     0.935748
1980-10-01     1.187092
1980-11-01     1.763320
1980-12-01     1.684599
Name: resid, dtype: float64
```

Log Transformation: comparison of original and transformed



Model building:

Steps:

1) Maintaining the order or sequencing is important in TSF. While choosing data for train or test, we need to ensure that sequencing is maintained.

2) We choose initial 80% of the data points as 'train' and rest as 'test'. We can choose major part of the initial period as 'train' and rest as 'test'.

Model 1: Linear Regression

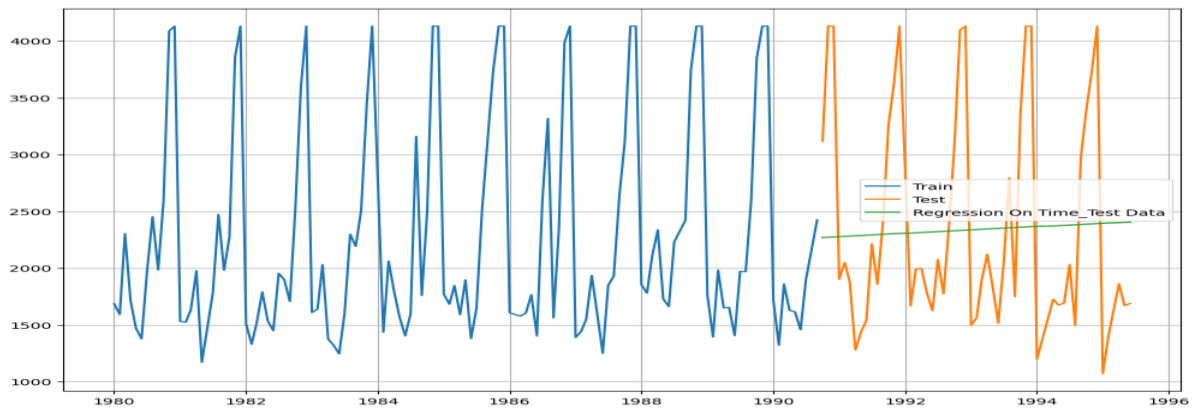
Steps:

1. First create a predictor (x) for the time series data which can be just incremental numbers for entire time period i.e. for both 'train' and 'test'

2. Note: If 'train' has 80 data points and test has 20 data points, x for train will range from 1 to 80 and x for 'test' will range from 81 to 100
3. Apply LinearRegression (y=f(x)) on the 'train' data and build a model and fit the model.
   a.
```
LinearRegression
LinearRegression()
```

4. Use the model to predict 'y' for the test period and get the required forecast.

Plot for Regression On Time_Test Data



Model Evaluation:

Use the Root mean square error method to calculate the accuracy.

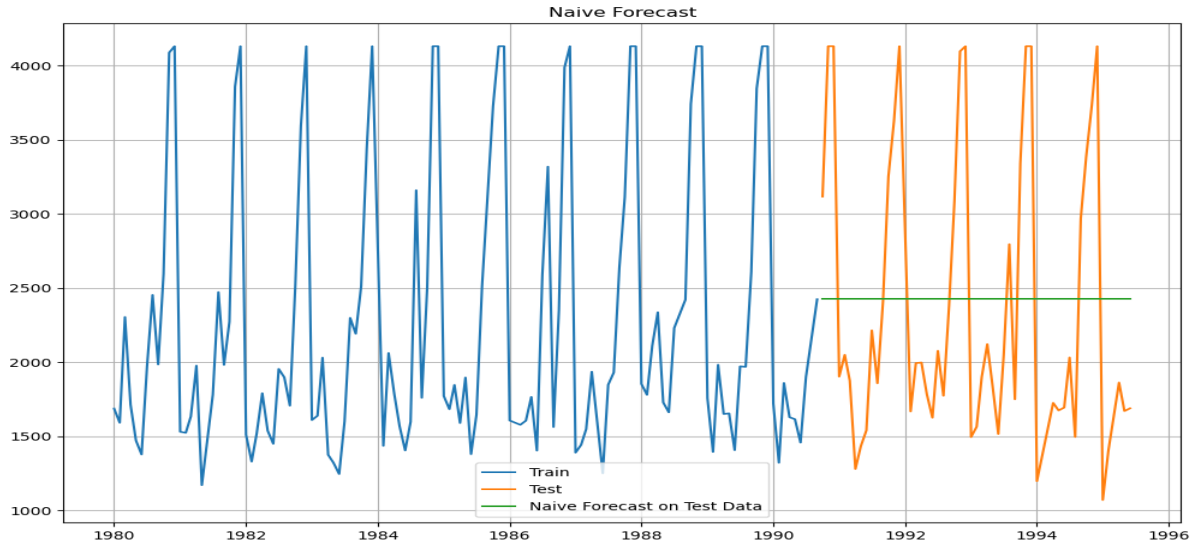
	Test RMSE
RegressionOnTime	965.462857

Model 2: Naive Approach

The prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.¶

Pick up the last value from the 'train' and use it as a forecast for future i.e. entire 'test' period.

Plot the graph for train and test data.



Model Evaluation:

Using RMSE Naïve approach gave 37.

