Time Series Forecasting Project

Content

1. Read the data as an appropriate Time Series data and plot the data.
2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.
3. Split the data into training and test. The test data should start in 1991.
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.
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.
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.
7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.
8. Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.
9. Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.
10. Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

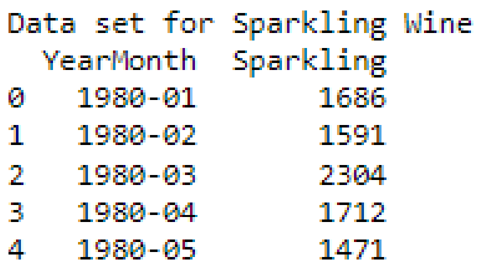
Problem:

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

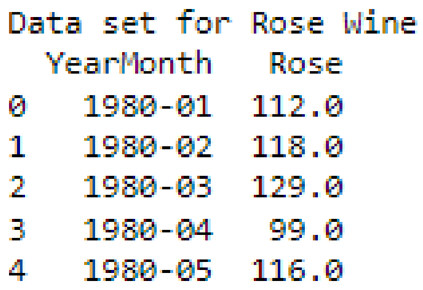
Data set for the Problem: Sparkling.csv and Rose.csv

1. Read the data as an appropriate Time Series data and plot the data.

Importing Sparkling and Rose wine sale data:

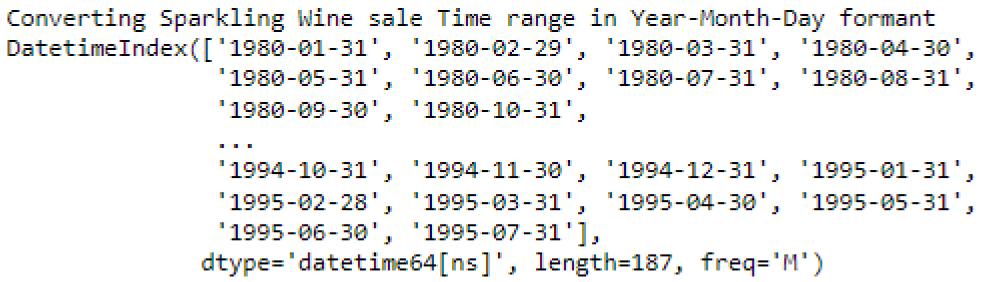


```
Data set for Sparkling Wine
  YearMonth  Sparkling
0   1980-01       1686
1   1980-02       1591
2   1980-03       2304
3   1980-04       1712
4   1980-05       1471
```

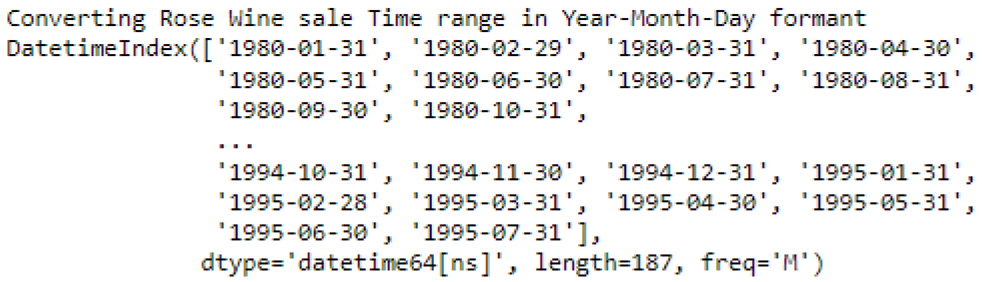


```
Data set for Rose Wine
  YearMonth   Rose
0   1980-01  112.0
1   1980-02  118.0
2   1980-03  129.0
3   1980-04   99.0
4   1980-05  116.0
```

Converting Sparkling and Rose Wine sale Time range in Year-Month-Day formant

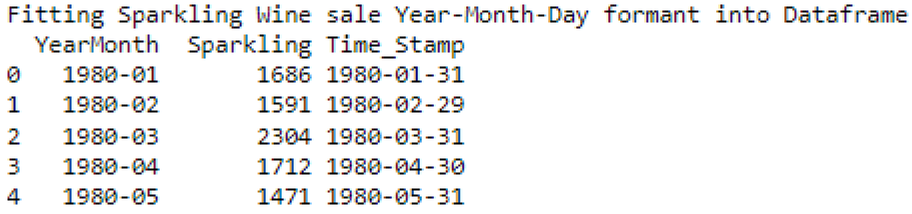


```
Converting Sparkling Wine sale Time range in Year-Month-Day formant
DatetimeIndex(['1980-01-31', '1980-02-29', '1980-03-31', '1980-04-30',
               '1980-05-31', '1980-06-30', '1980-07-31', '1980-08-31',
               '1980-09-30', '1980-10-31',
               ...
               '1994-10-31', '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')
```

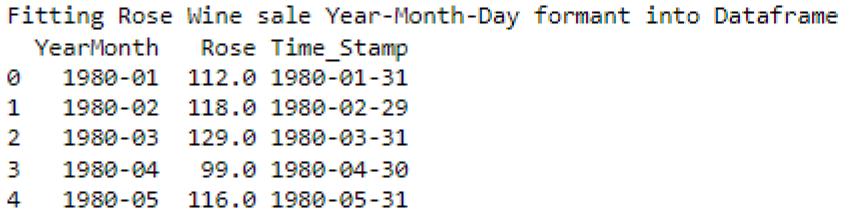


```
Converting Rose Wine sale Time range in Year-Month-Day formant
DatetimeIndex(['1980-01-31', '1980-02-29', '1980-03-31', '1980-04-30',
               '1980-05-31', '1980-06-30', '1980-07-31', '1980-08-31',
               '1980-09-30', '1980-10-31',
               ...
               '1994-10-31', '1994-11-30', '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')
```

Fitting Sparkling and Rose Wine sale Year-Month-Day formant into Dataframe

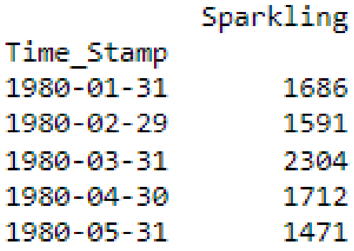


```
Fitting Sparkling Wine sale Year-Month-Day formant into Dataframe
  YearMonth  Sparkling Time_Stamp
0   1980-01       1686 1980-01-31
1   1980-02       1591 1980-02-29
2   1980-03       2304 1980-03-31
3   1980-04       1712 1980-04-30
4   1980-05       1471 1980-05-31
```

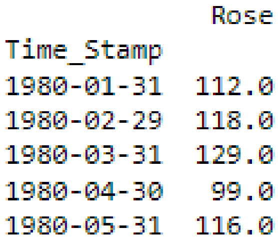


```
Fitting Rose Wine sale Year-Month-Day formant into Dataframe
  YearMonth   Rose Time_Stamp
0   1980-01  112.0 1980-01-31
1   1980-02  118.0 1980-02-29
2   1980-03  129.0 1980-03-31
3   1980-04   99.0 1980-04-30
4   1980-05  116.0 1980-05-31
```

Droping unwanted column and fixing index column for the both the dataset



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

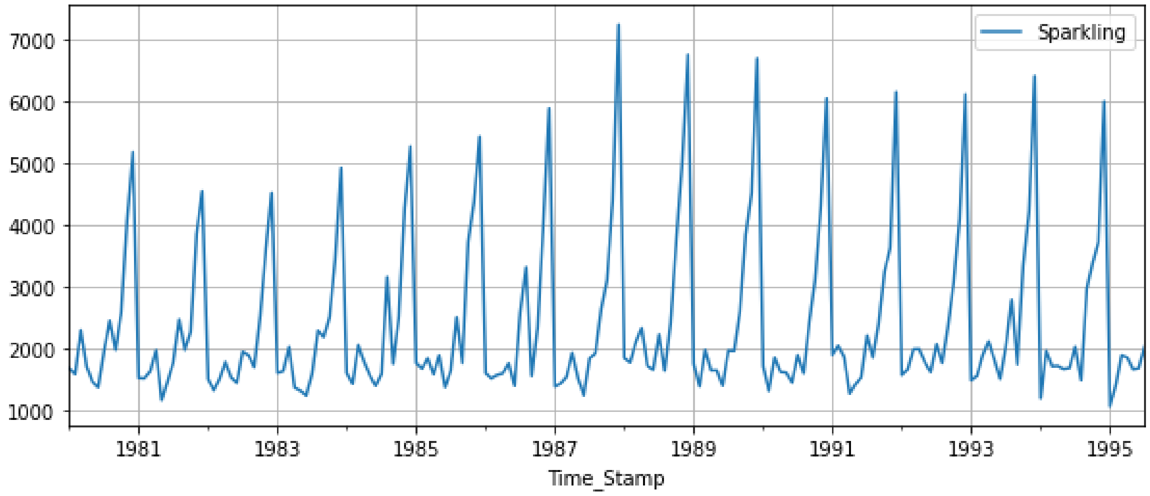


```
             Rose
Time_Stamp
1980-01-31  112.0
1980-02-29  118.0
1980-03-31  129.0
1980-04-30   99.0
1980-05-31  116.0
```

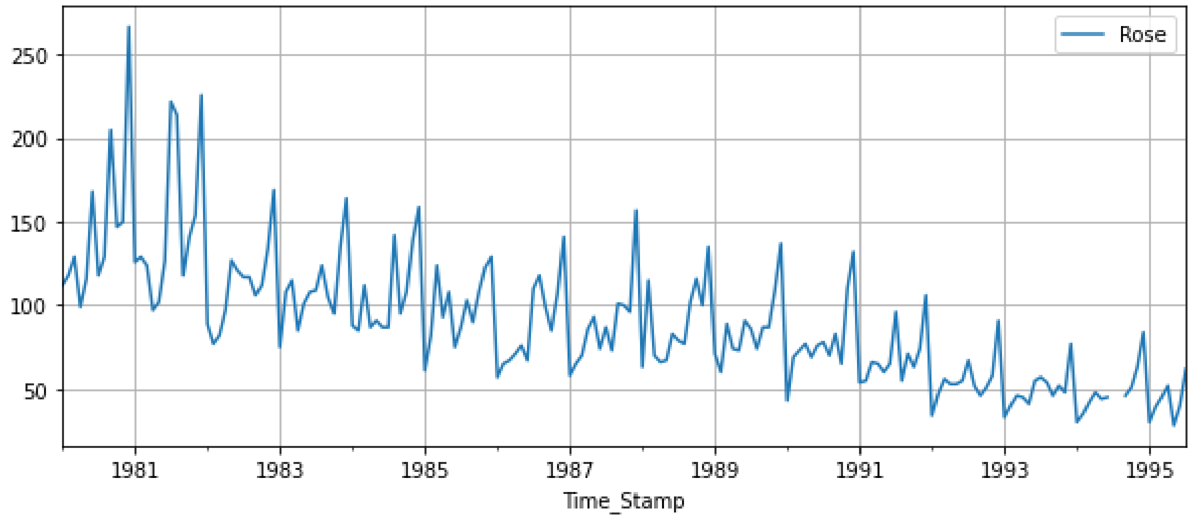
Now, we have our data ready for the Time Series Analysis.

Plot the Time Series to understand the behaviour of the data

Plot for Sparkling Wine Sale

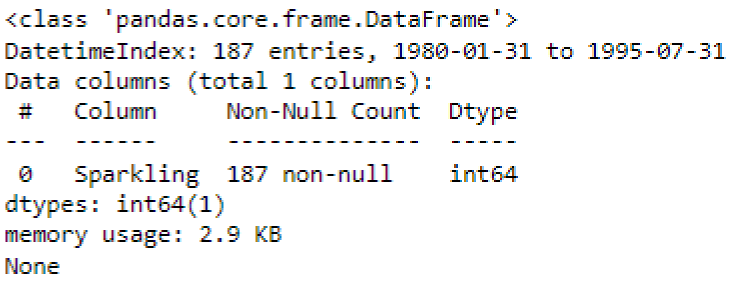


Plot for Rose Wine Sale

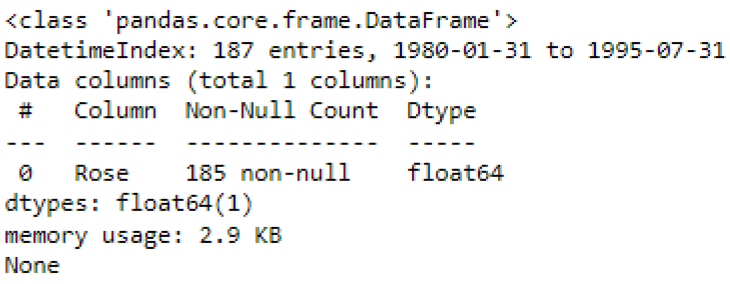


2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

Information of Data

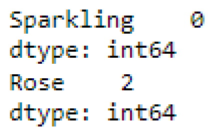


```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Sparkling  187 non-null    int64
dtypes: int64(1)
memory usage: 2.9 KB
None
```



```
<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    185 non-null    float64
dtypes: float64(1)
memory usage: 2.9 KB
None
```

Finding the missing Value



```
Sparkling    0
dtype: int64
Rose    2
dtype: int64
```

There is missing value in rose wine sales data. Using Interpolate() Method I am filling null value.

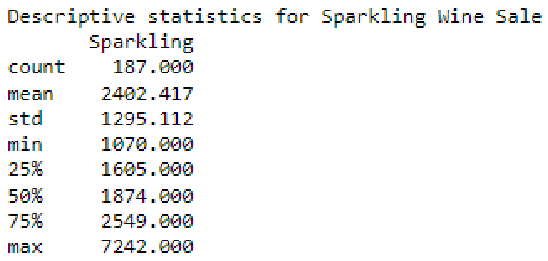
This method is more complex than the above fillna() method. It consists of different methodologies, including ‘linear’, ‘quadratic’, ‘nearest’.

Interpolation is a powerful method to fill missing values in time-series data.

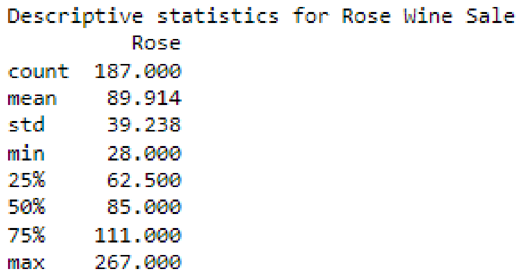
Checking for duplicate values

Sparkling = 11
Rose = 89
Due to time series data set we are not deleting the duplicate values.

