TIME SERIES FORECASTING

SPARKLING WINE SALES - PROJECT REPORT

BY,

RAGAVEDHNI K R

1. **Read the data as an appropriate Time Series data and plot the data.**
   - The data is read from the 'Sparkling.csv' file, the initial set of rows is as below.

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

   - Creating a DateTimeIndex using date_range() from Pandas for the entire length of the dataset.

```
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')
```

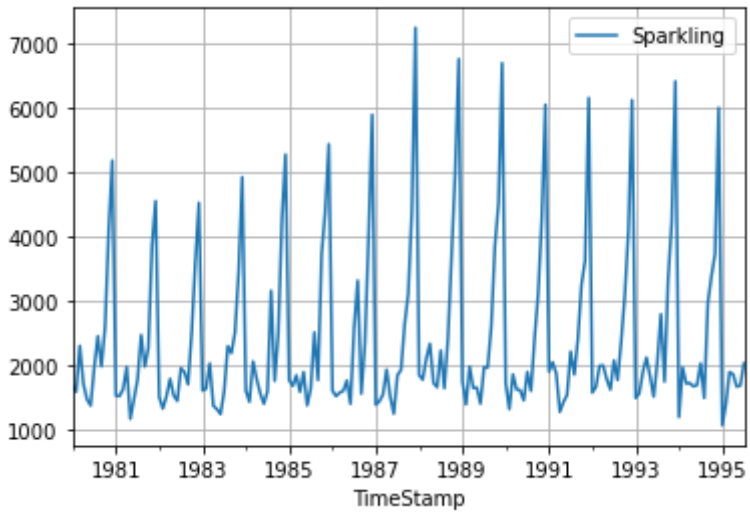
   - Including that as a TimeStamp column to the dataframe, as it is required to work on Time Series data.

	YearMonth	Sparkling	TimeStamp
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

   - Dropping the unwanted column and set the index of the dataframe as TimeStamp using the set_index(), to work with Time series data.

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

- We can plot the dataframe as a time series data as below.



- The X-axis is the TimeStamp column and the Y-axis is the count of sparkling wine sales across the years.

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

- The basic exploratory data analysis is made on the data using describe().

	Sparkling
count	187.000000
mean	2402.417112
std	1295.111540
min	1070.000000
25%	1605.000000
50%	1874.000000
75%	2549.000000
max	7242.000000

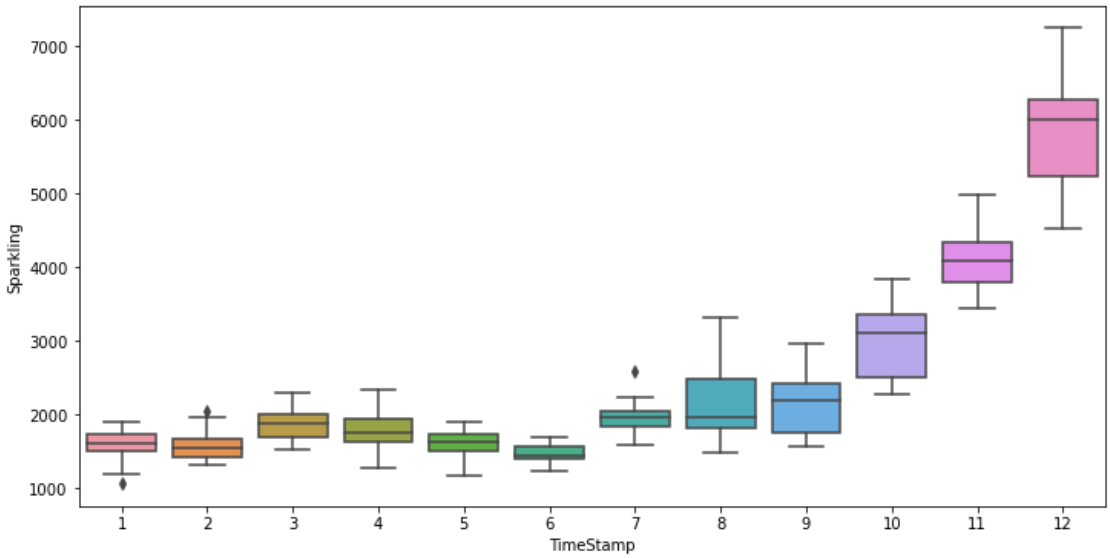
- The mean and median of the data can be seen as below.

```
Mean of the data Sparkling    2402.417112
dtype: float64
Median of the data Sparkling    1874.0
dtype: float64
```