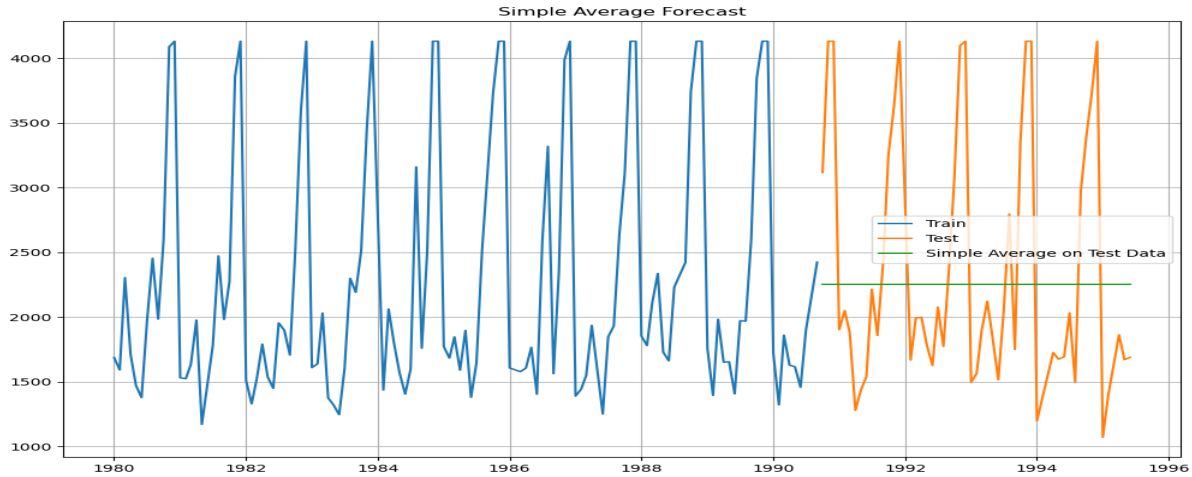
	Test RMSE
RegressionOnTime	965.462857
NaiveModel	961.301296

Method 3: Simple Average

We will forecast by using the average of the training values.¶

We are finding average of the value for entire 'train' period and use it as a forecast for future i.e. entire 'test' period.

Plot on Simple average



Model Evaluation:

The RMSE gave 63.

	Test RMSE
RegressionOnTime	965.462857
NaiveModel	961.301296
SimpleAverageModel	967.810983

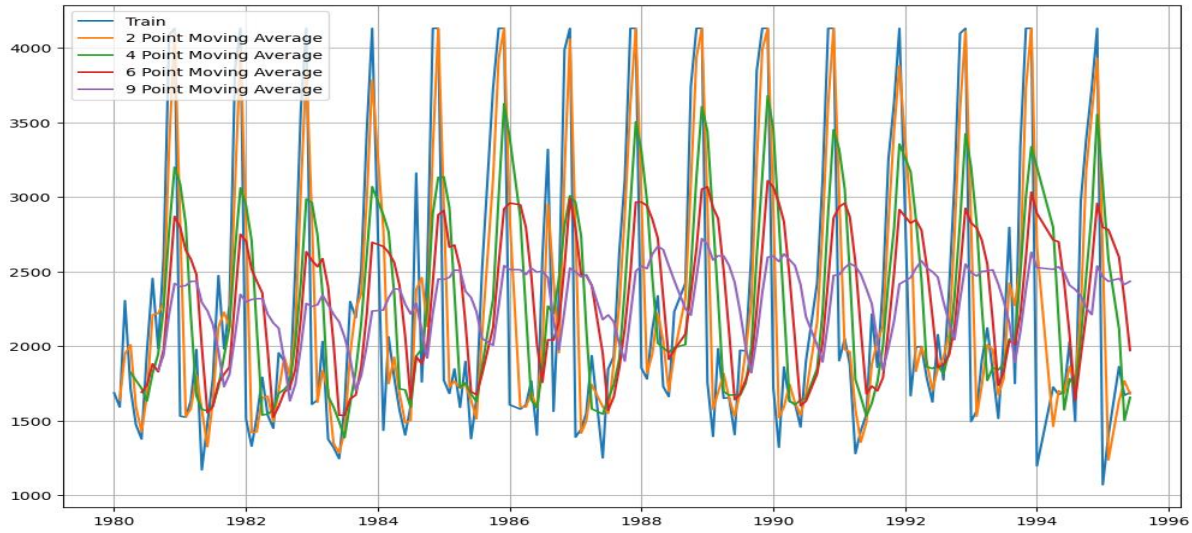
Method 4: Moving Average(MA)

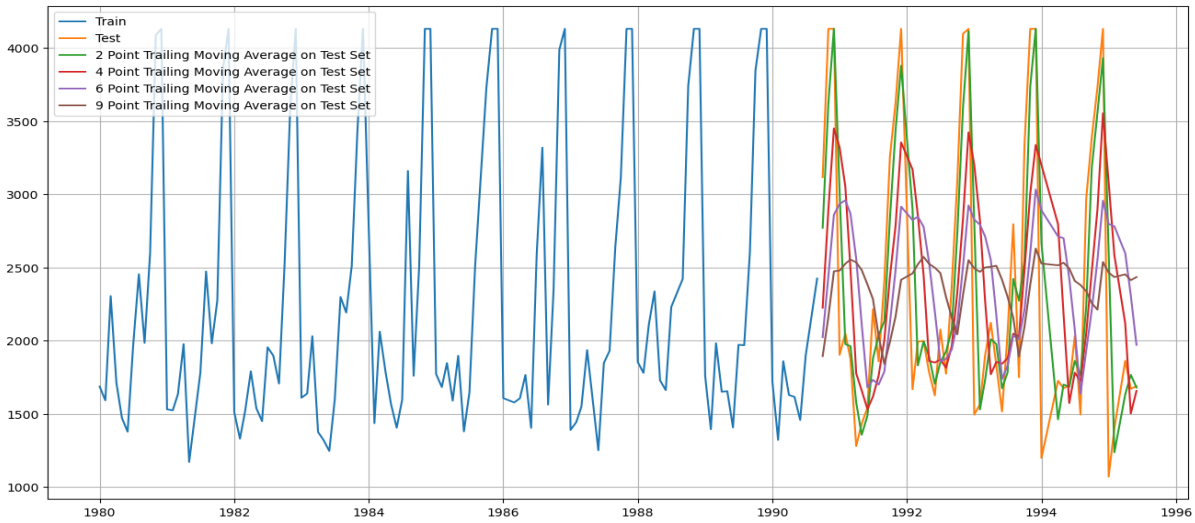
Steps:

1) Based on the time series pattern and business understanding we choose a window to apply moving average.

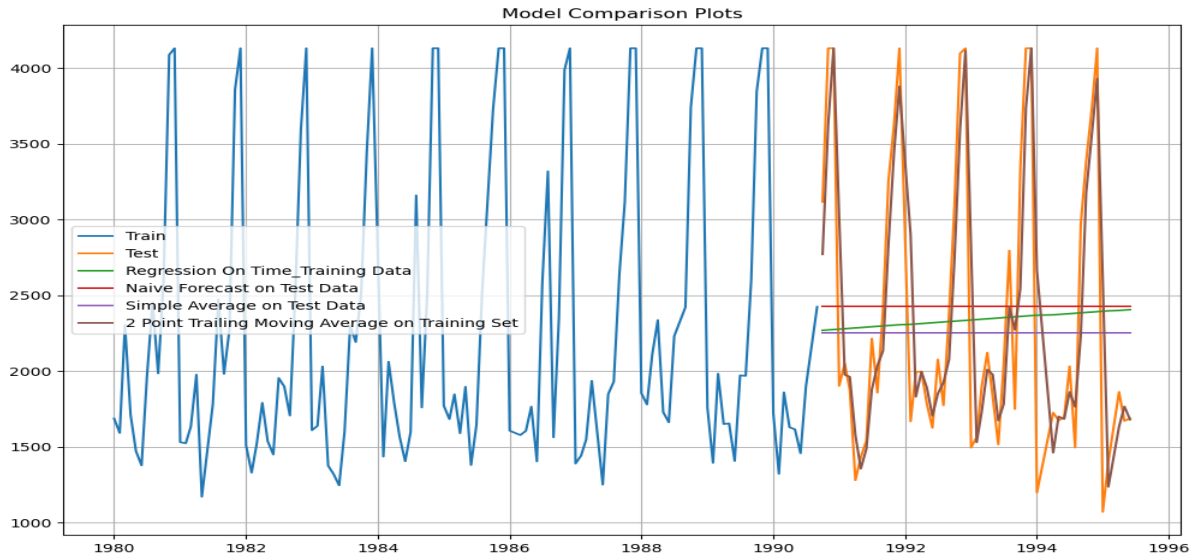
2) For the chosen window, we apply moving average from the start of train period and get forecast for 1 additional time period beyond train.

3) We use rolling function in Python to implement Moving Average.





Plotting on both the Training and Test data



Model Evaluation

The RMSE score for MA is as follows. We could see for 2point trailing MA we have best score.

	Test RMSE
RegressionOnTime	965.462857
NaiveModel	961.301296
SimpleAverageModel	967.810983
2pointTrailingMovingAverage	492.172935
4pointTrailingMovingAverage	836.448804
6pointTrailingMovingAverage	976.358671
9pointTrailingMovingAverage	1027.648292

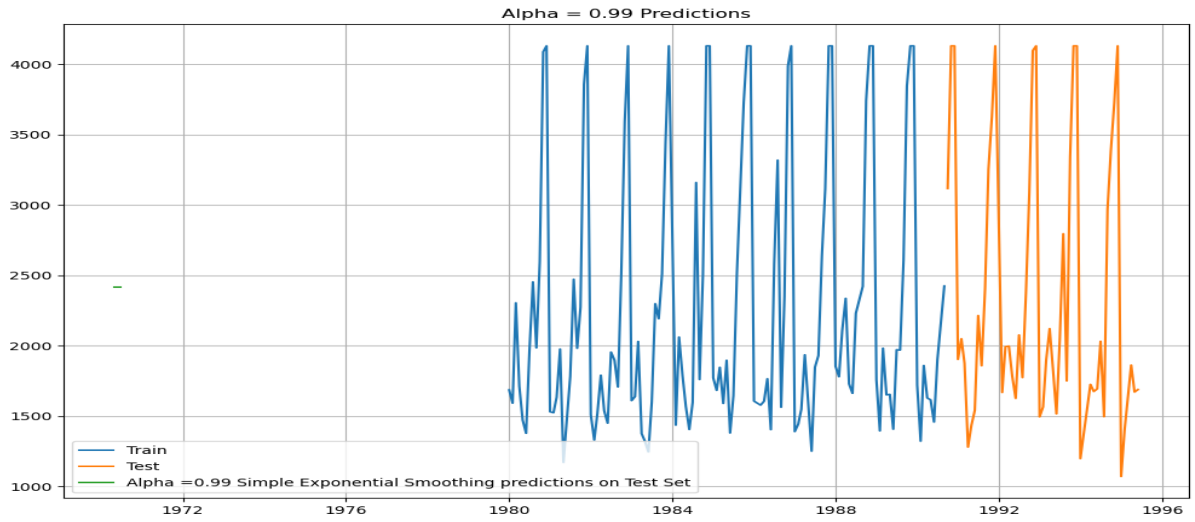
Exponential model

Simple exponential model building:

SES or one-parameter exponential smoothing is applicable to time series which do not contain either of trend or seasonality.