Check the basic measures of descriptive statistics of the Time Series



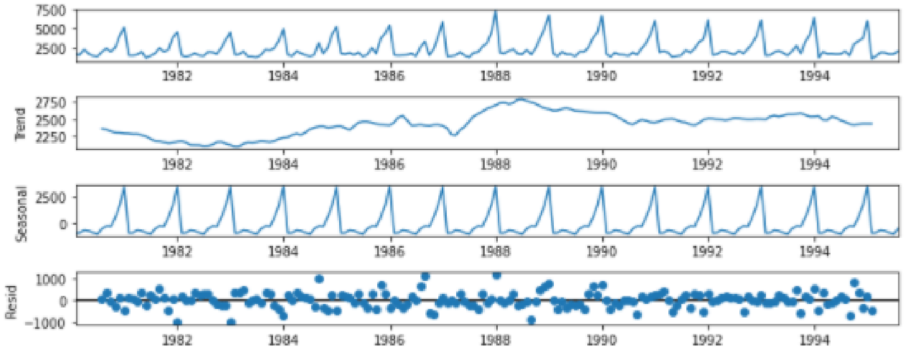
```
Descriptive statistics for Sparkling Wine Sale
       Sparkling
count    187.000
mean    2402.417
std     1295.112
min     1070.000
25%     1605.000
50%     1874.000
75%     2549.000
max     7242.000
```



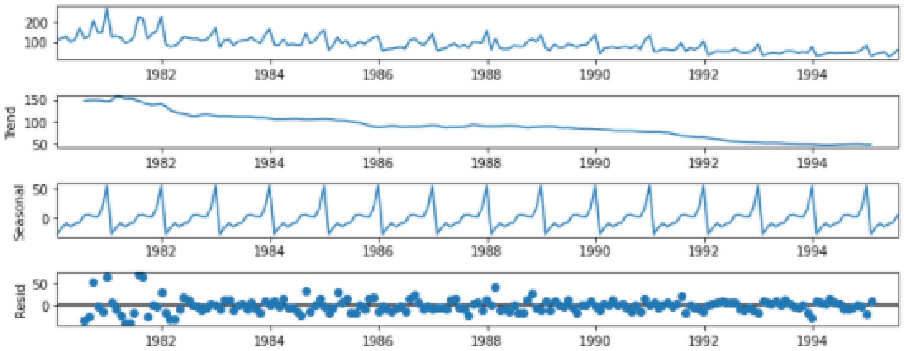
```
Descriptive statistics for Rose Wine Sale
          Rose
count  187.000
mean    89.914
std     39.238
min     28.000
25%     62.500
50%     85.000
75%    111.000
max    267.000
```

Decomposition

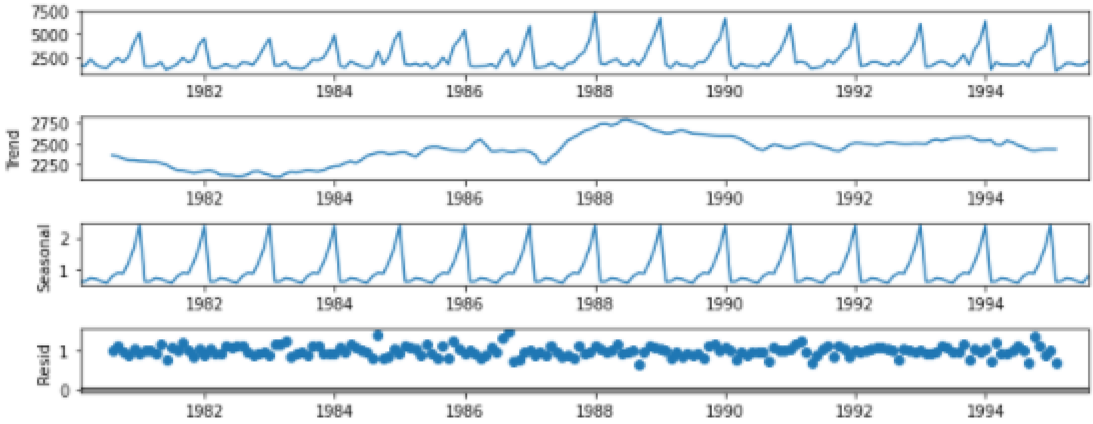
```
Decompose time series of Sparkling Wine Sale with additive model
```



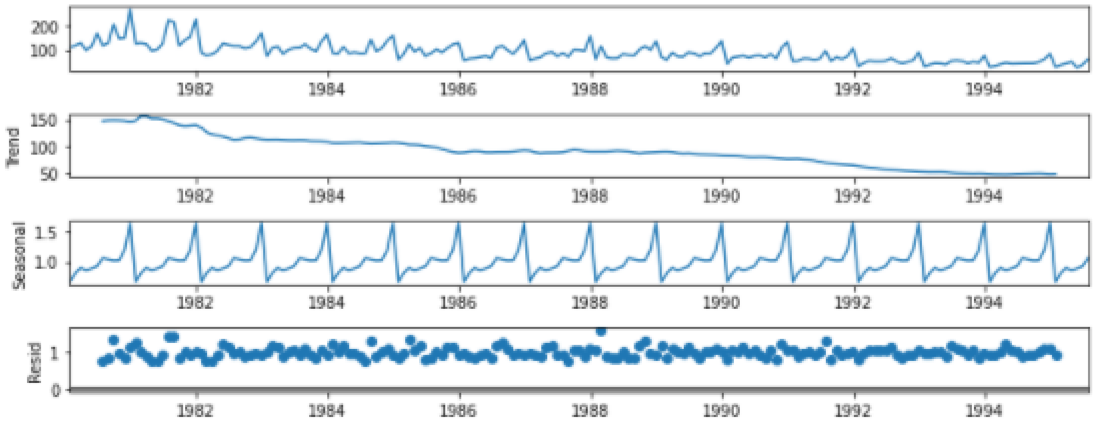
```
Decompose time series of Rose Wine Sale with additive model
```



```
Decompose time series of Sparkling Wine Sale with multiplicative model
```



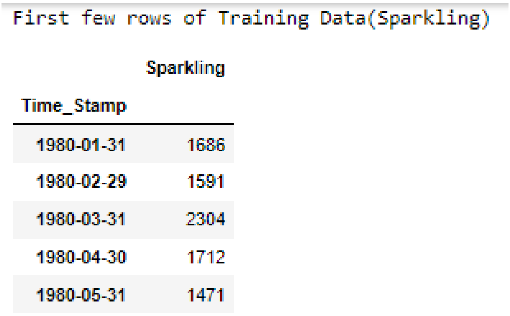
```
Decompose time series of Rose Wine Sale with multiplicative model
```



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

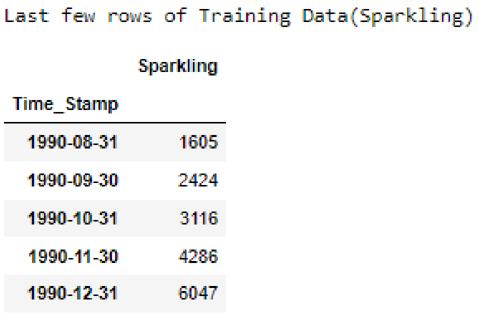
Split the data into train and test and plot the training and test data. (Test data should start 1991)

```
Shape of train and test of Sparkling Wine Sale Data
(132, 1)
(55, 1)


Shape of train and test of Rose Wine Sale Data
(132, 1)
(55, 1)
```



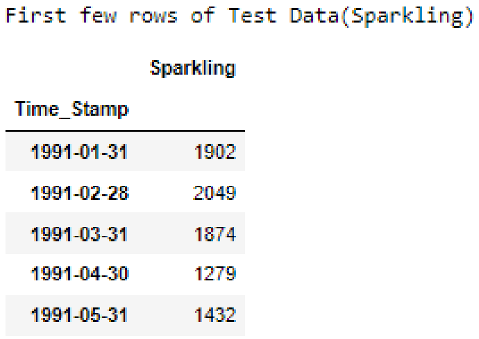
```
First few rows of Training Data(Sparkling)
```

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



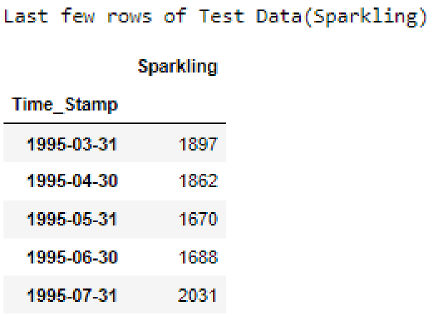
```
Last few rows of Training Data(Sparkling)
```

	Sparkling
Time_Stamp	
1990-08-31	1605
1990-09-30	2424
1990-10-31	3116
1990-11-30	4286
1990-12-31	6047



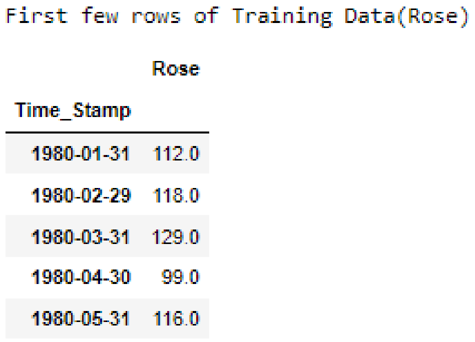
```
First few rows of Test Data(Sparkling)
```

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



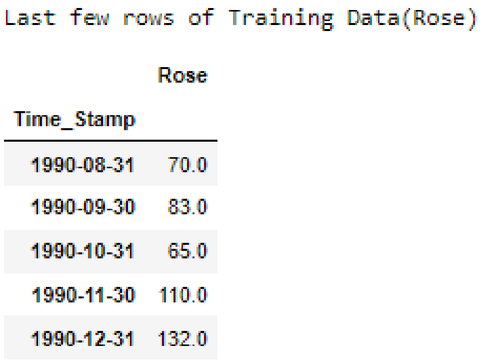
```
Last few rows of Test Data(Sparkling)
```

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



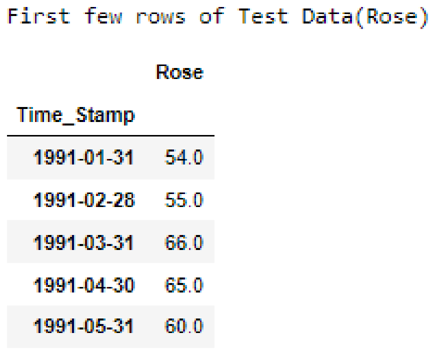
```
First few rows of Training Data(Rose)
```

	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



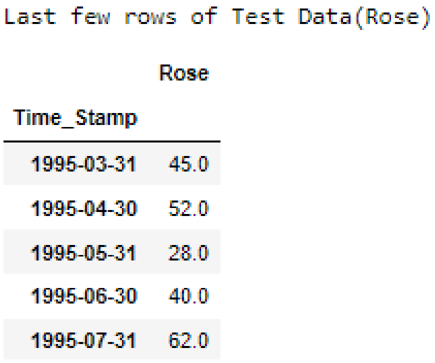
```
Last few rows of Training Data(Rose)
```

	Rose
Time_Stamp	
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



```
First few rows of Test Data(Rose)
```

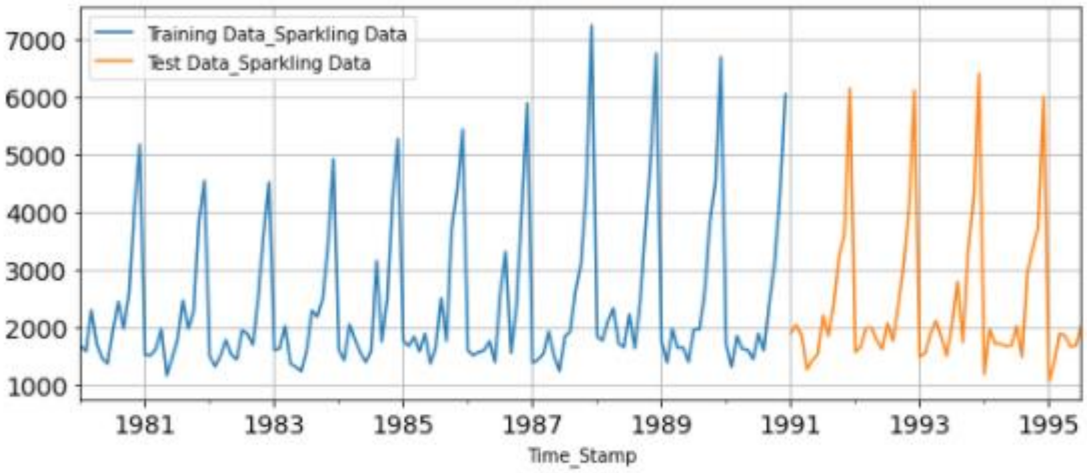
	Rose
Time_Stamp	
1991-01-31	54.0
1991-02-28	55.0
1991-03-31	66.0
1991-04-30	65.0
1991-05-31	60.0



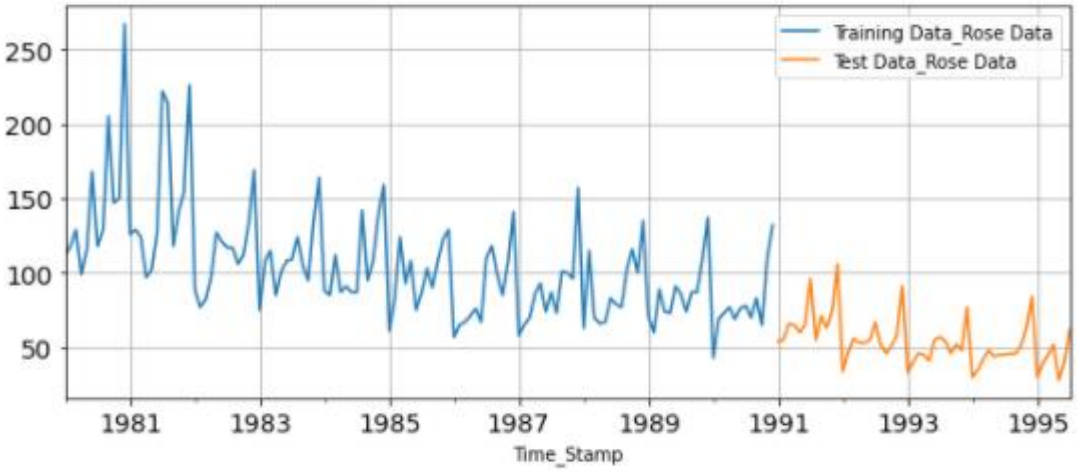
```
Last few rows of Test Data(Rose)
```

	Rose
Time_Stamp	
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 for train and test of Sparkling wine sale data
```

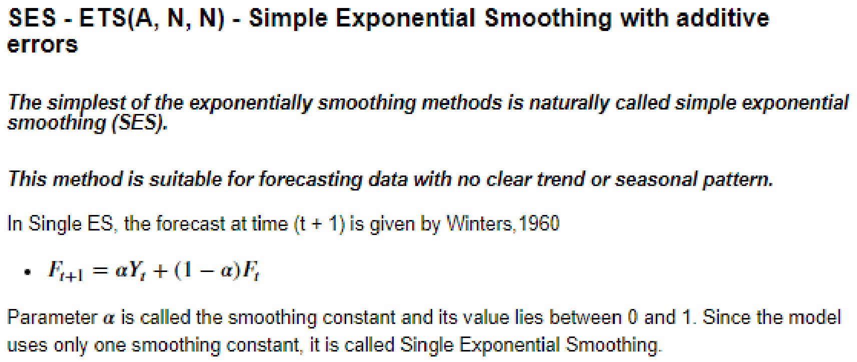


```
Plot for train and test Rose wine sale data
```



5. 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.

Exponential smoothing



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.

In Single ES, the forecast at time (t + 1) is given by Winters,1960

- $F_{t+1} = \alpha Y_t + (1 - \alpha) F_t$

Parameter α 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.

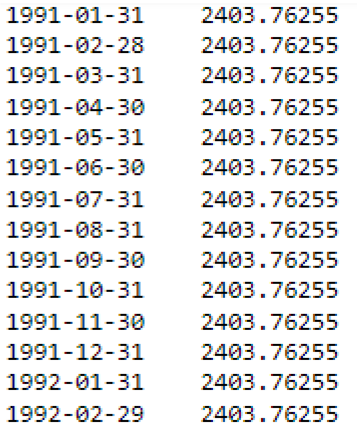
Step1: Creating class for Sparkling wine sale dataset

Step2: Fitting the Simple Exponential Smoothing model for Sparkling wine data and asking python to choose the optimal parameters

Step3: Check the parameters for Sparkling wine data

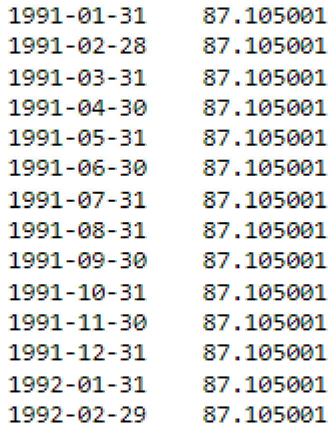
```
Parameters for Sparkling wine sales data
{'smoothing_level': 0.0, 'smoothing_slope': nan, 'smoothing_seasonal': nan, 'damping_slope': nan,
'initial_level': 2403.762550263244, 'initial_slope': nan, 'initial_seasons': array([], dtype=float64),
'use_boxcox': False, 'lamda': None, 'remove_bias': False}

Parameters for Rose wine sales data
{'smoothing_level': 0.09874995867958046, 'smoothing_slope': nan, 'smoothing_seasonal': nan,
'damping_slope': nan, 'initial_level': 134.38699135899094, 'initial_slope': nan, 'initial_seasons':
array([], dtype=float64), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Step4: Using the fitted model on the training set to forecast on the test set(Sparkling Wine sales data)



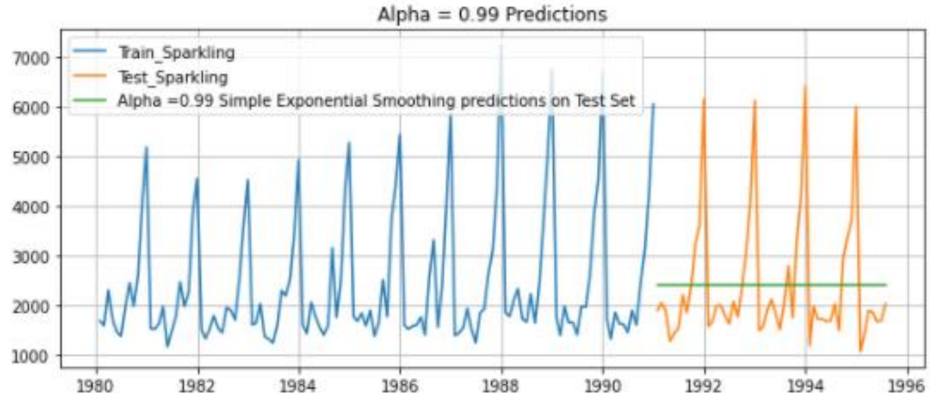
```
1991-01-31    2403.76255
1991-02-28    2403.76255
1991-03-31    2403.76255
1991-04-30    2403.76255
1991-05-31    2403.76255
1991-06-30    2403.76255
1991-07-31    2403.76255
1991-08-31    2403.76255
1991-09-30    2403.76255
1991-10-31    2403.76255
1991-11-30    2403.76255
1991-12-31    2403.76255
1992-01-31    2403.76255
1992-02-29    2403.76255
```