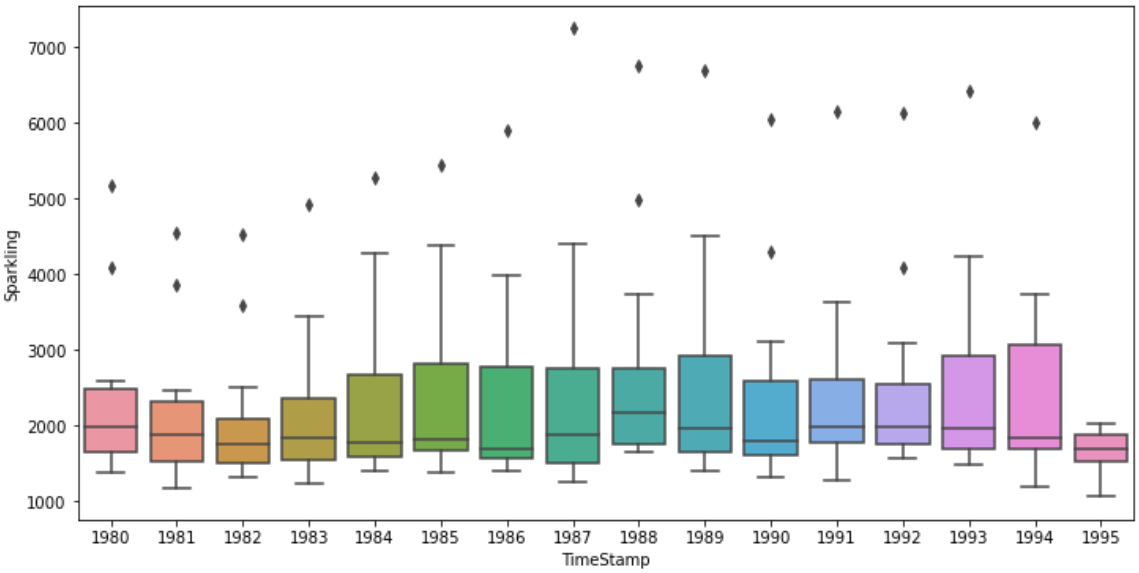
- Checking the null values in the data.

```
Checking the null values in the data: Sparkling    0
dtype: int64
```

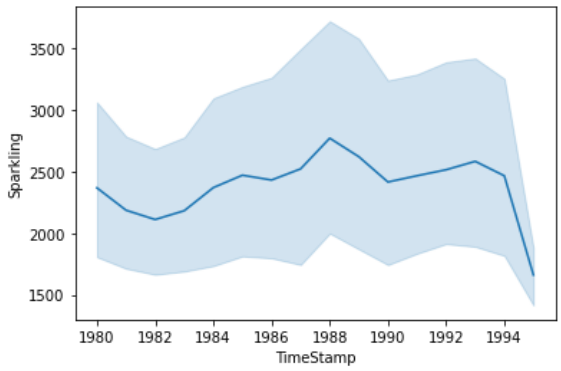
- Plotting a monthly boxplot for the Sparkling wine sales taking all the years into account. We can see that the sales are gradually increasing till July and sudden raise from August. The sale is higher in the December as many celebrations are held.

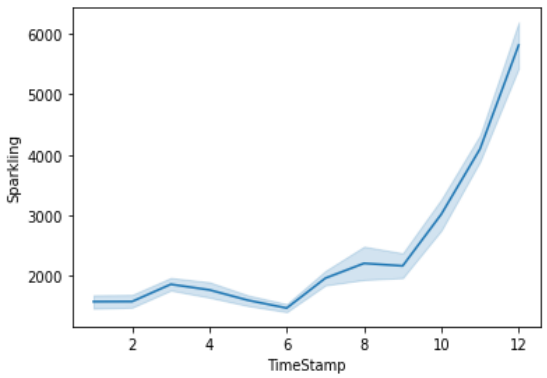


- Plotting year on year boxplot for the sparkling wine sales. The sale is nearly the same across all the years.

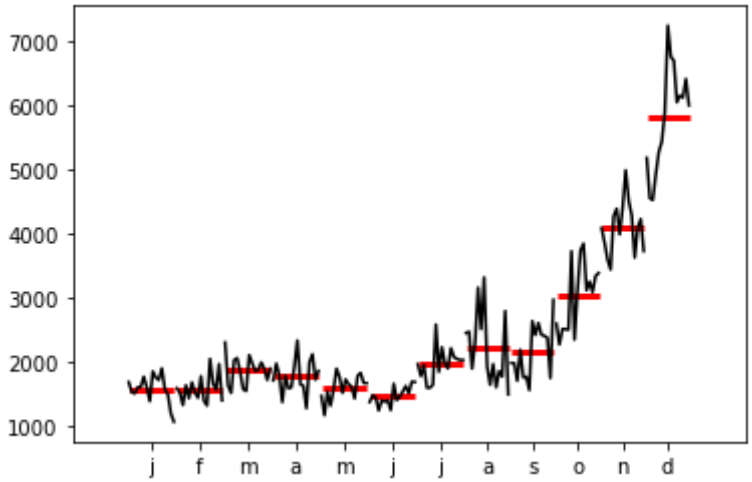


- Plotting yearly line plot across all the years and monthly line plot for all years.