```
{'smoothing_level': 0.9800654219861422,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1684.3642593666143,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Plot the train and test and forecast values

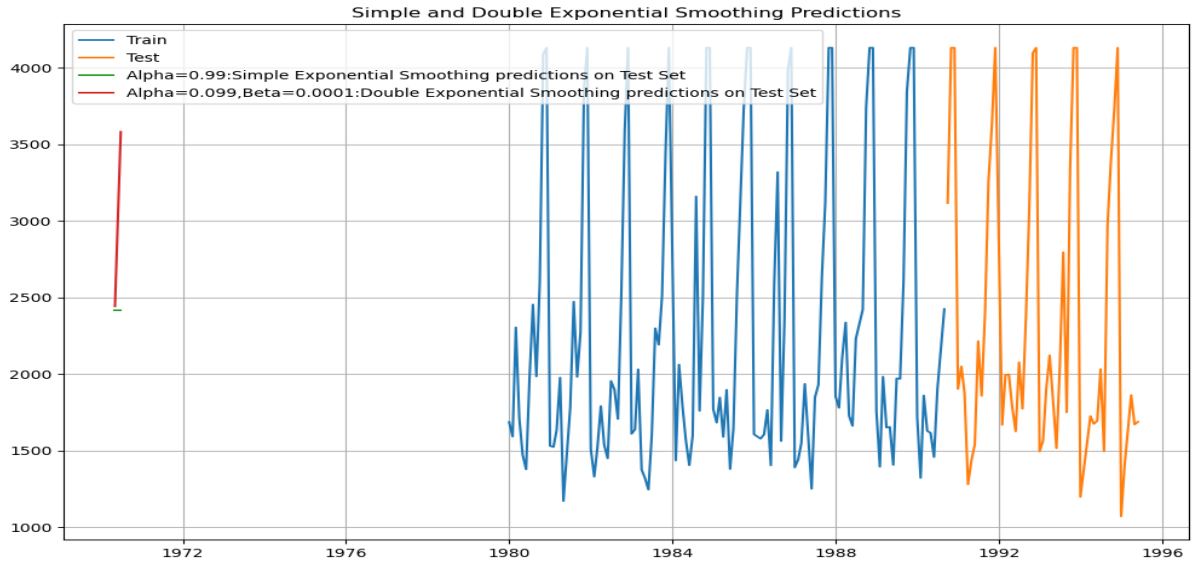


Model evaluation:

	Test RMSE
Alpha=0.99,SES	960.818517

Double exponential smoothing:

This method is applicable where trend is present in the data but no seasonality.