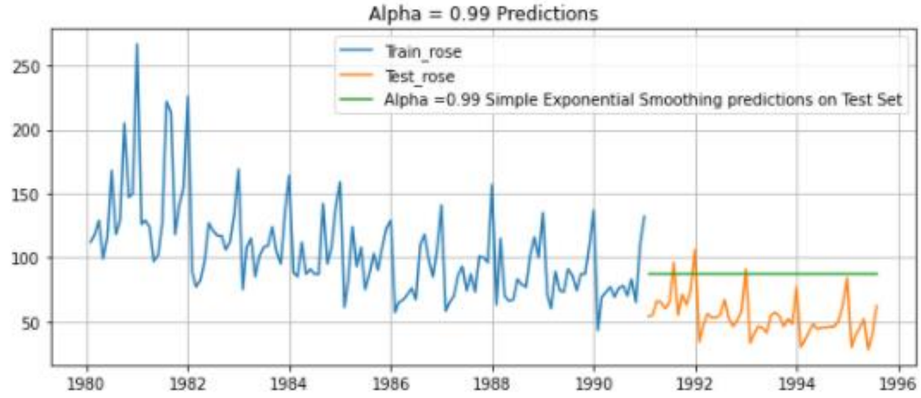
Using the fitted model on the training set to forecast on the test set(Rose Wine sales data)



```
1991-01-31    87.105001
1991-02-28    87.105001
1991-03-31    87.105001
1991-04-30    87.105001
1991-05-31    87.105001
1991-06-30    87.105001
1991-07-31    87.105001
1991-08-31    87.105001
1991-09-30    87.105001
1991-10-31    87.105001
1991-11-30    87.105001
1991-12-31    87.105001
1992-01-31    87.105001
1992-02-29    87.105001
```

Plotting the Training data, Test data and the forecasted values for sparkling and Rose wine sale data



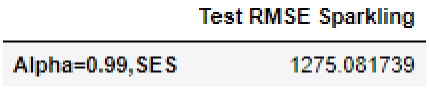


Mean Absolute Percentage Error (MAPE) - Function Definition

```
def MAPE(y_true, y_pred):
    return np.mean((np.abs(y_true-y_pred))/(y_true))*100
```

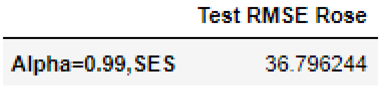
Calculating RMSE value for sparkling and Rose wine sale data using mean_squared_error

Test RMSE Sparkling wine sale data

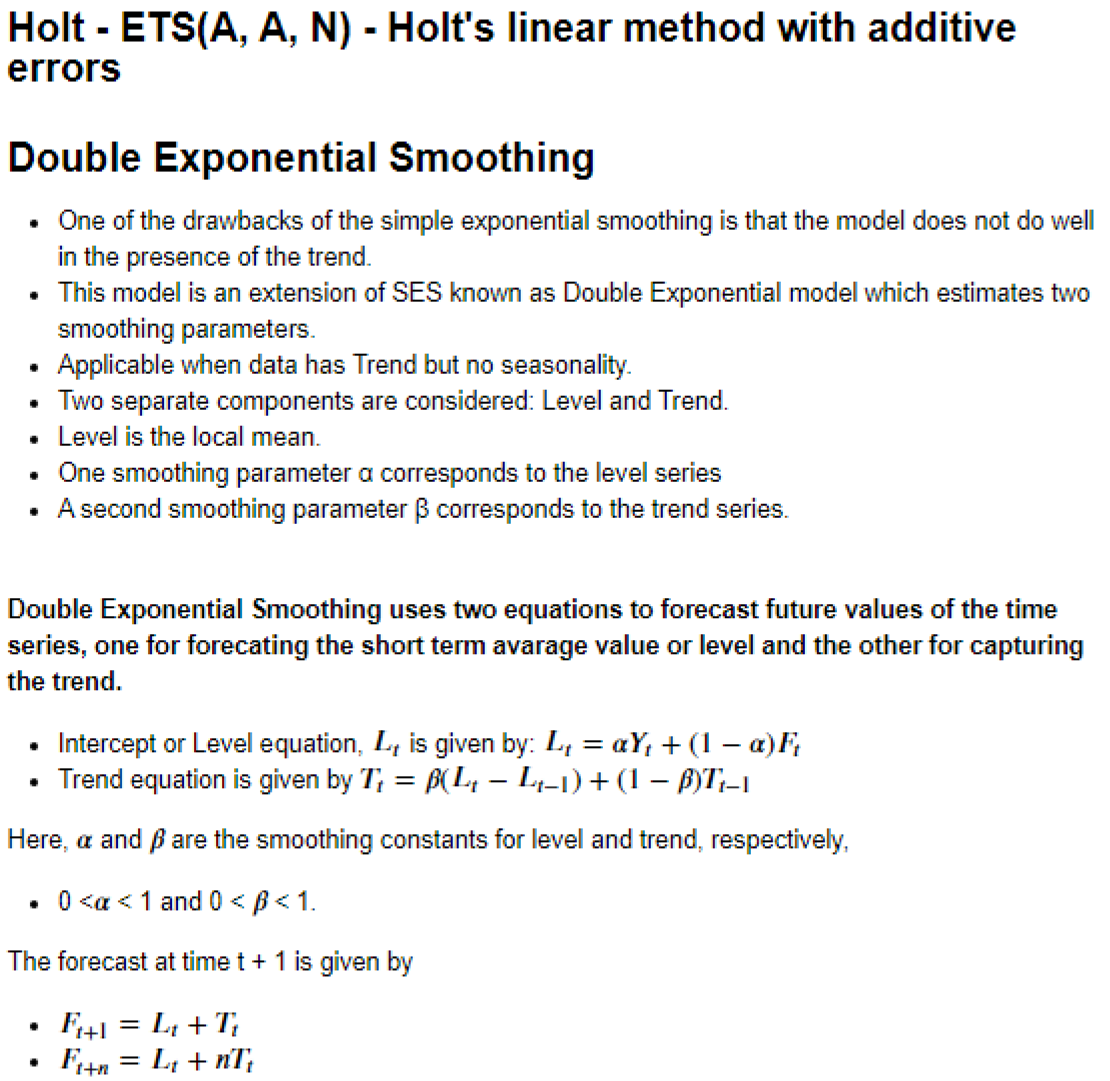


	Test RMSE Sparkling
Alpha=0.99,SES	1275.081739

Test RMSE Rose wine sale data



	Test RMSE Rose
Alpha=0.99,SES	36.796244



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

Double Exponential Smoothing

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

Double Exponential Smoothing uses two equations to forecast future values of the time series, one for forecating the short term avarage value or level and the other for capturing the trend.

- Intercept or Level equation, L_t is given by: $L_t = \alpha Y_t + (1 - \alpha)F_t$
- Trend equation is given by $T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$

Here, α and β are the smoothing constants for level and trend, respectively,

- 0 <α < 1 and 0 < β < 1.

The forecast at time t + 1 is given by

- $F_{t+1} = L_t + T_t$
- $F_{t+n} = L_t + nT_t$

Step1: Initializing the Double Exponential Smoothing Model for sparkling and Rose wine sale data
Step2: Fitting the model
Step3:

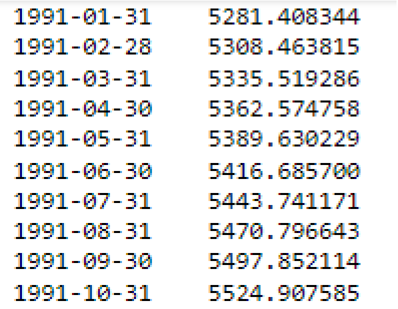
Holt model Exponential Smoothing Estimated Parameters for sparkling wine sale data

```
{'smoothing_level': 0.6477838823329748, 'smoothing_slope': 0.0, 'smoothing_seasonal': nan,
'damping_slope': nan, 'initial_level': 1686.0837646037185, 'initial_slope': 27.05547124370359,
'initial_seasons': array([], dtype=float64), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Holt model Exponential Smoothing Estimated Parameters for rose wine sale data

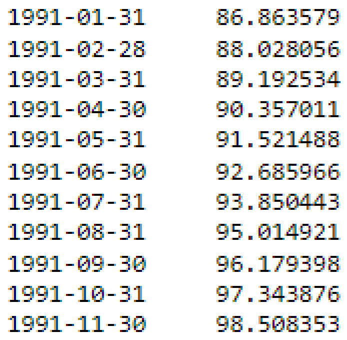
```
{'smoothing_level': 0.15789473684210525, 'smoothing_slope': 0.15789473684210525,
'smoothing_seasonal': nan, 'damping_slope': nan, 'initial_level': 112.0, 'initial_slope': 6.0,
'initial_seasons': array([], dtype=float64), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Forecasting using this model for the duration of the test set or sparkling wine sale data



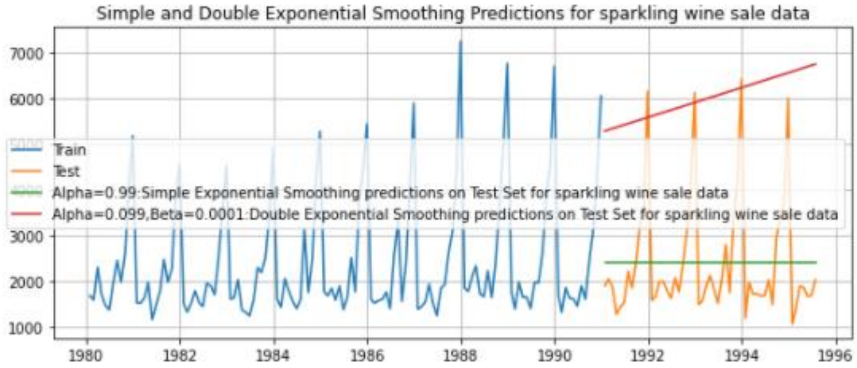
```
1991-01-31    5281.408344
1991-02-28    5308.463815
1991-03-31    5335.519286
1991-04-30    5362.574758
1991-05-31    5389.630229
1991-06-30    5416.685700
1991-07-31    5443.741171
1991-08-31    5470.796643
1991-09-30    5497.852114
1991-10-31    5524.907585
```

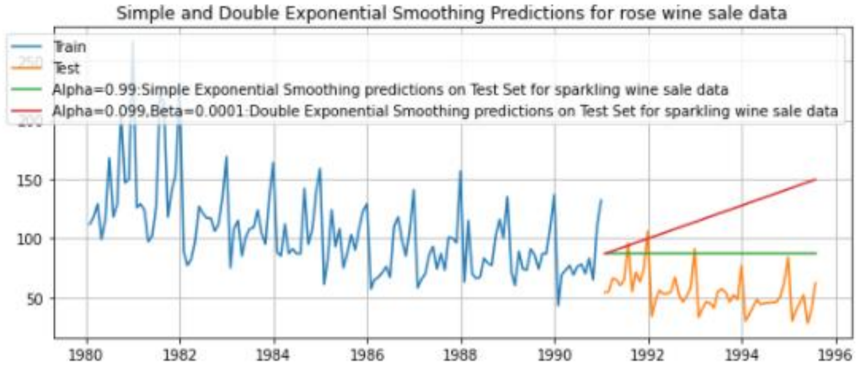
Forecasting using this model for the duration of the test set or Rose wine sale data



```
1991-01-31     86.863579
1991-02-28     88.028056
1991-03-31     89.192534
1991-04-30     90.357011
1991-05-31     91.521488
1991-06-30     92.685966
1991-07-31     93.850443
1991-08-31     95.014921
1991-09-30     96.179398
1991-10-31     97.343876
1991-11-30     98.508353
```

Plotting the Training data, Test data and the forecasted values for sparkling and Rose wine





Calculate RMSE value for sparkling wine sale data by mean_squared_error

Test RMSE Sparkling and Rose wine sale data:
DES RMSE Sparkling: 3850.8478154538407
DES RMSE Rose: 70.57245196981661

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

Step1: Initializing the Double Exponential Smoothing Model for sparkling and Rose wine sale data

Step2: Fitting the model

Step3:

Holt Winters model Exponential Smoothing Estimated Parameters for sparkling wine sale data

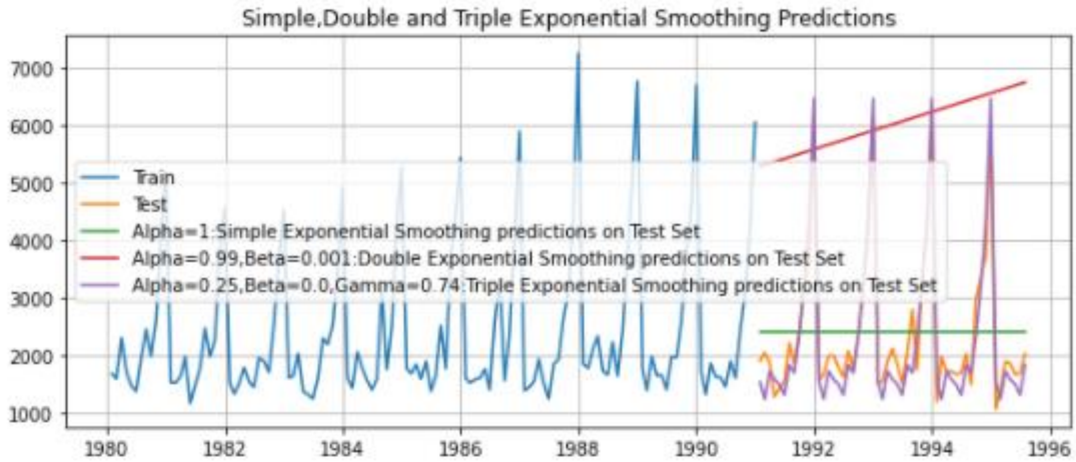
```
{'smoothing_level': 0.08621043130197674, 'smoothing_slope':
4.534154798535049e-09, 'smoothing_seasonal': 0.47637161151204716,
'damping_slope': nan, 'initial_level': 1684.9037953795528, 'initial_slope':
0.003939090229177399, 'initial_seasons': array([  39.17594081,  -37.22346237,
464.45609233,  206.27645763,
       -140.60405271, -156.56958591,  338.10668461,  856.91481885,
        403.65380447,  971.29856276, 2401.54344377, 3426.51448586]),
'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

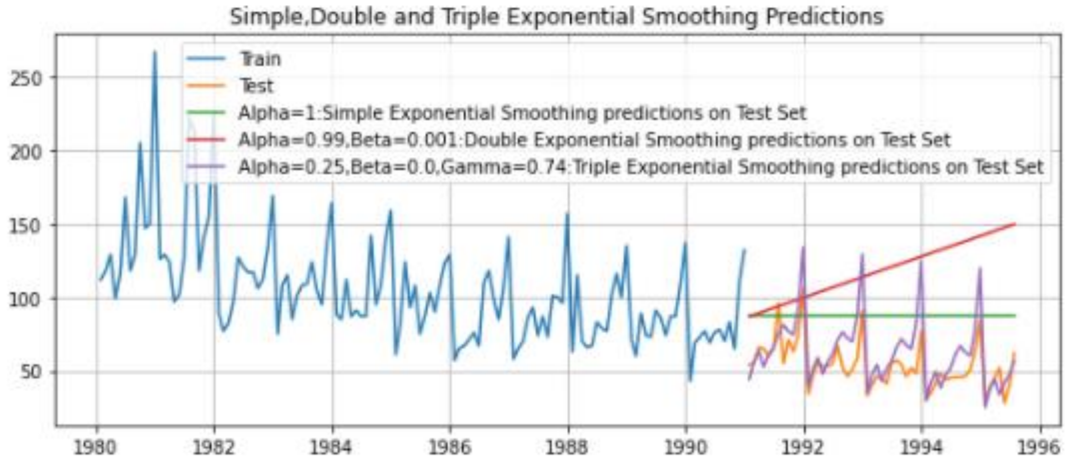
Holt Winters model Exponential Smoothing Estimated Parameters for rose wine sale data