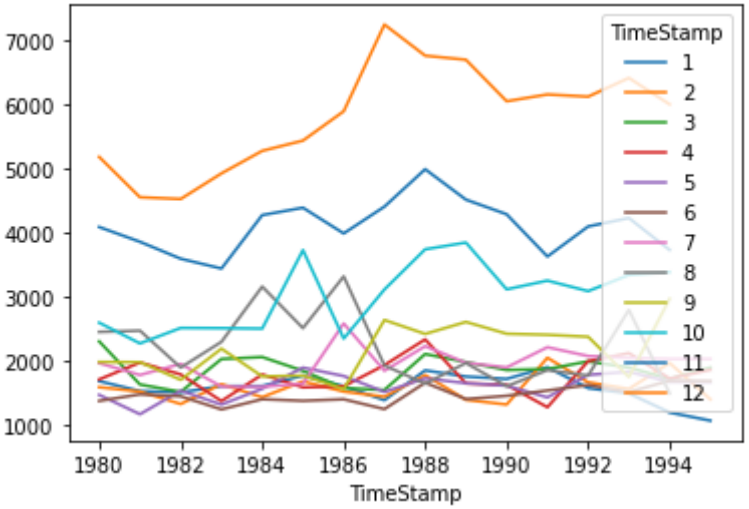




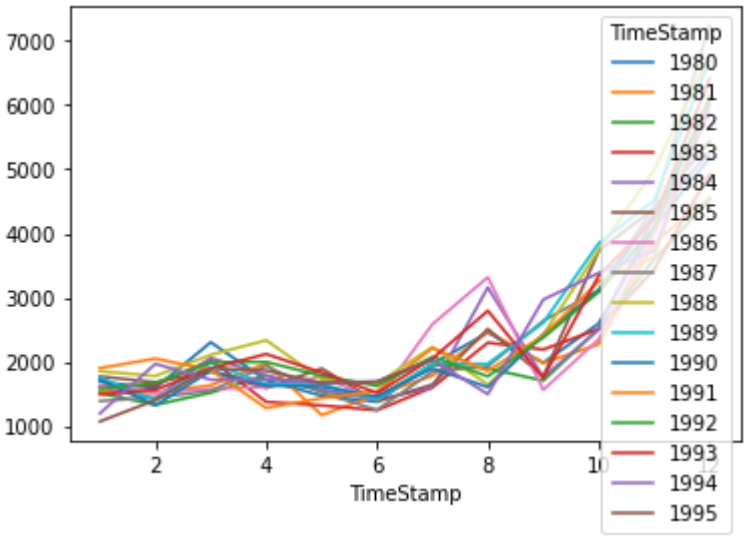
- Plotting the monthplot from statsmodels.graphics.tsaplots package.



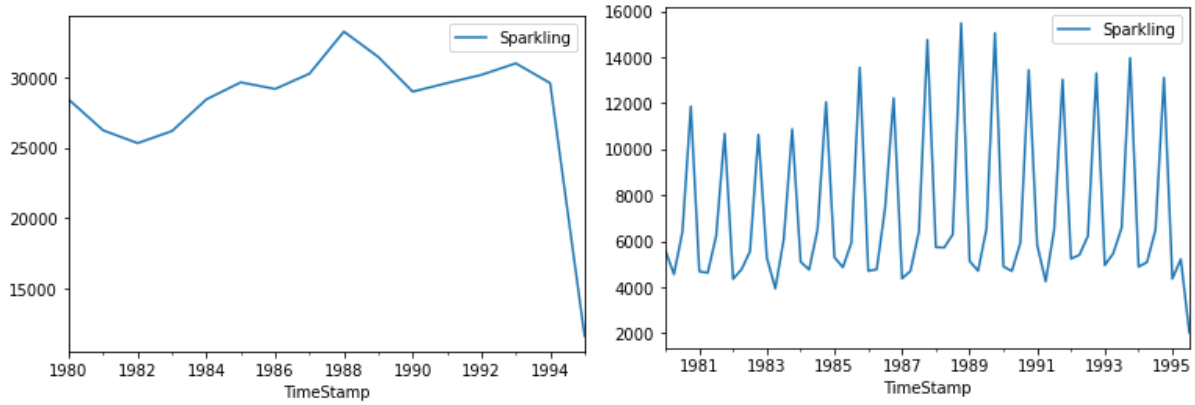
- Plotting the Time Series according to different months for across the years.



- Plotting the Time series according to across all the months for different years.



- Reading and plotting the data yearly and quarterly using the resample() and mean().