Model Evaluation:

	Test RMSE
Alpha=0.99,SES	960.818517
Alpha=1,Beta=0.0189:DES	1225.066988

Inference

Here, we see that the Simple Exponential Smoothing is doing well.

Triple Exponential smoothing:

This is an extension of Holt's method when seasonality is found in the data.
This is also known as three parameters exponential or triple exponential because of the three smoothing parameters α, β and γ. This is a general method and a true multi-step ahead forecast.

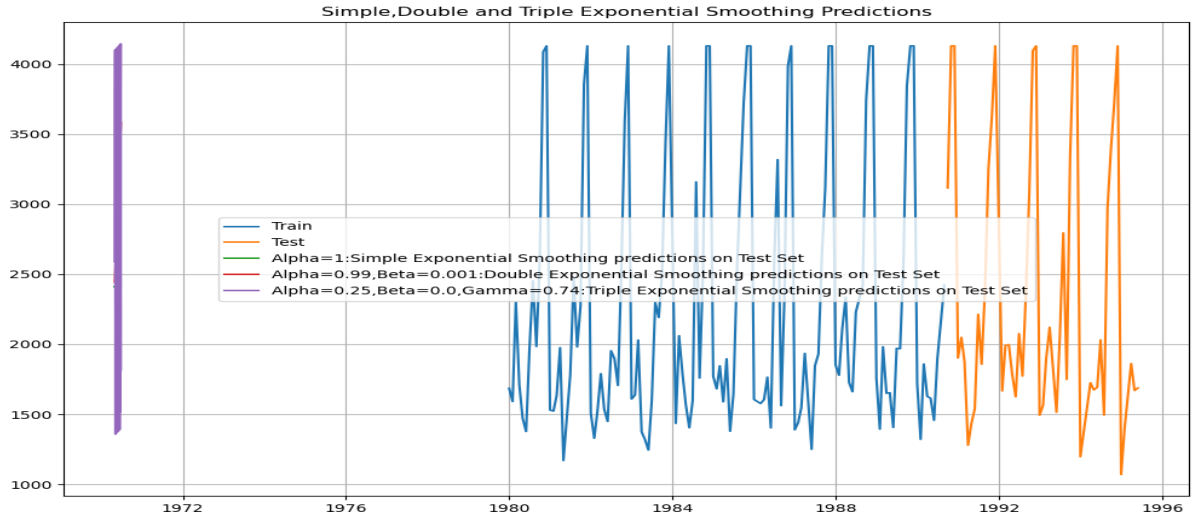
Holt-Winters - ETS(A, A, A) - Holt Winter's linear method with additive errors

The Holt-Winters ETS (A, A, A) method is a version of the Holt-Winters model that uses additive error, additive trend, and additive seasonality. It is used for time series forecasting when the data exhibits both trend and seasonality, and the seasonal variations are constant over time.

Components of ETS (A, A, A):

- Error (E): Additive error (A)
- Trend (T): Additive trend (A)

- Seasonality (S): Additive seasonality (A)



Model evaluation:

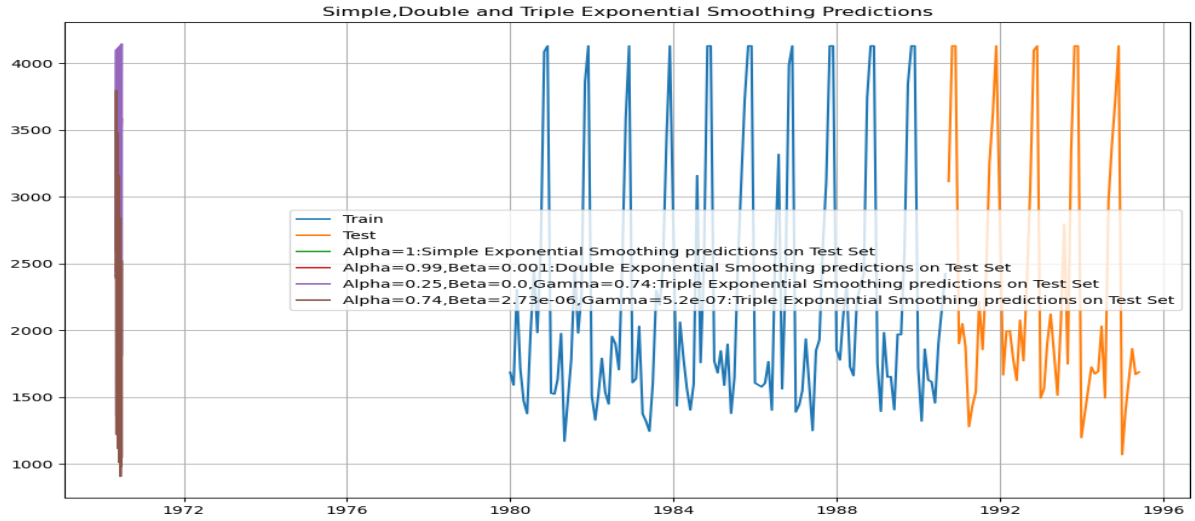
	Test RMSE
Alpha=0.99,SES	960.818517
Alpha=1,Beta=0.0189:DES	1225.066988
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	1258.659662

Holt-Winters - ETS(A, A, M) - Holt Winter's linear method

The **Holt-Winters ETS (A, A, M)** method is another variant of the Exponential Smoothing (ETS) model, where the error and trend are modelled additively, but the seasonality is modelled multiplicatively.

Components of ETS (A, A, M):

- **Error (E)**: Additive error (A)
- **Trend (T)**: Additive trend (A)
- **Seasonality (S)**: Multiplicative seasonality (M)



Model Evaluation:

	Test RMSE
Alpha=0.99,SES	960.818517
Alpha=1,Beta=0.0189:DES	1225.066988
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	1258.659662
Alpha=0.74,Beta=2.73e-06,Gamma=5.2e-07,Gamma=0:TES	1203.722393

Inference:

We see that the multiplicative seasonality model has done well when compared to the additive seasonality in Triple Exponential Smoothing model. But still simple exponential soothing model is better.