```
{'smoothing_level': 0.13346905584155852, 'smoothing_slope':
0.013798044930131528, 'smoothing_seasonal': 0.0, 'damping_slope': nan,
'initial_level': 77.90998273991845, 'initial_slope': 0.0, 'initial_seasons': array([
37.19347871,  49.53447903,  57.45342246,  46.82461047,
        55.5675085 ,  60.9978818 ,  70.94829431,  76.95581437,
        72.98548228,  71.11492918,  89.18261025, 131.38117683]), 'use_boxcox':
False, 'lamda': None, 'remove_bias': False}
```

Step4: Forecasting using this model for the duration of the test set(Sparkling and Rose Wine)

Step5: Plotting the Training data, Test data and the forecasted values for sparkling and rose wine data set





Calculating RMSE value for sparkling and Rose wine sale data by mean_squared_error

TES RMSE Sparkling:	362.7541597031013
TES RMSE Rose:	16.443203233657176

Linear Regression

Step1:
For this particular linear regression, we are going to regress the 'Sparkling' and 'Rose' variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

Training Time instance for sparkling data
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance for sparkling data
[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, 183, 184, 185, 186, 187]

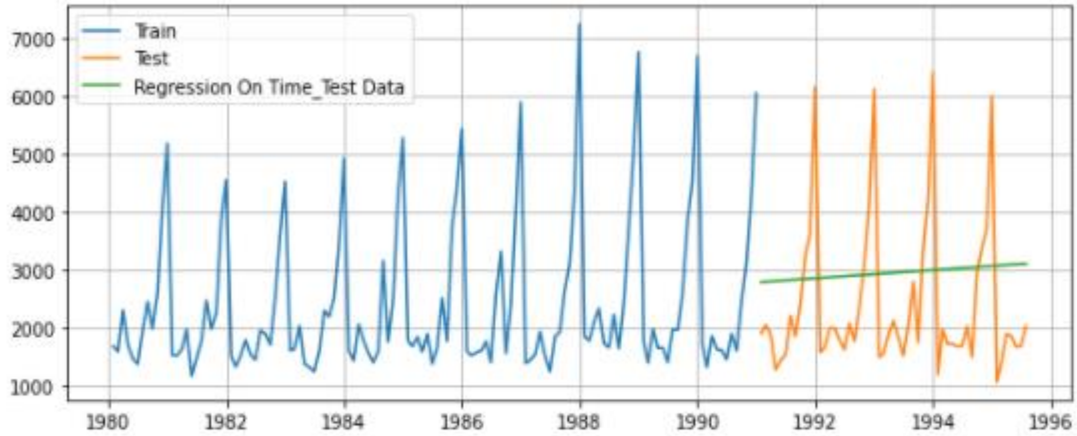
Training Time instance for rose data
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance for rose data
[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, 183, 184, 185, 186, 187]

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

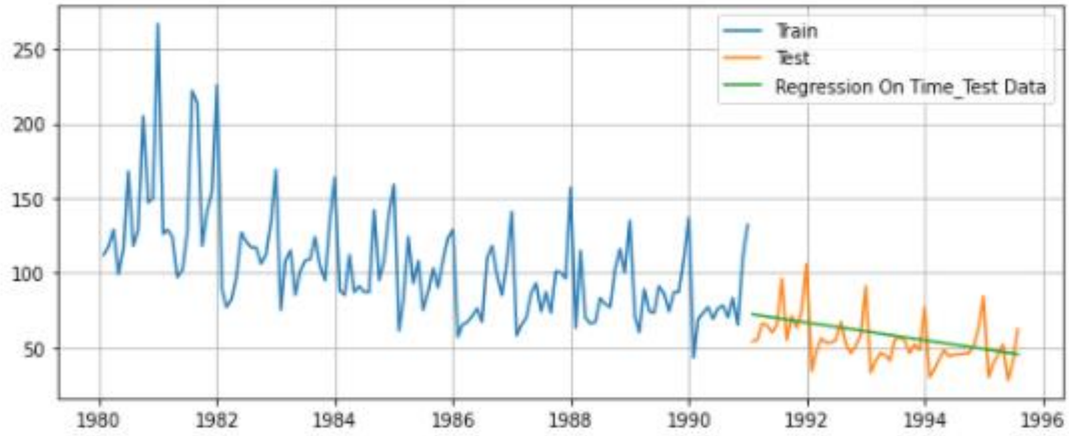
Step2: Fitting Time Instant data to Sparkling and RoseDataset

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

Plotting Regression on time test data for Sparkling wine sale data



Plotting Regression on time test data for Rose wine sale data



Model Evaluation:
RMSE for Sparkling Wine Sale Data using mean_squared_error

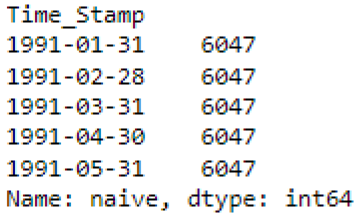
For RegressionOnTime forecast on the Test Data of Sparkling Wine Sale Data , RMSE is 1389.135

For RegressionOnTime forecast on the Test Data of Rose Wine Sale Data , RMSE is 15.269

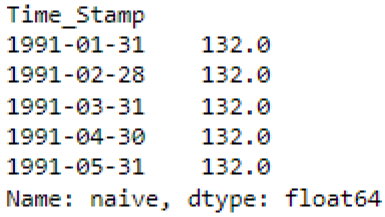
Naive Approach: $\hat{y}_{t+1} = y_t$

For this particular naive model, we say that 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.

Step1: np.asarray for both Dataset

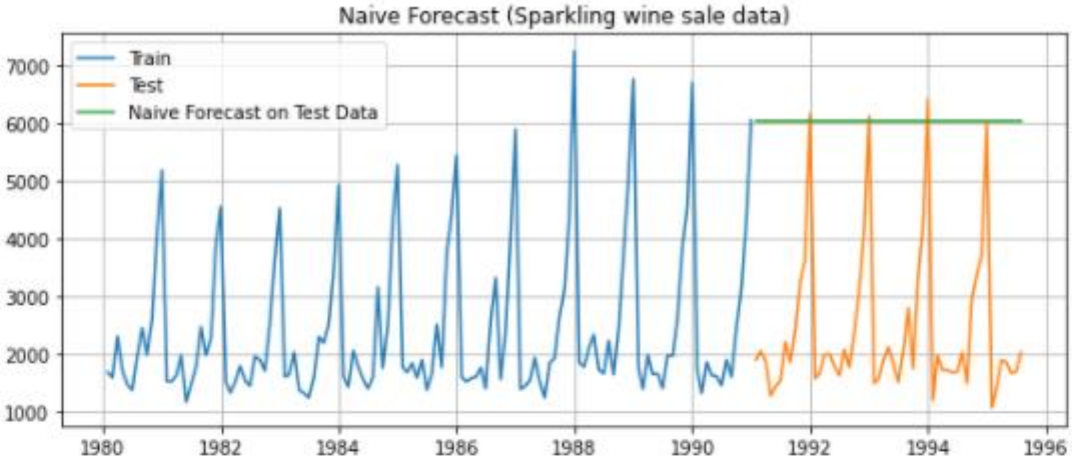


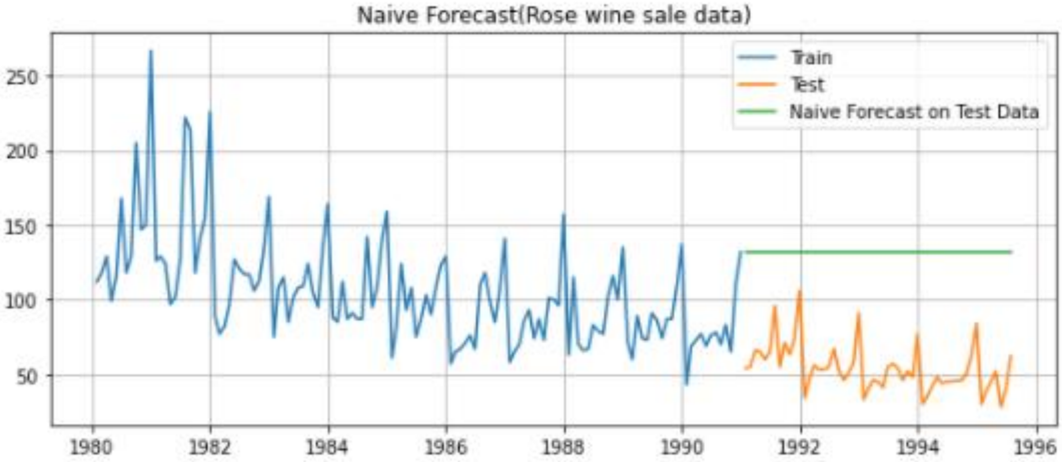
```
Time_Stamp
1991-01-31    6047
1991-02-28    6047
1991-03-31    6047
1991-04-30    6047
1991-05-31    6047
Name: naive, dtype: int64
```



```
Time_Stamp
1991-01-31    132.0
1991-02-28    132.0
1991-03-31    132.0
1991-04-30    132.0
1991-05-31    132.0
Name: naive, dtype: float64
```

Plotting Naives on data for Sparkling and Rose wine sale data





Model Evaluation:

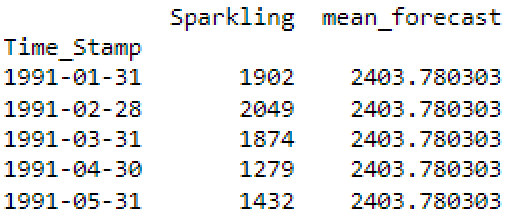
RMSE for Sparkling and Rose Wine Sale Data using mean_squared_error

For Naives forecast on the Test Data for Sparkling wine sale data, RMSE is 3864.279
For Naives forecast on the Test Data for Rose wine sale data, RMSE is 3864.279

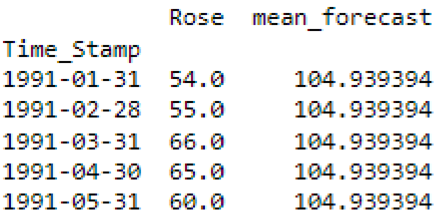
Simple Average

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

Step1: Applying Mean value for Sparkling wine sale data

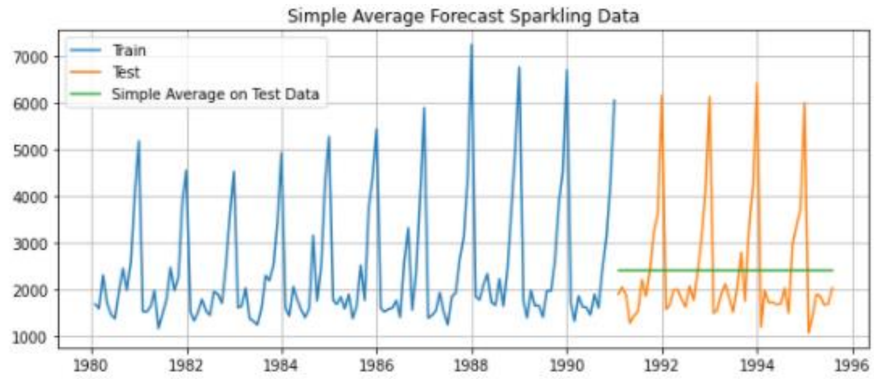


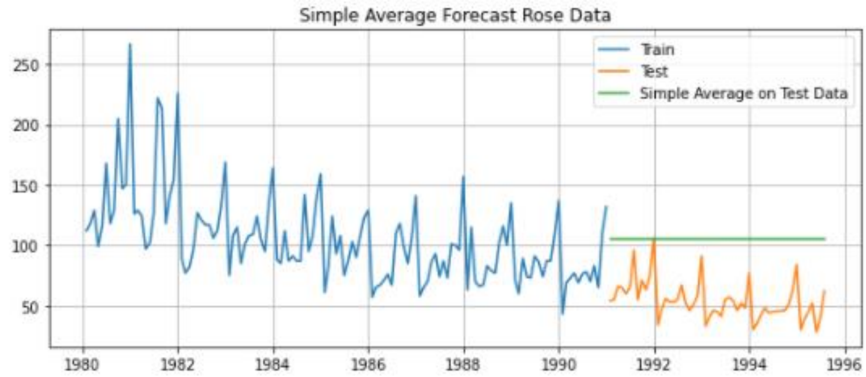
```
            Sparkling  mean_forecast
Time_Stamp
1991-01-31       1902    2403.780303
1991-02-28       2049    2403.780303
1991-03-31       1874    2403.780303
1991-04-30       1279    2403.780303
1991-05-31       1432    2403.780303
```



```
             Rose  mean_forecast
Time_Stamp
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
```

Plotting Average on time test data for Sparkling and Rose wine sale data

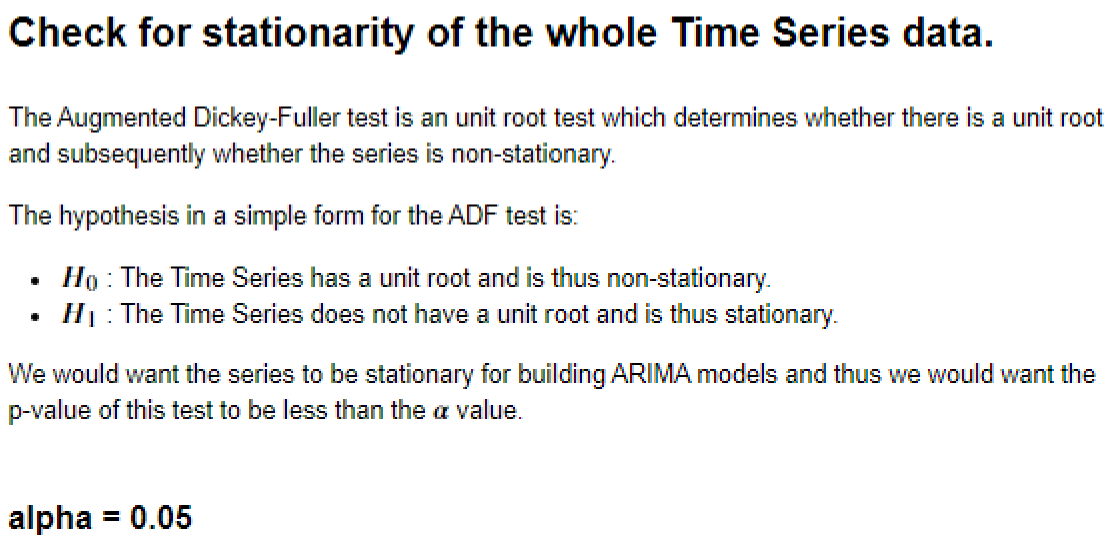




Model Evaluation:
RMSE for Sparkling and Rose Wine Sale Data using mean_squared_error

For Simple Average forecast on the Test Data of Sparkling Wine Sale Data,
RMSE is 1275.082
For Simple Average forecast on the Test Data of rose Wine Sale Data,
RMSE is 53.461

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.



Check for stationarity of the whole Time Series data.

The Augmented Dickey-Fuller test is an unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

The hypothesis in a simple form for the ADF test is:

- H_0 : The Time Series has a unit root and is thus non-stationary.
- H_1 : The Time Series does not have a unit root and is thus stationary.

We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the α value.

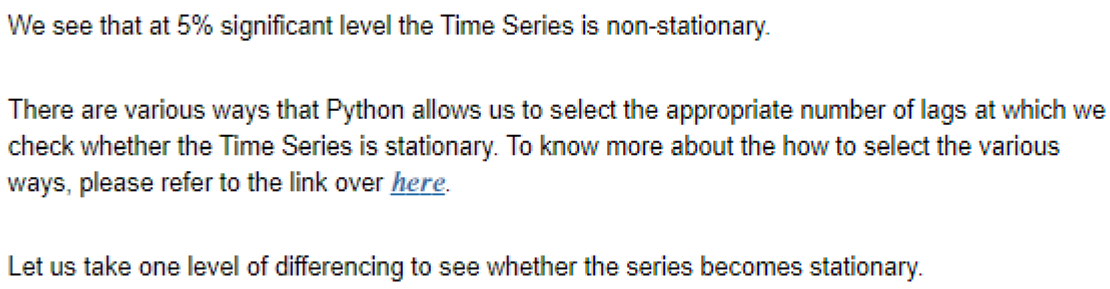
alpha = 0.05

Augmented Dickey Fuller Test (ADF Test) for Sparkling Wine sale data

Augmented Dickey Fuller Test (ADF Test) for Sparkling Wine sale data
DF test statistic is -1.798
DF test p-value is 0.7055958459932714
Number of lags used 12

Augmented Dickey Fuller Test (ADF Test) for Rose Wine sale data

Augmented Dickey Fuller Test (ADF Test) for Rose Wine sale data
DF test statistic is -2.240
DF test p-value is 0.46713716277930917
Number of lags used 13



We see that at 5% significant level the Time Series is non-stationary.

There are various ways that Python allows us to select the appropriate number of lags at which we check whether the Time Series is stationary. To know more about the how to select the various ways, please refer to the link over *here*.

Let us take one level of differencing to see whether the series becomes stationary.

One level of differencing for Sparkling Wine sale data

DF test statistic is -44.912
DF test p-value is 0.000
Number of lags used 10

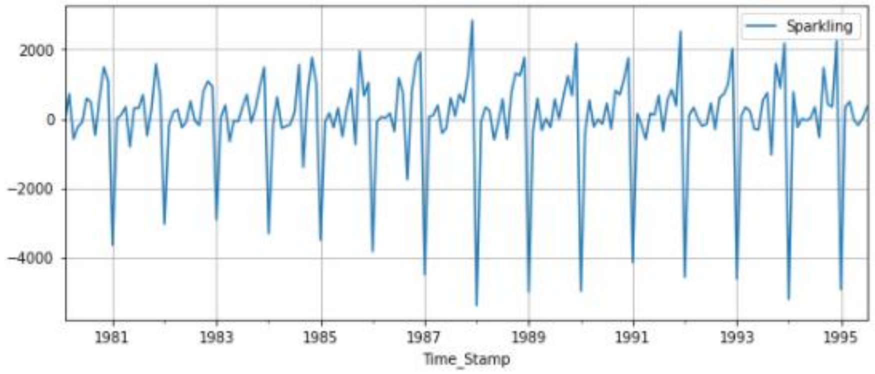
One level of differencing for Rose Wine sale data

One level of differencing for Rose Wine sale data
DF test statistic is -8.162
DF test p-value is 0.000
Number of lags used 12

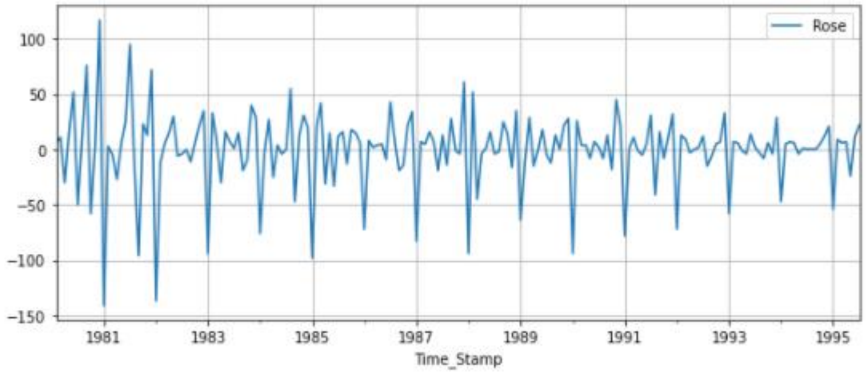
Now, let us go ahead and plot the stationary series.

Plot for One level of differencing for Sparkling and Rose wine sale data

```
Plot for one level of differencingfor Sparkling Wine sale data
```



```
Plot for one level of differencing for Rose Wine sale data
```



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.

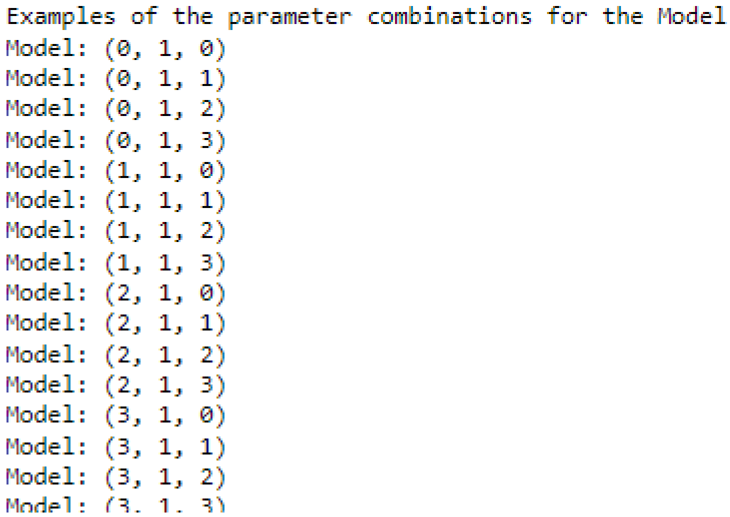
Build an Automated version of a SARIMA model for which the best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC).

Step1:

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

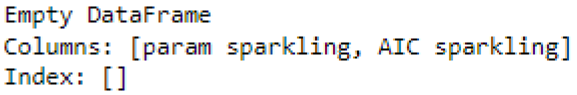
The loop helps us in getting a combination of different parameters of p and q in the range of 0 and 2

We have kept the value of d as 1 as we need to take a difference of the series to make it stationary.

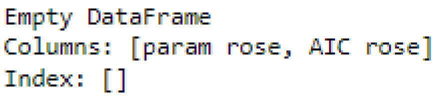


```
Examples of the parameter combinations for the Model
Model: (0, 1, 0)
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (0, 1, 3)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (1, 1, 3)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
Model: (2, 1, 3)
Model: (3, 1, 0)
Model: (3, 1, 1)
Model: (3, 1, 2)
Model: (3, 1, 3)
```

Step2: Creating an empty Data frame with column names only for sparkling data



```
Empty DataFrame
Columns: [param sparkling, AIC sparkling]
Index: []
```



```
Empty DataFrame
Columns: [param rose, AIC rose]
Index: []
```

Step3:
Running a loop within the pdq parameters defined by itertools.
Using the parameters from the loop.

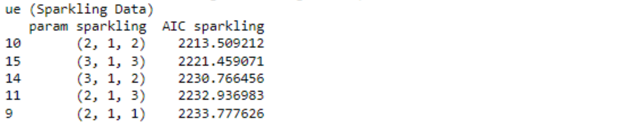
Appending the AIC values and the model parameters to the previously created data frame Sparkling.

```
ARIMA(0, 1, 0) - AIC sparkling:2267.6630357855465
ARIMA(0, 1, 1) - AIC sparkling:2263.060015591812
ARIMA(0, 1, 2) - AIC sparkling:2234.408323132853
ARIMA(0, 1, 3) - AIC sparkling:2233.9948577616415
ARIMA(1, 1, 0) - AIC sparkling:2266.6085393190097
  warn('Non-invertible starting MA parameters found.'
ARIMA(1, 1, 1) - AIC sparkling:2235.755094660315
ARIMA(1, 1, 2) - AIC sparkling:2234.5272004515837
ARIMA(1, 1, 3) - AIC sparkling:2235.607807333272
ARIMA(2, 1, 0) - AIC sparkling:2260.365743968097
ARIMA(2, 1, 1) - AIC sparkling:2233.777626229804
ARIMA(2, 1, 2) - AIC sparkling:2213.509212349606
ARIMA(2, 1, 3) - AIC sparkling:2232.9369831015447
ARIMA(3, 1, 0) - AIC sparkling:2257.72337899794
ARIMA(3, 1, 1) - AIC sparkling:2235.4985697021007
ARIMA(3, 1, 2) - AIC sparkling:2230.7664562362497
ARIMA(3, 1, 3) - AIC sparkling:2221.459071080334
```

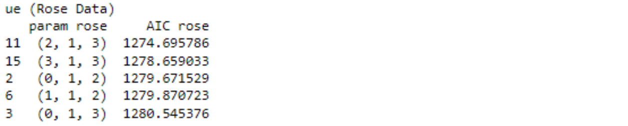
Appending the AIC values and the model parameters to the previously created data frame Rose.

```
ARIMA(0, 1, 0) - AIC rose:1333.1546729124348
ARIMA(0, 1, 1) - AIC rose:1282.3098319748274
ARIMA(0, 1, 2) - AIC rose:1279.6715288535775
ARIMA(0, 1, 3) - AIC rose:1280.5453761734652
ARIMA(1, 1, 0) - AIC rose:1317.3503105381546
ARIMA(1, 1, 1) - AIC rose:1280.5742295380087
ARIMA(1, 1, 2) - AIC rose:1279.870723423192
ARIMA(1, 1, 3) - AIC rose:1281.8707223309975
ARIMA(2, 1, 0) - AIC rose:1298.6110341604958
ARIMA(2, 1, 1) - AIC rose:1281.507862186847
ARIMA(2, 1, 2) - AIC rose:1281.8707222264254
```

Step3: Sort the above AIC values in the ascending order to get the parameters for the minimum AIC value



```
AIC values in the ascending order to get the parameters for the minimum AIC val
ue (Sparkling Data)
   param sparkling  AIC sparkling
10       (2, 1, 2)    2213.509212
15       (3, 1, 3)    2221.459071
14       (3, 1, 2)    2230.766456
11       (2, 1, 3)    2232.936983
9        (2, 1, 1)    2233.777626
```



```
AIC values in the ascending order to get the parameters for the minimum AIC val
ue (Rose Data)
   param rose     AIC rose
11  (2, 1, 3)  1274.695786
15  (3, 1, 3)  1278.659033
2   (0, 1, 2)  1279.671529
6   (1, 1, 2)  1279.870723
3   (0, 1, 3)  1280.545376
```

Step4:

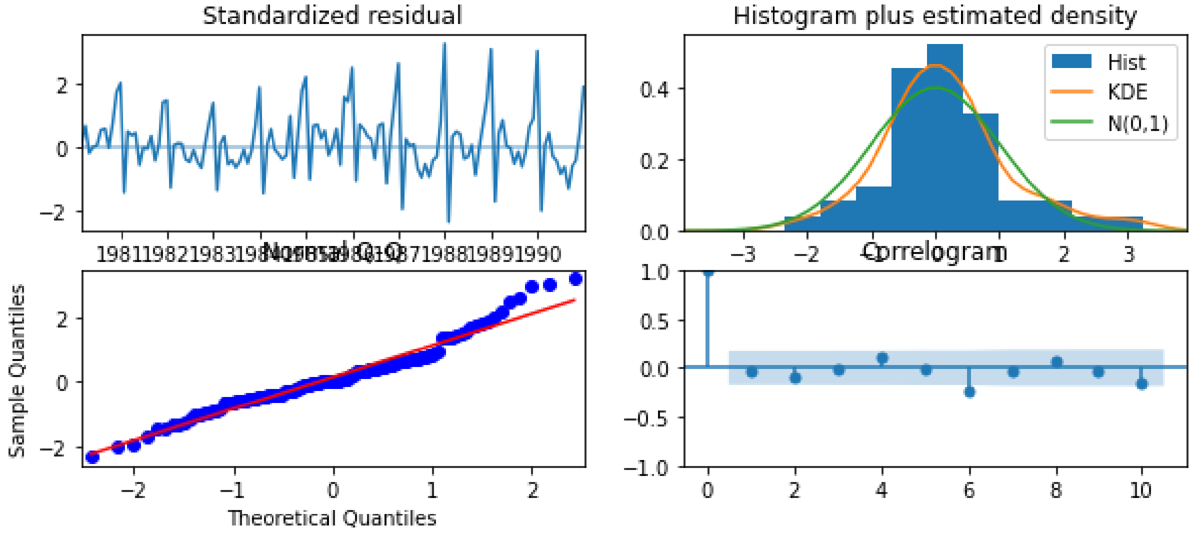
Fitting minimum AIC value for Sparkling wine sale data

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Sparkling   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1101.755
Date:                Sun, 20 Mar 2022   AIC                           2213.509
Time:                        20:03:23   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     28.781      0.000       1.223       1.401
ar.L2         -0.5593      0.072     -7.741      0.000      -0.701      -0.418
ma.L1         -1.9917      0.109    -18.217      0.000      -2.206      -1.777
ma.L2          0.9999      0.110      9.109      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 (Q):                      293.72   Jarque-Bera (JB):                1
4.46
Prob(Q):                              0.00   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 1.84e
+28. Standard errors may be unstable.
```

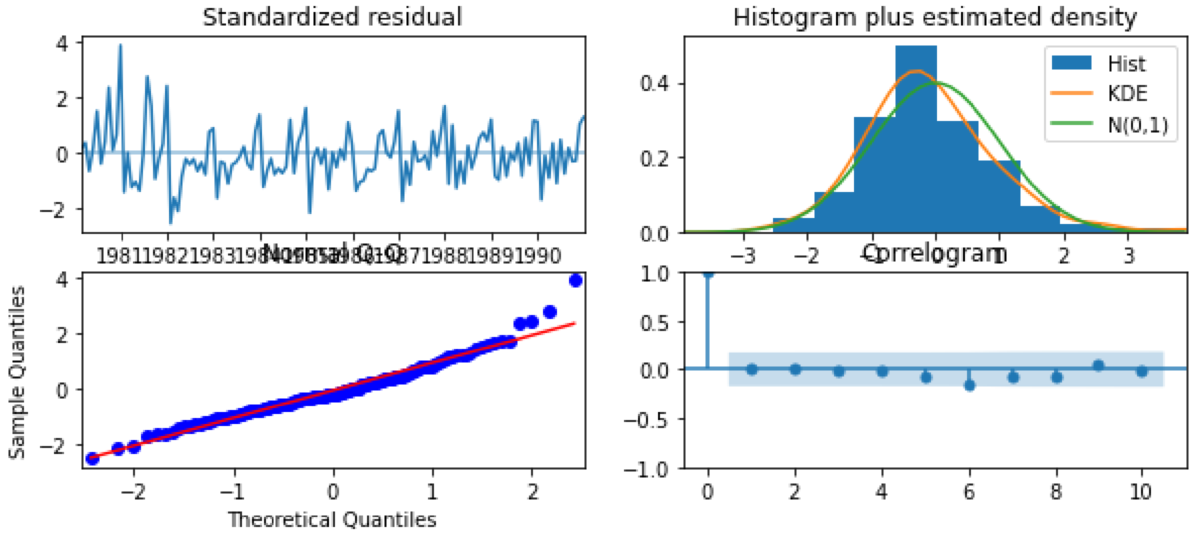
Fitting minimum AIC value for Rose wine sale data

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   Rose   No. Observations:                  132
Model:                 ARIMA(2, 1, 3)   Log Likelihood                -631.348
Date:                Sun, 20 Mar 2022   AIC                           1274.696
Time:                        20:03:23   BIC                           1291.947
Sample:                    01-31-1980   HQIC                          1281.706
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.6779      0.084    -20.019      0.000      -1.842      -1.514
ar.L2         -0.7287      0.084     -8.691      0.000      -0.893      -0.564
ma.L1          1.0445      0.570      1.833      0.067      -0.072       2.161
ma.L2         -0.7717      0.128     -6.009      0.000      -1.023      -0.520
ma.L3         -0.9042      0.515     -1.755      0.079      -1.914       0.106
sigma2       859.4992    477.796      1.799      0.072     -76.964    1795.962
===============================================================================
====
Ljung-Box (Q):                      101.08   Jarque-Bera (JB):                2
4.43
Prob(Q):                              0.00   Prob(JB):
0.00
Heteroskedasticity (H):               0.40   Skew:
0.71
Prob(H) (two-sided):                  0.00   Kurtosis:
4.57
===============================================================================
====

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

Diagnostics plot for Sparkling Wine sale data



Diagnostics plot for Rose Wine sale data

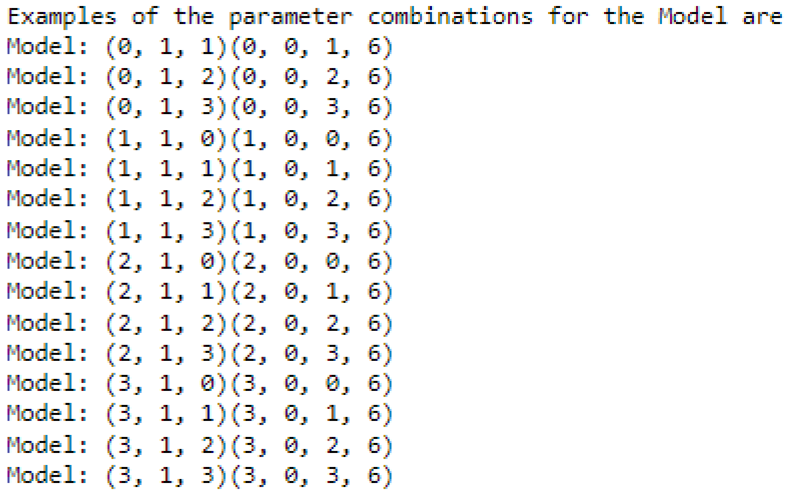


Predict on the Test Set using this model and evaluate the model.