Check for Stationarity

A stationary time series has statistical properties (mean, variance, autocovariance) that do not change over time.

Augmented Dickey-Fuller (ADF) Test:

The ADF test is one of the most widely used statistical tests for stationarity.

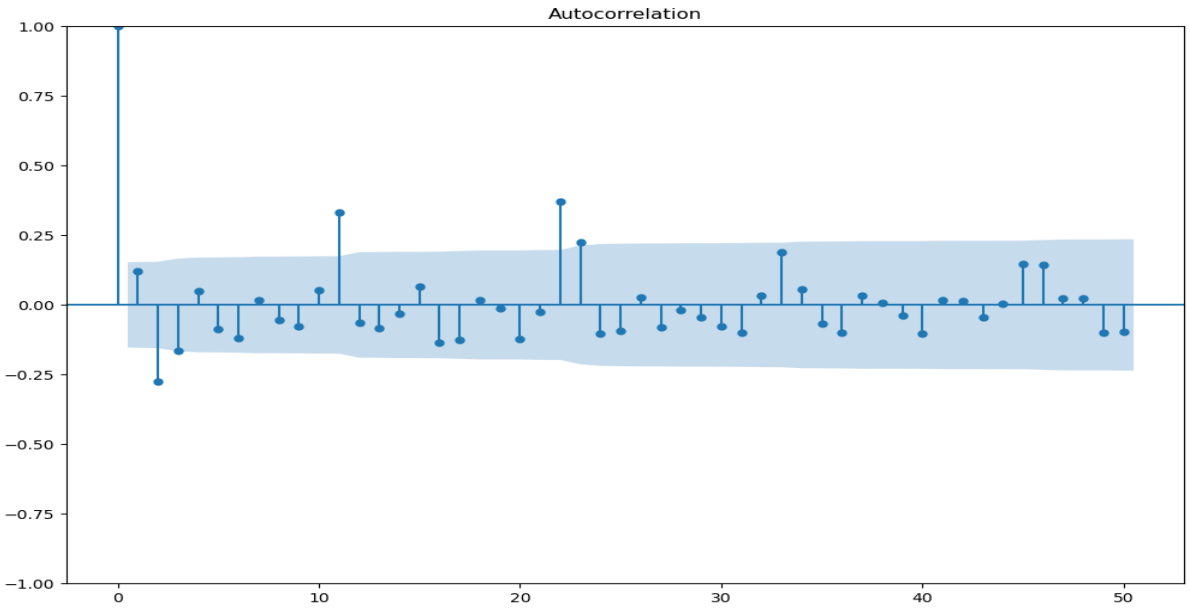
Steps to interpret ADF test:

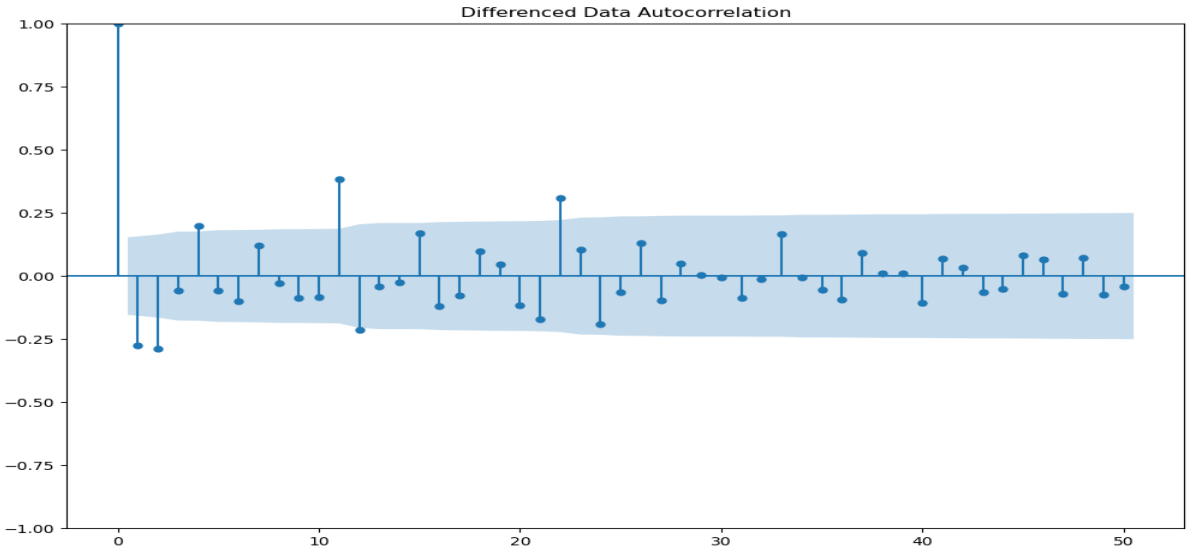
- Null hypothesis: The series has a unit root (non-stationary).
- Alternative hypothesis: The series is stationary.
- If the p-value is less than a chosen significance level (e.g., 0.05), the null hypothesis is rejected, and the series is considered stationary.

How to Make a Time Series Stationary (If It's Not)

- Differencing
- Log Transformation
- De-trending
- De-seasonalization

Plot the Autocorrelation function plots on the whole data.





ARIMA

ARIMA is suitable for non-seasonal data and focuses on trends and cycles.

Build an Automated version of an ARMA model for which the best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

Based on values of p,q,r the model is formed.

```
Some parameter combinations for the Model...
Model: (0, 0, 1)
Model: (0, 0, 2)
Model: (1, 0, 0)
Model: (1, 0, 1)
Model: (1, 0, 2)
Model: (2, 0, 0)
Model: (2, 0, 1)
Model: (2, 0, 2)
```

Fit the ARIMA model.

The Param and AIC in ascending order after fit.

	param	AIC
8	(2, 0, 2)	1990.533366
5	(1, 0, 2)	1990.548944
6	(2, 0, 0)	1990.569915
2	(0, 0, 2)	1990.995447
7	(2, 0, 1)	1992.466659
4	(1, 0, 1)	1993.104945
1	(0, 0, 1)	1993.679799
3	(1, 0, 0)	2001.923237
0	(0, 0, 0)	2035.531039

The auto ARIMA summary table

```
                               SARIMAX Results
==========================================================================
===
Dep. Variable:                Sparkling   No. Observations:
123
Model:                 ARIMA(2, 0, 2)   Log Likelihood                -989.
267
Date:                Fri, 11 Oct 2024   AIC                           1990.
533
Time:                        15:00:36   BIC                           2007.
406
Sample:                             0   HQIC                          1997.
387
                                  - 123
Covariance Type:                  opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
const       2253.1245    145.215     15.516      0.000    1968.508    2537.
741
ar.L1         -0.3747      0.243     -1.541      0.123      -0.851       0.
102
ar.L2         -0.1953      0.160     -1.220      0.222      -0.509       0.
118
ma.L1          1.0705      0.222      4.826      0.000       0.636       1.
505
ma.L2          0.5932      0.120      4.929      0.000       0.357       0.
829
sigma2      5.633e+05   1.01e+05      5.605      0.000    3.66e+05     7.6e
+05
==========================================================================
========
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):
2.69
Prob(Q):                              0.91   Prob(JB):
0.26
Heteroskedasticity (H):               0.99   Skew:
0.14
Prob(H) (two-sided):                  0.97   Kurtosis:
2.34
==========================================================================
========
```

Evaluation:

Based on mean squared error its 964.48

	RMSE
ARIMA(2,0,2)	964.489449

After changing the values of p,q,r the values looks different.

	param	AIC
7	(2, 1, 1)	1980.440490
8	(2, 1, 2)	1982.294502
5	(1, 1, 2)	1982.525105
2	(0, 1, 2)	1983.712121
4	(1, 1, 1)	1990.782435
6	(2, 1, 0)	2006.726679
0	(0, 1, 0)	2017.052322
3	(1, 1, 0)	2019.046486
1	(0, 1, 1)	2019.061227

```
==========================================================================
===
Dep. Variable:                Sparkling   No. Observations:
123
Model:                  ARIMA(2, 1, 1)   Log Likelihood                -986.
220
Date:                 Fri, 11 Oct 2024   AIC                           1980.
440
Time:                         15:00:37   BIC                           1991.
657
Sample:                              0   HQIC                          1984.
996
                                 - 123
Covariance Type:                   opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ar.L1          0.6690      0.085      7.905      0.000       0.503       0.
835
ar.L2         -0.3095      0.106     -2.917      0.004      -0.517      -0.
102
ma.L1         -1.0000      0.166     -6.018      0.000      -1.326      -0.
674
sigma2      5.902e+05   2.82e-07   2.09e+12      0.000     5.9e+05     5.9e
+05
==========================================================================
========
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):
1.51
Prob(Q):                              0.96   Prob(JB):
0.47
Heteroskedasticity (H):               0.90   Skew:
-0.06
Prob(H) (two-sided):                  0.73   Kurtosis:
2.47
==========================================================================
========
```

Evaluation:

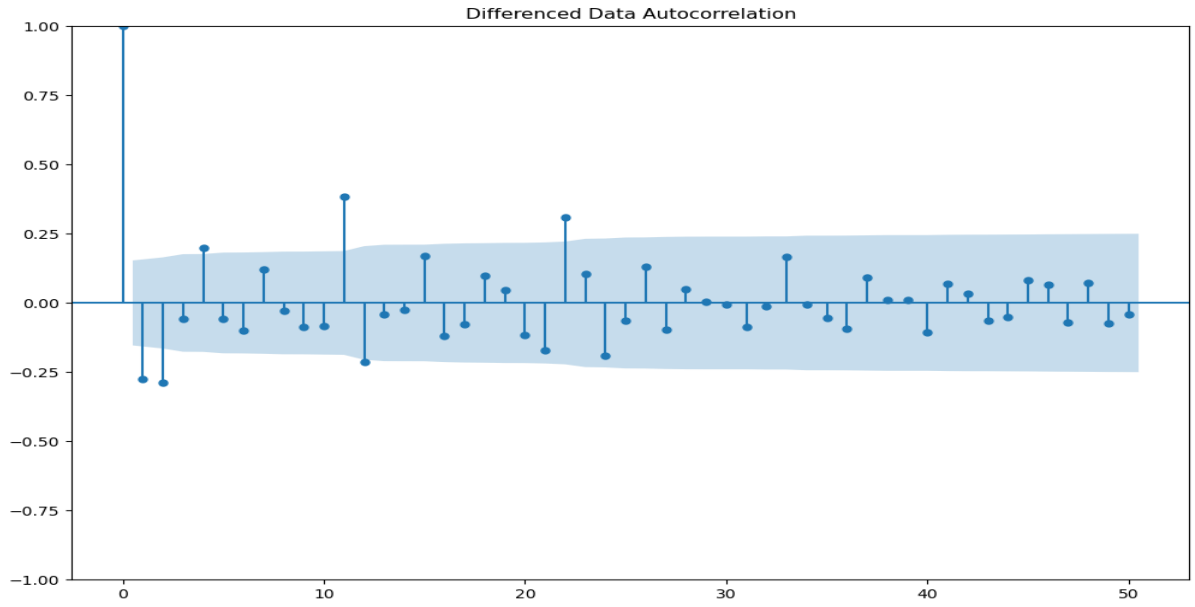
	RMSE
ARIMA(2,0,2)	964.489449
ARIMA(2,1,1)	962.023946

SARIMA:

SARIMA is an extension of ARIMA that incorporates seasonality, making it better for time series data with seasonal fluctuations.

Build an Automated version of a SARIMA model for which the best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC).

ACF plot for Differenced Data Autocorrelation



Based on p,q, d, D values the model.