RMSE of Sparkling Wine Sale Data: 1299.9795334022403
MAPE of Sparkling Wine Sale Data: 47.099972867001924

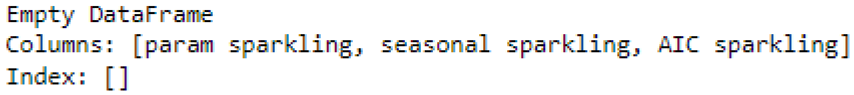
RMSE of Rose Wine Sale Data: 36.8159446995155
MAPE of Rose Wine Sale Data: 75.84498541161119

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

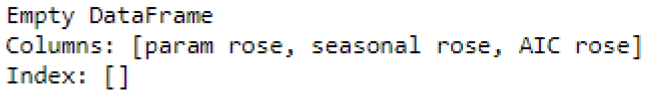


```
Examples of the parameter combinations for the Model are
Model: (0, 1, 1)(0, 0, 1, 6)
Model: (0, 1, 2)(0, 0, 2, 6)
Model: (0, 1, 3)(0, 0, 3, 6)
Model: (1, 1, 0)(1, 0, 0, 6)
Model: (1, 1, 1)(1, 0, 1, 6)
Model: (1, 1, 2)(1, 0, 2, 6)
Model: (1, 1, 3)(1, 0, 3, 6)
Model: (2, 1, 0)(2, 0, 0, 6)
Model: (2, 1, 1)(2, 0, 1, 6)
Model: (2, 1, 2)(2, 0, 2, 6)
Model: (2, 1, 3)(2, 0, 3, 6)
Model: (3, 1, 0)(3, 0, 0, 6)
Model: (3, 1, 1)(3, 0, 1, 6)
Model: (3, 1, 2)(3, 0, 2, 6)
Model: (3, 1, 3)(3, 0, 3, 6)
```

Creating Empty Data frame



```
Empty DataFrame
Columns: [param sparkling, seasonal sparkling, AIC sparkling]
Index: []
```



```
Empty DataFrame
Columns: [param rose, seasonal rose, AIC rose]
Index: []
```

Appending the AIC values and the model parameters to the previously created data frame Sparkling.

SARIMA(0, 1, 0)x(0, 0, 0, 6) - AIC sparkling:2251.3597196862966

SARIMA(0, 1, 0)x(0, 0, 1, 6) - AIC sparkling:2152.3780761716307

SARIMA(0, 1, 0)x(0, 0, 2, 6) - AIC sparkling:1955.6355536892759

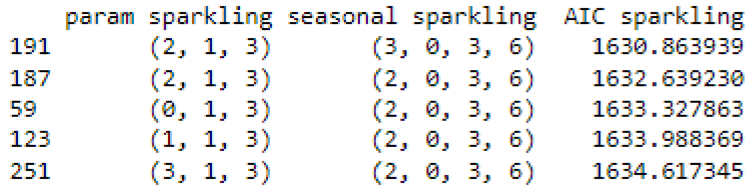
SARIMA(0, 1, 0)x(0, 0, 3, 6) - AIC sparkling:1863.784515497349

SARIMA(0, 1, 0)x(1, 0, 0, 6) - AIC sparkling:2164.4097581959904

Appending the AIC values and the model parameters to the previously created data frame Rose.

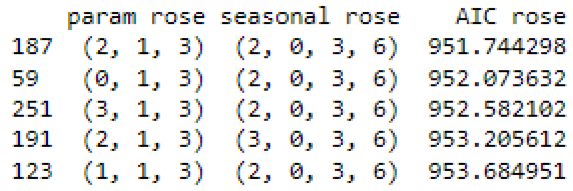
SARIMA(0, 1, 0)x(0, 0, 0, 6) - AIC rose:1323.9657875279158
SARIMA(0, 1, 0)x(0, 0, 1, 6) - AIC rose:1264.4996261113863
SARIMA(0, 1, 0)x(0, 0, 2, 6) - AIC rose:1144.7077471827183
SARIMA(0, 1, 0)x(0, 0, 3, 6) - AIC rose:1081.271383062523
SARIMA(0, 1, 0)x(1, 0, 0, 6) - AIC rose:1274.7897737087983
SARIMA(0, 1, 0)x(1, 0, 1, 6) - AIC rose:1241.787094514905
SARIMA(0, 1, 0)x(1, 0, 2, 6) - AIC rose:1146.3093266722124
SARIMA(0, 1, 0)x(1, 0, 3, 6) - AIC rose:1058.9861743124393
SARIMA(0, 1, 0)x(2, 0, 0, 6) - AIC rose:1137.9167236212038

Sorting SARIMA Model AIC value for Sparkling Wine Sale data



```
     param sparkling seasonal sparkling  AIC sparkling
191        (2, 1, 3)       (3, 0, 3, 6)    1630.863939
187        (2, 1, 3)       (2, 0, 3, 6)    1632.639230
59         (0, 1, 3)       (2, 0, 3, 6)    1633.327863
123        (1, 1, 3)       (2, 0, 3, 6)    1633.988369
251        (3, 1, 3)       (2, 0, 3, 6)    1634.617345
```

Sorting SARIMA Model AIC value for Rose Wine Sale data



```
     param rose seasonal rose    AIC rose
187  (2, 1, 3)  (2, 0, 3, 6)  951.744298
59   (0, 1, 3)  (2, 0, 3, 6)  952.073632
251  (3, 1, 3)  (2, 0, 3, 6)  952.582102
191  (2, 1, 3)  (3, 0, 3, 6)  953.205612
123  (1, 1, 3)  (2, 0, 3, 6)  953.684951
```

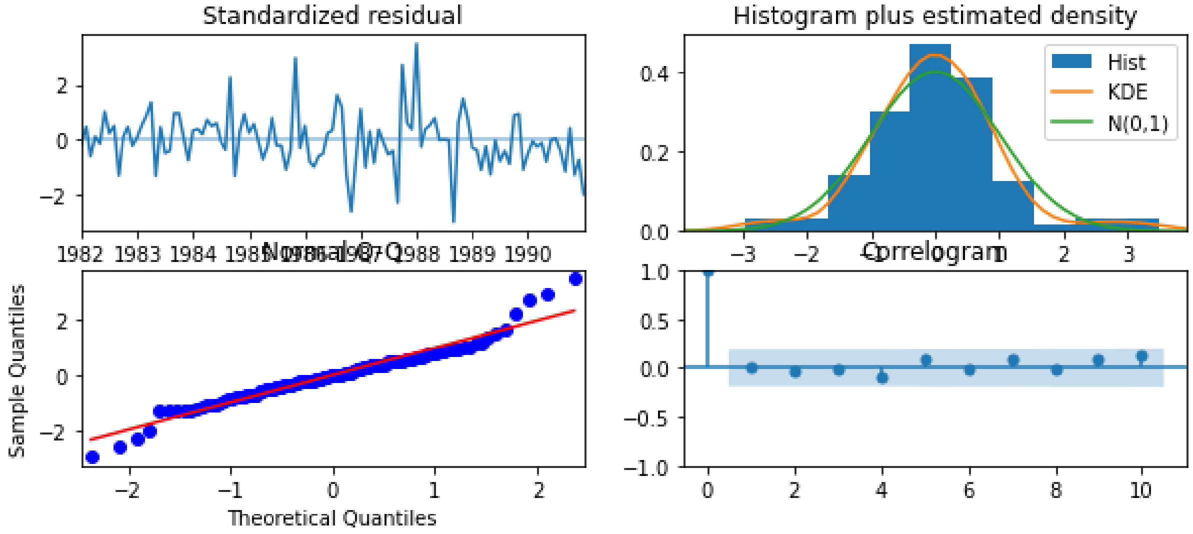
Fitting best value for Sparkling value

```
                                     SARIMAX Results                                      
==========================================================================================
Dep. Variable:                            Sparkling   No. Observations:                132
Model:             SARIMAX(3, 1, 3)x(2, 0, 3, 6)   Log Likelihood                -805.309
Date:                              Sun, 20 Mar 2022   AIC                           1634.617
Time:                                      20:20:07   BIC                           1666.914
Sample:                                  01-31-1980   HQIC                          1647.715
                                       - 12-31-1990                                         
Covariance Type:                                opg                                         
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.1695      0.154     -7.615      0.000      -1.471      -0.869
ar.L2         -0.9186      0.163     -5.631      0.000      -1.238      -0.599
ar.L3          0.0197      0.143      0.137      0.891      -0.261       0.301
ma.L1          0.4248      0.172      2.465      0.014       0.087       0.763
ma.L2         -0.1132      0.129     -0.875      0.381      -0.367       0.140
ma.L3         -0.8639      0.110     -7.885      0.000      -1.079      -0.649
ar.S.L6       -0.0015      0.030     -0.051      0.959      -0.059       0.056
ar.S.L12       1.0451      0.022     47.830      0.000       1.002       1.088
ma.S.L6       -0.0478      0.168     -0.285      0.776      -0.376       0.281
ma.S.L12      -0.6461      0.107     -6.051      0.000      -0.855      -0.437
ma.S.L18       0.1289      0.184      0.699      0.484      -0.233       0.490
sigma2      1.399e+05   1.32e-06   1.06e+11      0.000     1.4e+05     1.4e+05
===================================================================================
Ljung-Box (Q):                       23.94   Jarque-Bera (JB):                22.00
Prob(Q):                              0.98   Prob(JB):                         0.00
Heteroskedasticity (H):               1.40   Skew:                             0.35
Prob(H) (two-sided):                  0.32   Kurtosis:                         5.09
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.71e+27. Standard
errors may be unstable.
```

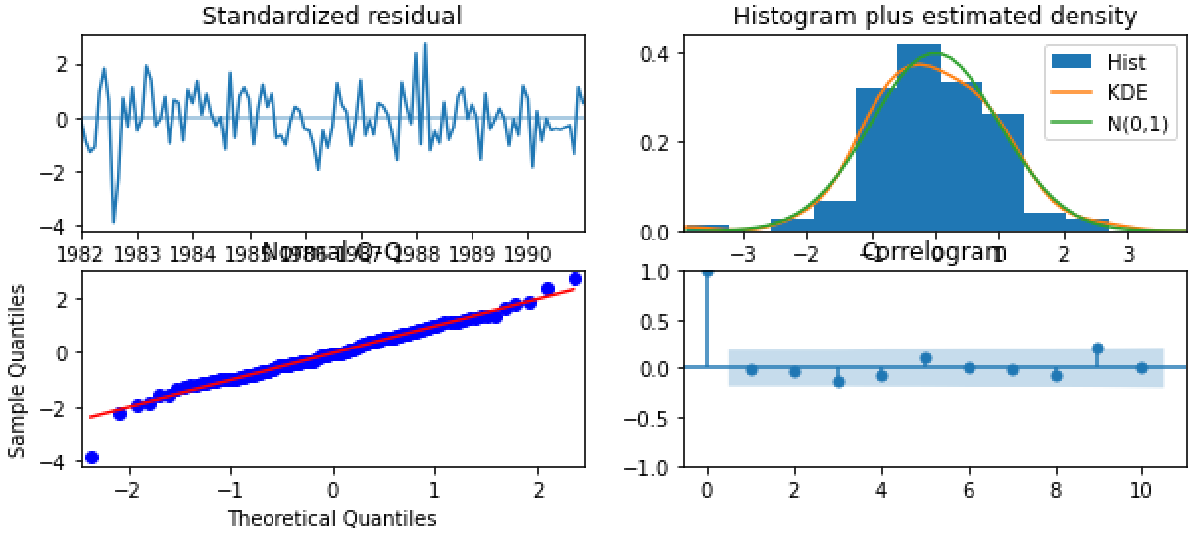
Fitting best value for Rose value

```
SARIMAX Results
==========================================================================================
Dep. Variable:                              Rose   No. Observations:                   132
Model:             SARIMAX(1, 1, 3)x(2, 0, 3, 6)   Log Likelihood                -466.842
Date:                           Sun, 20 Mar 2022   AIC                            953.685
Time:                                   20:20:10   BIC                            980.598
Sample:                               01-31-1980   HQIC                           964.599
                                    - 12-31-1990
Covariance Type:                             opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2239      0.499     -0.448      0.654      -1.203       0.755
ma.L1         -0.5388    609.478     -0.001      0.999   -1195.095    1194.017
ma.L2         -0.4138    281.140     -0.001      0.999    -551.438     550.611
ma.L3         -0.0474     28.890     -0.002      0.999     -56.671      56.576
ar.S.L6       -0.1229      0.057     -2.170      0.030      -0.234      -0.012
ar.S.L12       0.7304      0.057     12.771      0.000       0.618       0.842
ma.S.L6        0.1249      0.132      0.944      0.345      -0.134       0.384
ma.S.L12      -0.3300      0.122     -2.695      0.007      -0.570      -0.090
ma.S.L18       0.2257      0.131      1.725      0.085      -0.031       0.482
sigma2       288.1495   1.76e+05      0.002      0.999   -3.44e+05    3.45e+05
====================================================================================
Ljung-Box (Q):                       26.29   Jarque-Bera (JB):                  8.05
Prob(Q):                              0.95   Prob(JB):                          0.02
Heteroskedasticity (H):               0.60   Skew:                             -0.26
Prob(H) (two-sided):                  0.13   Kurtosis:                          4.23
```

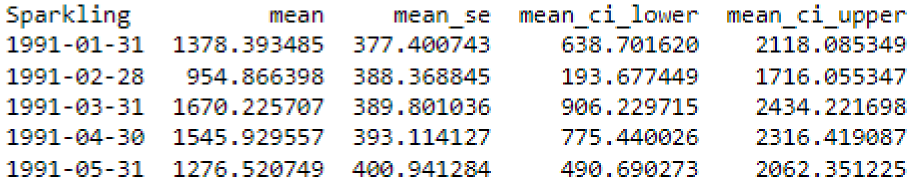
Plot diagnostics for SARIMA model for Sparkling Data



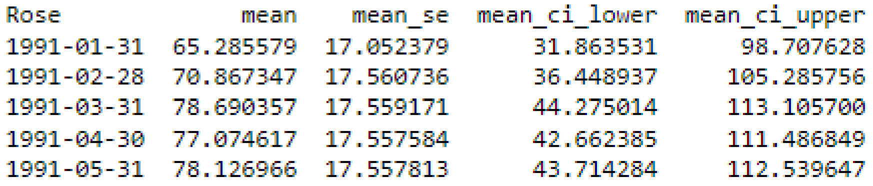
Plot diagnostics for SARIMA model for Rose Data



Predict on the Test Set using this model and evaluate the model.



```
Sparkling          mean     mean_se  mean_ci_lower  mean_ci_upper
1991-01-31  1378.393485  377.400743     638.701620    2118.085349
1991-02-28   954.866398  388.368845     193.677449    1716.055347
1991-03-31  1670.225707  389.801036     906.229715    2434.221698
1991-04-30  1545.929557  393.114127     775.440026    2316.419087
1991-05-31  1276.520749  400.941284     490.690273    2062.351225
```



```
Rose             mean    mean_se  mean_ci_lower  mean_ci_upper
1991-01-31  65.285579  17.052379      31.863531      98.707628
1991-02-28  70.867347  17.560736      36.448937     105.285756
1991-03-31  78.690357  17.559171      44.275014     113.105700
1991-04-30  77.074617  17.557584      42.662385     111.486849
1991-05-31  78.126966  17.557813      43.714284     112.539647
```

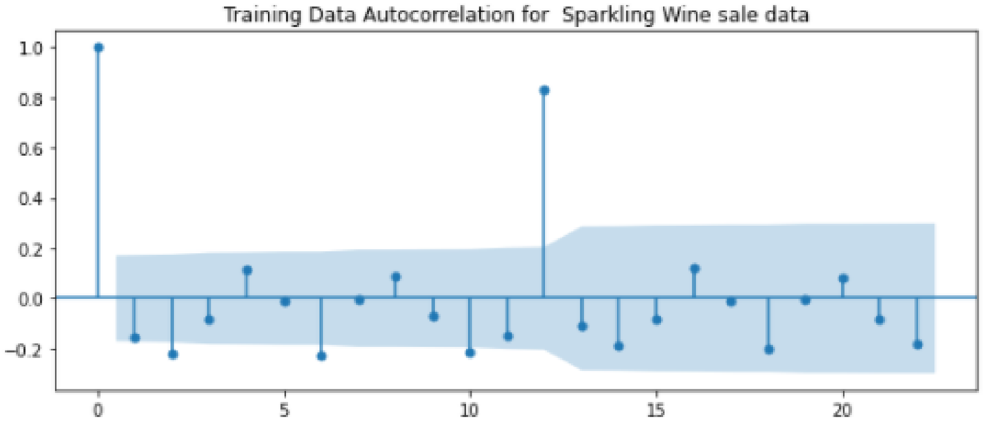
Sparkling wine sale data SARIMA RMSE: 735.8331873635088
Sparkling wine sale data SARIMA MAPE: 32.11660171496043

Rose wine sale data SARIMA RMSE: 30.9467970263849
Rose wine sale data SARIMA MAPE: 63.310743577469964

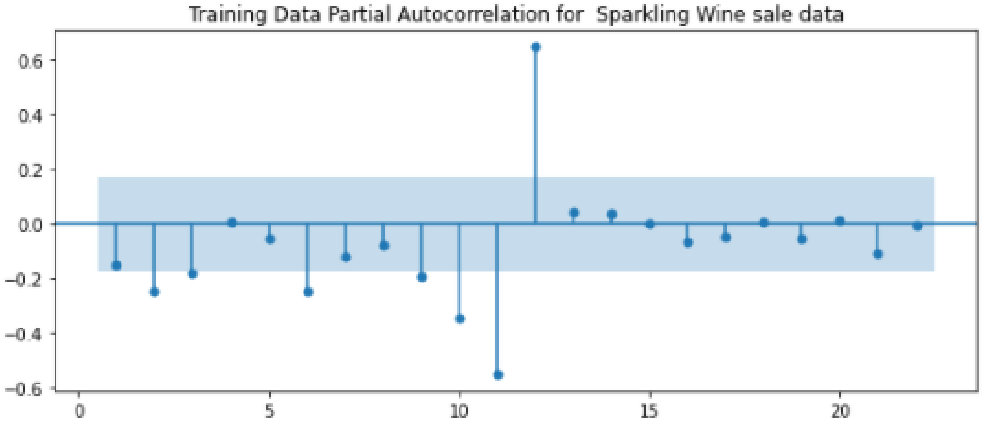
7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.