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 6)
Model: (0, 1, 2)(0, 0, 2, 6)
Model: (1, 1, 0)(1, 0, 0, 6)
Model: (1, 1, 1)(1, 0, 1, 6)
Model: (1, 1, 2)(1, 0, 2, 6)
Model: (2, 1, 0)(2, 0, 0, 6)
Model: (2, 1, 1)(2, 0, 1, 6)
Model: (2, 1, 2)(2, 0, 2, 6)
```

After the best parameter estimate:

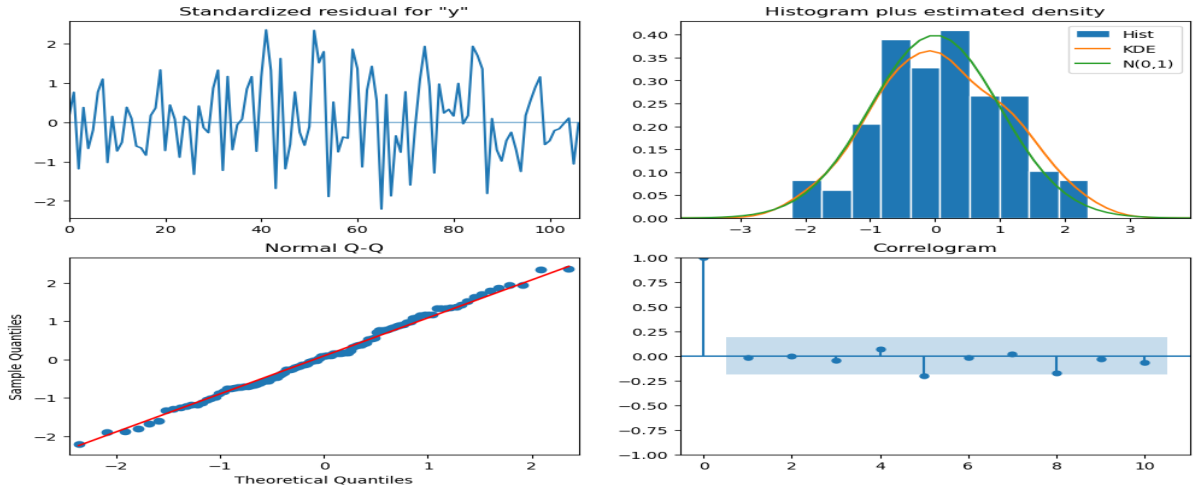
]:

	param	seasonal	AIC
77	(2, 1, 2)	(1, 0, 2, 6)	1704.080122
80	(2, 1, 2)	(2, 0, 2, 6)	1704.127493
26	(0, 1, 2)	(2, 0, 2, 6)	1709.896420
53	(1, 1, 2)	(2, 0, 2, 6)	1711.540312
23	(0, 1, 2)	(1, 0, 2, 6)	1712.390780

The summary result

```
==========================================================================
==============
Dep. Variable:                                  y   No. Observations:
123
Model:             SARIMAX(2, 1, 2)x(1, 0, 2, 6)   Log Likelihood
-844.040
Date:                           Fri, 11 Oct 2024   AIC
1704.080
Time:                                   15:01:00   BIC
1725.463
Sample:                                        0   HQIC
1712.748
                                           - 123
Covariance Type:                             opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ar.L1          0.6839      0.176      3.882      0.000       0.339       1.
029
ar.L2         -0.4853      0.106     -4.587      0.000      -0.693      -0.
278
ma.L1         -1.5467      0.289     -5.361      0.000      -2.112      -0.
981
ma.L2          0.5151      0.282      1.824      0.068      -0.038       1.
069
ar.S.L6       -0.8408      0.108     -7.818      0.000      -1.052      -0.
630
ma.S.L6        0.4475      0.124      3.610      0.000       0.205       0.
690
ma.S.L12       0.1760      0.132      1.338      0.181      -0.082       0.
434
sigma2      3.643e+05   7.26e+04      5.019      0.000    2.22e+05    5.07e
+05
==========================================================================
========
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):
1.21
Prob(Q):                              0.89   Prob(JB):
0.55
Heteroskedasticity (H):               1.65   Skew:
0.07
Prob(H) (two-sided):                  0.14   Kurtosis:
```

The plot diagnostics



Evaluation:

The RMSE gives 866.

	RMSE
ARIMA(2,0,2)	964.489449
ARIMA(2,1,1)	962.023946
SARIMA(2,1,2)(1, 0, 2, 6)	866.920652

Inference:

The Sarima model is giving optimum of RMSE 866.

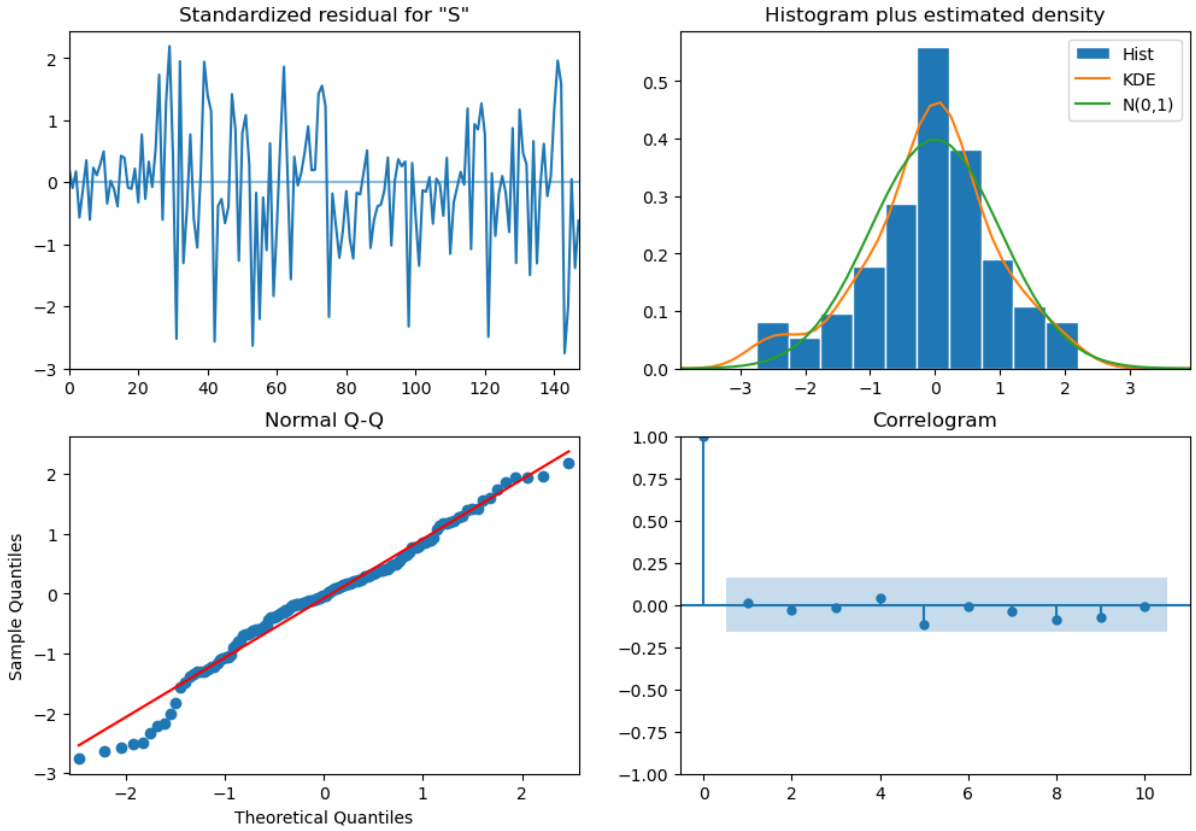
Building the most optimum model on the Full Data.

Using sm.tsa.statespace.SARIMAX for building the best model.

Summary result