Build a version of the ARIMA model for which the best parameters are selected by looking at the ACF and the PACF plots.

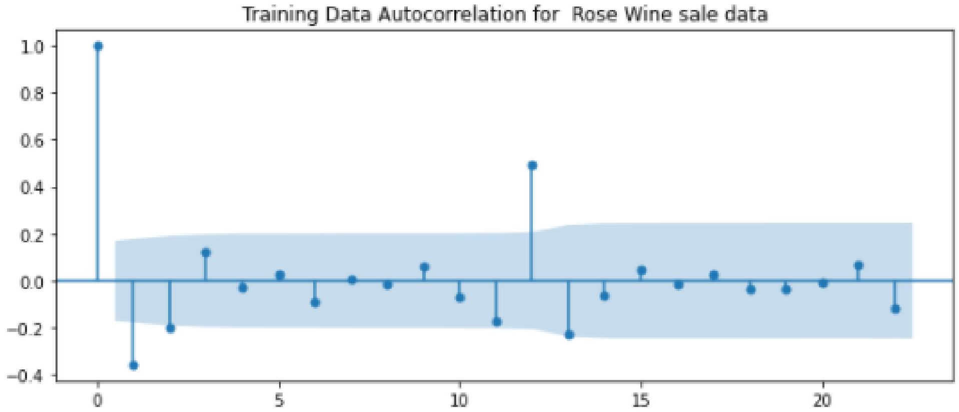
Let us look at the ACF and the PACF plots once more.



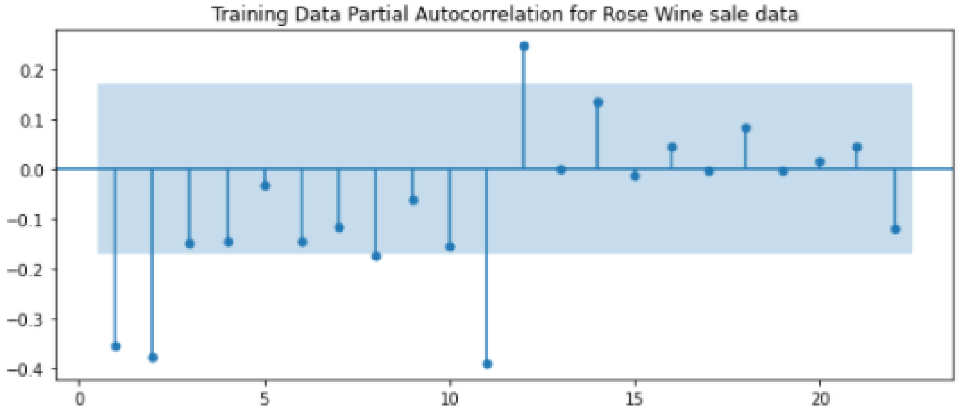
Training Data Autocorrelation for Sparkling Wine sale data



Training Data Partial Autocorrelation for Sparkling Wine sale data

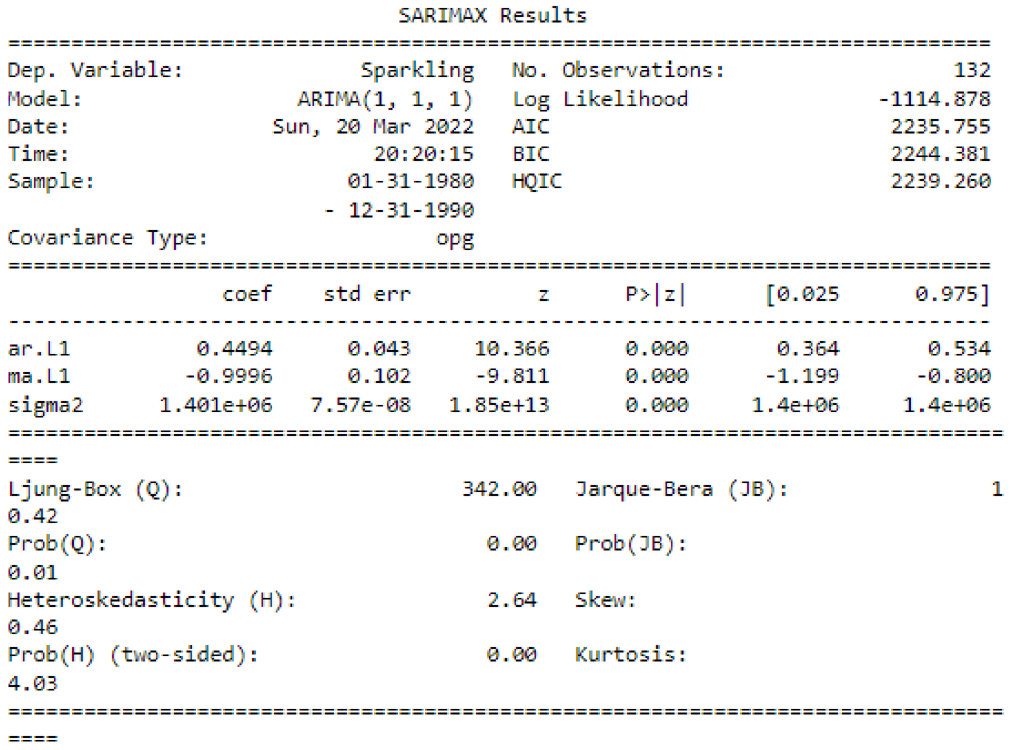


Training Data Autocorrelation for Rose Wine sale data



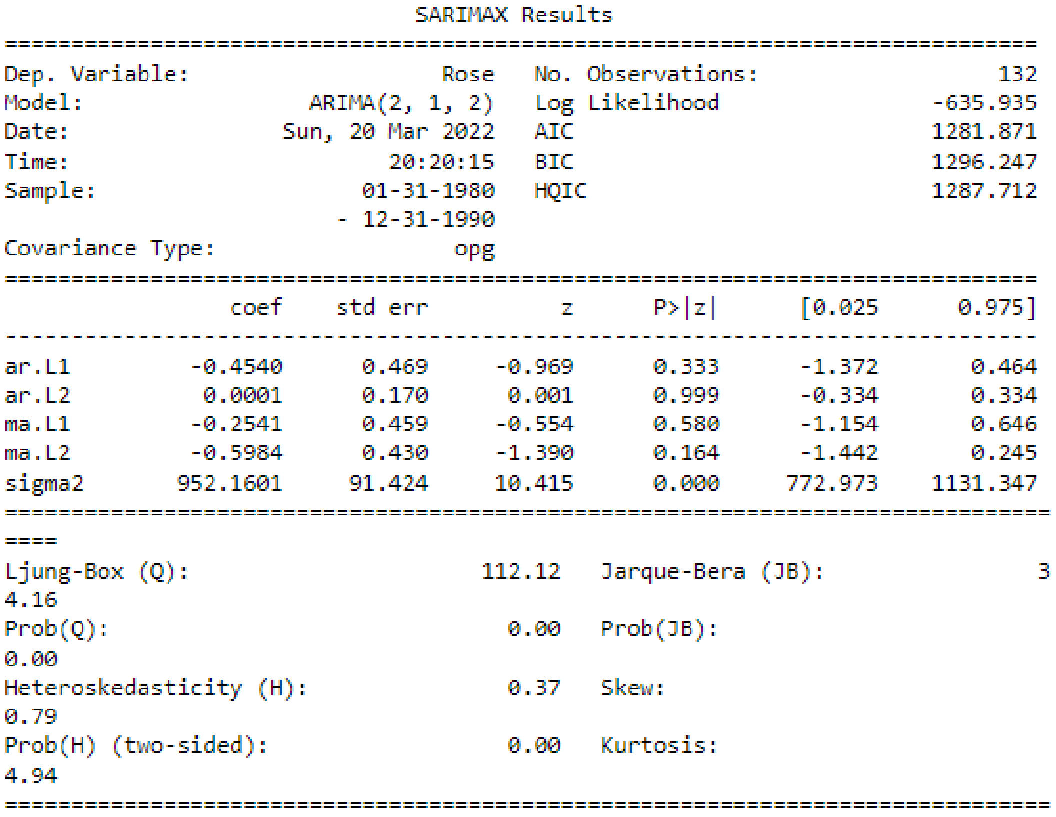
Training Data Partial Autocorrelation for Rose Wine sale data

By looking at the above plots, we will take the value of p and q to be 1 and 1 respectively.
Sparkling Data



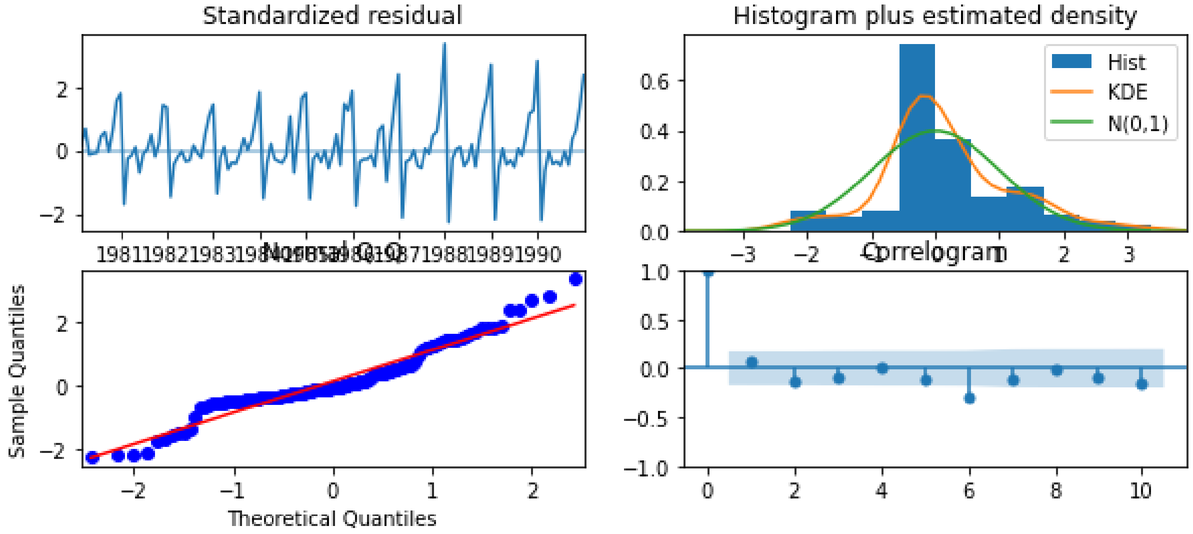
```
                               SARIMAX Results
==============================================================================
Dep. Variable:            Sparkling   No. Observations:                  132
Model:               ARIMA(1, 1, 1)   Log Likelihood               -1114.878
Date:              Sun, 20 Mar 2022   AIC                           2235.755
Time:                      20:20:15   BIC                           2244.381
Sample:                  01-31-1980   HQIC                          2239.260
                       - 12-31-1990
Covariance Type:                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4494      0.043     10.366      0.000       0.364       0.534
ma.L1         -0.9996      0.102     -9.811      0.000      -1.199      -0.800
sigma2      1.401e+06   7.57e-08   1.85e+13      0.000     1.4e+06     1.4e+06
===============================================================================
====
Ljung-Box (Q):                      342.00   Jarque-Bera (JB):                  1
0.42
Prob(Q):                              0.00   Prob(JB):
0.01
Heteroskedasticity (H):               2.64   Skew:
0.46
Prob(H) (two-sided):                  0.00   Kurtosis:
4.03
===============================================================================
====
```

Rose Data

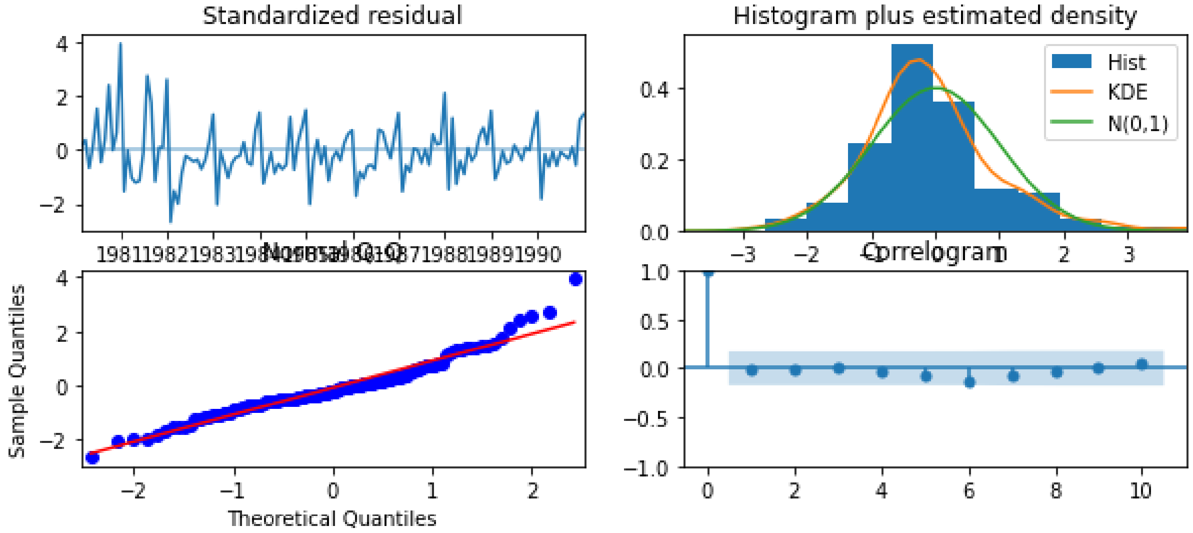


```
                               SARIMAX Results
==============================================================================
Dep. Variable:                 Rose   No. Observations:                  132
Model:               ARIMA(2, 1, 2)   Log Likelihood                -635.935
Date:              Sun, 20 Mar 2022   AIC                           1281.871
Time:                      20:20:15   BIC                           1296.247
Sample:                  01-31-1980   HQIC                          1287.712
                       - 12-31-1990
Covariance Type:                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4540      0.469     -0.969      0.333      -1.372       0.464
ar.L2          0.0001      0.170      0.001      0.999      -0.334       0.334
ma.L1         -0.2541      0.459     -0.554      0.580      -1.154       0.646
ma.L2         -0.5984      0.430     -1.390      0.164      -1.442       0.245
sigma2       952.1601     91.424     10.415      0.000     772.973    1131.347
===============================================================================
====
Ljung-Box (Q):                      112.12   Jarque-Bera (JB):                  3
4.16
Prob(Q):                              0.00   Prob(JB):
0.00
Heteroskedasticity (H):               0.37   Skew:
0.79
Prob(H) (two-sided):                  0.00   Kurtosis:
4.94
===============================================================================
```

Let us analyse the residuals from the various diagnostics plot.
<u>Sparkling Data</u>



Rose Data



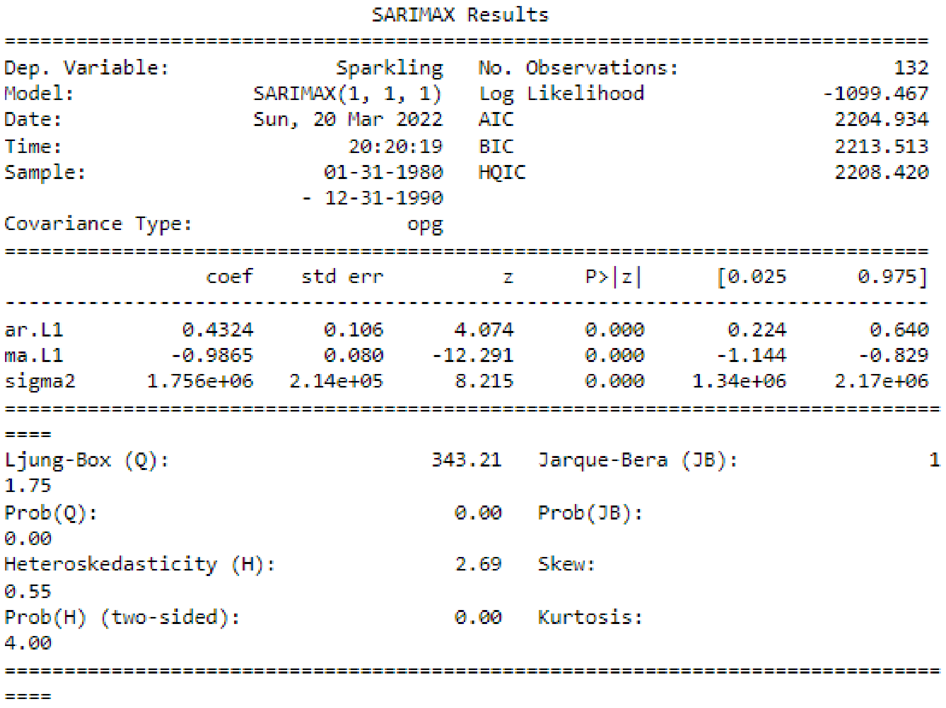
Predict on the Test Set using this model and evaluate the model.

Sparkling wine sale data with Arima with ACF and PACF model, RMSE: 1319.936733605669

Rose wine sale data with Arima with ACF and PACF model, RMSE: 36.87119661928125

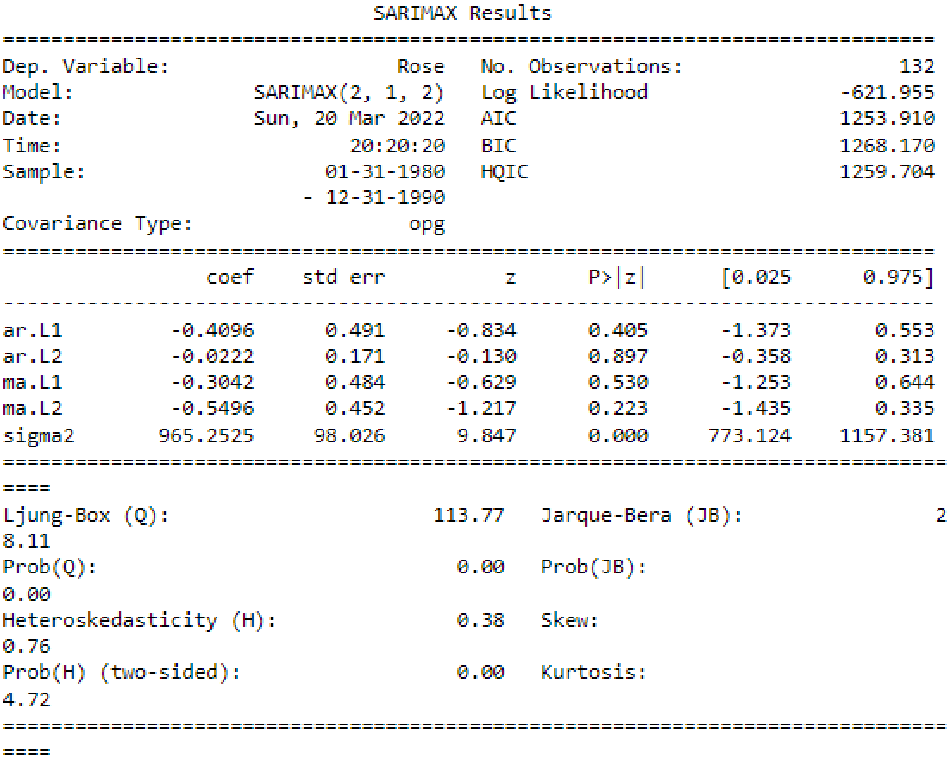
Build a version of the SARIMA model for which the best parameters are selected by looking at the ACF and the PACF plots. - Seasonality at 6.

For sparkling



```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Sparkling   No. Observations:                  132
Model:               SARIMAX(1, 1, 1)   Log Likelihood               -1099.467
Date:                Sun, 20 Mar 2022   AIC                           2204.934
Time:                        20:20:19   BIC                           2213.513
Sample:                    01-31-1980   HQIC                          2208.420
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4324      0.106      4.074      0.000       0.224       0.640
ma.L1         -0.9865      0.080    -12.291      0.000      -1.144      -0.829
sigma2      1.756e+06   2.14e+05      8.215      0.000    1.34e+06    2.17e+06
===============================================================================
====
Ljung-Box (Q):                      343.21   Jarque-Bera (JB):                 1
1.75
Prob(Q):                              0.00   Prob(JB):
0.00
Heteroskedasticity (H):               2.69   Skew:
0.55
Prob(H) (two-sided):                  0.00   Kurtosis:
4.00
===============================================================================
====
```

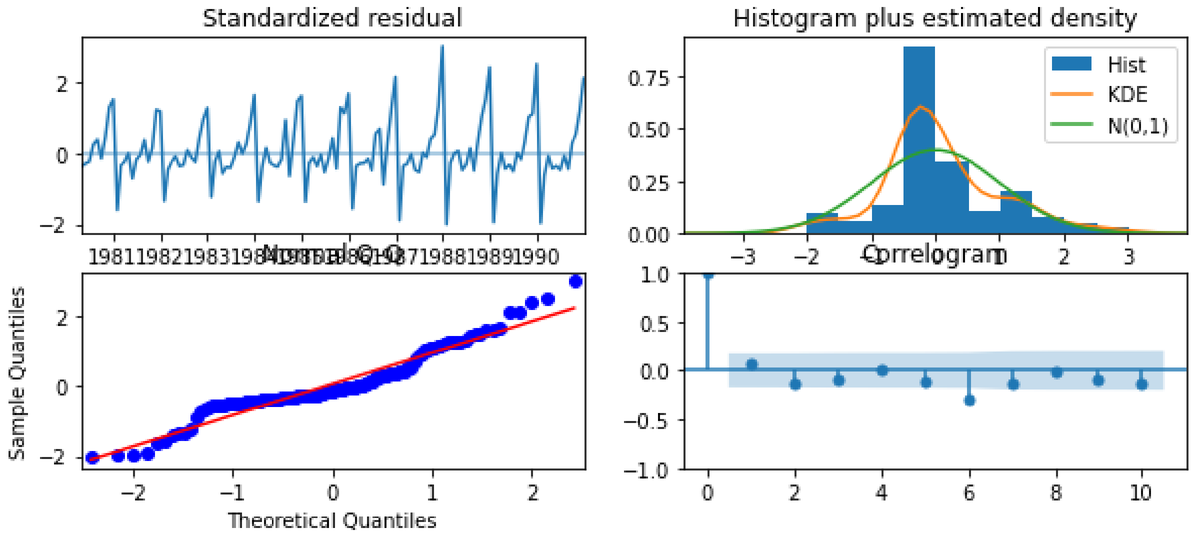
For Rose



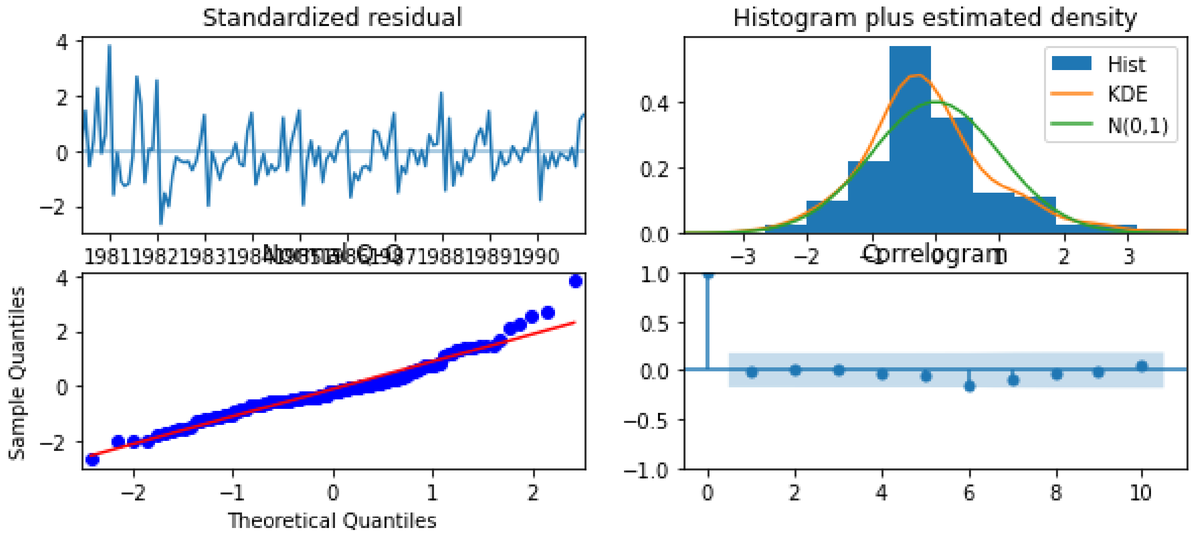
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   Rose   No. Observations:                  132
Model:               SARIMAX(2, 1, 2)   Log Likelihood                -621.955
Date:                Sun, 20 Mar 2022   AIC                           1253.910
Time:                        20:20:20   BIC                           1268.170
Sample:                    01-31-1980   HQIC                          1259.704
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4096      0.491     -0.834      0.405      -1.373       0.553
ar.L2         -0.0222      0.171     -0.130      0.897      -0.358       0.313
ma.L1         -0.3042      0.484     -0.629      0.530      -1.253       0.644
ma.L2         -0.5496      0.452     -1.217      0.223      -1.435       0.335
sigma2       965.2525     98.026      9.847      0.000     773.124    1157.381
===============================================================================
====
Ljung-Box (Q):                      113.77   Jarque-Bera (JB):                 2
8.11
Prob(Q):                              0.00   Prob(JB):
0.00
Heteroskedasticity (H):               0.38   Skew:
0.76
Prob(H) (two-sided):                  0.00   Kurtosis:
4.72
===============================================================================
====
```

Let us analyse the residuals from the various diagnostics plot.

<u>Sparkling Data</u>



Rose Data



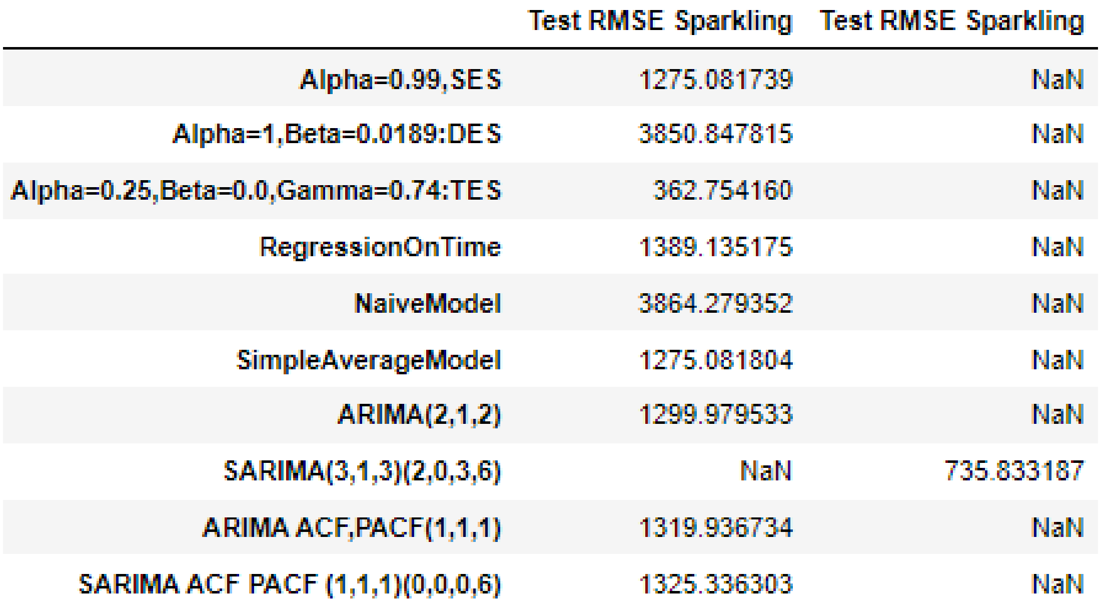
Predict on the Test Set using this model and evaluate the model.

RMSE for Sparkling Data: 1325.3363028682475

RMSE for Rose Data: 36.807209578009605

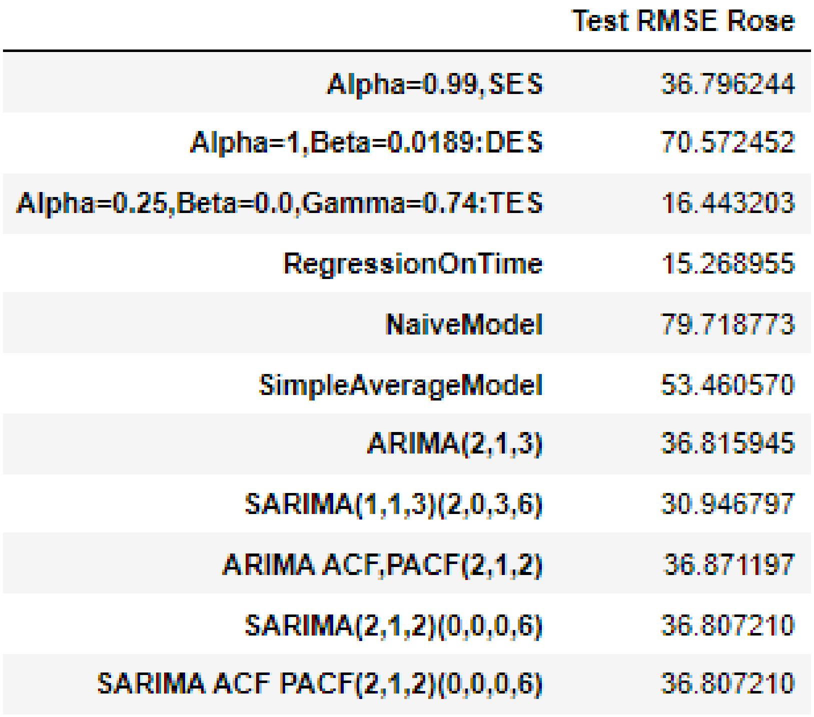
8. Build a table (create a data frame) with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

For Sparkling Wine sale Data



	Test RMSE Sparkling	Test RMSE Sparkling
Alpha=0.99,SES	1275.081739	NaN
Alpha=1,Beta=0.0189:DES	3850.847815	NaN
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	362.754160	NaN
RegressionOnTime	1389.135175	NaN
NaiveModel	3864.279352	NaN
SimpleAverageModel	1275.081804	NaN
ARIMA(2,1,2)	1299.979533	NaN
SARIMA(3,1,3)(2,0,3,6)	NaN	735.833187
ARIMA ACF,PACF(1,1,1)	1319.936734	NaN
SARIMA ACF PACF (1,1,1)(0,0,0,6)	1325.336303	NaN

For Rose Wine sale Data



	Test RMSE Rose
Alpha=0.99,SES	36.796244
Alpha=1,Beta=0.0189:DES	70.572452
Alpha=0.25,Beta=0.0,Gamma=0.74:TES	16.443203
RegressionOnTime	15.268955
NaiveModel	79.718773
SimpleAverageModel	53.460570
ARIMA(2,1,3)	36.815945
SARIMA(1,1,3)(2,0,3,6)	30.946797
ARIMA ACF,PACF(2,1,2)	36.871197
SARIMA(2,1,2)(0,0,0,6)	36.807210
SARIMA ACF PACF(2,1,2)(0,0,0,6)	36.807210

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

Here, we have a scenario where our training data was stationary but our full data was not stationary. So, we will use the same parameters as our training data but with adding a level of differencing which is needed for the data to be stationary.

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

Step1:
Initializing the Double Exponential Smoothing Model for sparkling wine sale data
Step2:
Fitting the model
==Holt Winters model Exponential Smoothing Estimated Parameters for sparkling wine sale data ==

```
{'smoothing_level': 0.05263157894736842, 'smoothing_slope':
0.05263157894736842, 'smoothing_seasonal': 0.3684210526315789,
'damping_slope': nan, 'initial_level': 1580.0, 'initial_slope': 0.01, 'initial_seasons': array([
106.,   11.,  724.,  132., -109., -203.,  386.,  873.,  404.,
      1016., 2507
```

Forecasting using this model for the duration of the whole set(Sparkling Wine sale data)
Evaluate the model on the whole data and predict 12 months into the future (till the end of next year).

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

There is a need for increase the sale by reducing price or some other parameter