```
==========================================================================
==============
Dep. Variable:                          Sparkling   No. Observations:
164
Model:             SARIMAX(2, 1, 2)x(1, 0, 2, 6)   Log Likelihood
-1187.348
Date:                           Fri, 11 Oct 2024   AIC
2390.696
Time:                                   15:01:02   BIC
2414.673
Sample:                                        0   HQIC
2400.438
                                           - 164
Covariance Type:                             opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ar.L1          0.8016      0.134      5.997      0.000       0.540       1.
064
ar.L2         -0.4701      0.072     -6.523      0.000      -0.611      -0.
329
ma.L1         -1.6313      0.130    -12.529      0.000      -1.886      -1.
376
ma.L2          0.6535      0.129      5.053      0.000       0.400       0.
907
ar.S.L6        0.2345      0.210      1.118      0.264      -0.177       0.
646
ma.S.L6       -0.7660      0.212     -3.618      0.000      -1.181      -0.
351
ma.S.L12      -0.2824      0.222     -1.273      0.203      -0.717       0.
153
sigma2      4.633e+05   1.02e+05      4.531      0.000    2.63e+05    6.64e
+05
==========================================================================
========
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):
4.59
Prob(Q):                              0.85   Prob(JB):
0.10
Heteroskedasticity (H):               1.11   Skew:
-0.39
Prob(H) (two-sided):                  0.72   Kurtosis:
3.38
```

Plot for full data

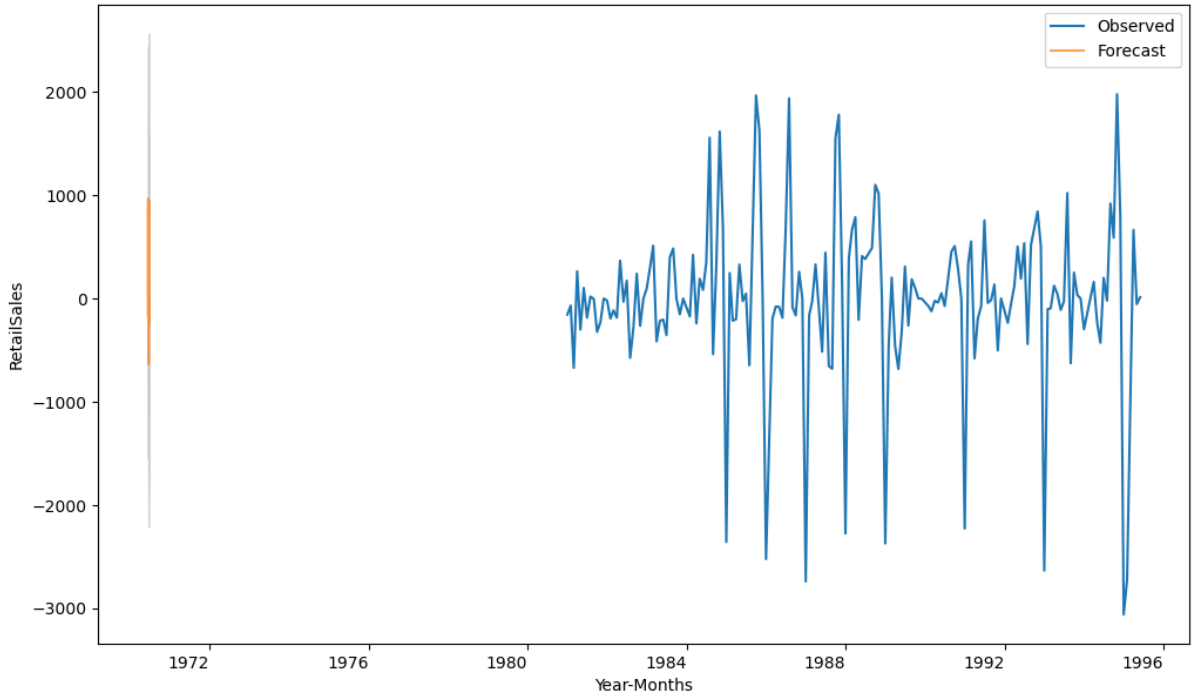


Evaluate the model on the whole and predict 12 months into the future

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
164	-156.389616	710.551684	-1549.045326	1236.266094
165	868.039002	721.039446	-545.172344	2281.250347
166	969.098139	753.922280	-508.562378	2446.758656
167	-24.150289	784.811021	-1562.351625	1514.051048
168	-454.313371	786.742409	-1996.300157	1087.673415

The RMSE for full data is 707. It’s the best version we got so far.

The plot on full data



Inference:

The forecast for next 12 months gave optimum value of 707. It's the best version we got so far.

Actionable Insights and Recommendations:

1. **Trend Analysis**: The metric represents sales, customer engagement, or any performance indicator, understanding the peaks and troughs helps in identifying the periods of strong or weak performance.
2. **Seasonality**: The recurring patterns could suggest seasonality. A deeper analysis uncovers these patterns repeat seasonally. This could inform inventory management or promotional planning.
3. **Forecasting**: Implementing a forecasting model (e.g., ARIMA) can predict future values of the "Rose"/"Sparkling" metric, helping to make informed decisions about resource allocation, production, or marketing.