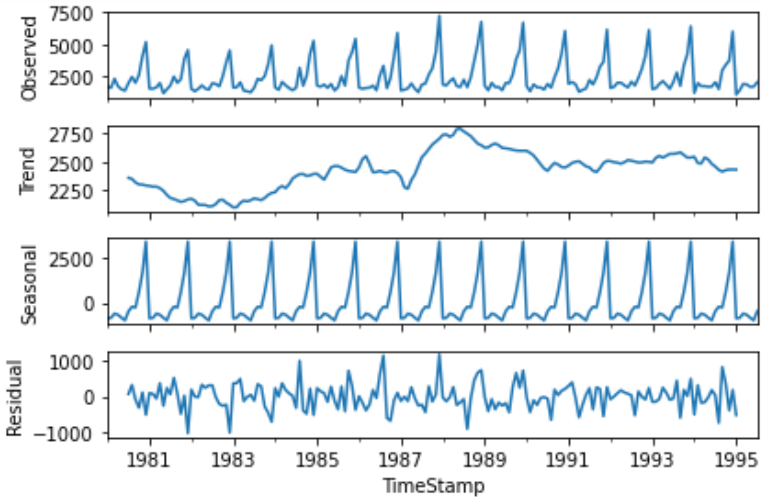


DECOMPOSITION:

- Decomposing the data using seasonal_decompose() from the statsmodels.tsa.seasonal module. Checking the model as additive or multiplicative.

ADDITIVE MODEL:

- If the seasonality and residual components are independent of the trend, then it is an additive series.



- Checking on the first 10 rows in trends, seasonal, residual components separately.

TREND

TimeStamp	Sparkling
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	2360.666667
1980-08-31	2351.333333
1980-09-30	2320.541667
1980-10-31	2303.583333

SEASONAL

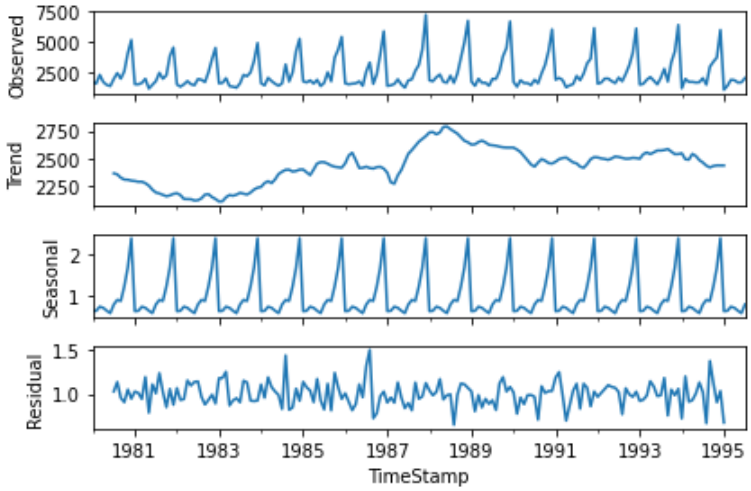
TimeStamp	Sparkling
1980-01-31	-854.260599
1980-02-29	-830.350678
1980-03-31	-592.356630
1980-04-30	-658.490559
1980-05-31	-824.416154
1980-06-30	-967.434011
1980-07-31	-465.502265
1980-08-31	-214.332821
1980-09-30	-254.677265
1980-10-31	599.769957

RESIDUAL

TimeStamp	Sparkling
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	70.835599
1980-08-31	315.999487
1980-09-30	-81.864401
1980-10-31	-307.353290

MULTIPLICATIVE MODEL:

- If the seasonality and residual components are in dependent, i.e., they fluctuate on trend, then it is a multiplicative series.



- Checking on the first 10 rows in trends, seasonal, residual components separately.

TREND

TimeStamp	Sparkling
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	2360.666667
1980-08-31	2351.333333
1980-09-30	2320.541667
1980-10-31	2303.583333

SEASONAL

TimeStamp	Sparkling
1980-01-31	0.649843
1980-02-29	0.659214
1980-03-31	0.757440
1980-04-30	0.730351
1980-05-31	0.660609
1980-06-30	0.603468
1980-07-31	0.809164
1980-08-31	0.918822
1980-09-30	0.894367
1980-10-31	1.241789

RESIDUAL

TimeStamp	Sparkling
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	1.029230
1980-08-31	1.135407
1980-09-30	0.955954
1980-10-31	0.907513

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

- The data is split into train data for training the model and test data for predicting the data using the model.
- Taking 70% of the data as train data (till the year 1990) and 30% of the data as test data (from the year 1991).
- Checking the shape of the train data and test data.

```
Shape of the train data (132, 1)
Shape of the test data (55, 1)
```

- Viewing the initial sets and end sets of rows from train and test data.

First few rows of Training Data

TimeStamp	Sparkling
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

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

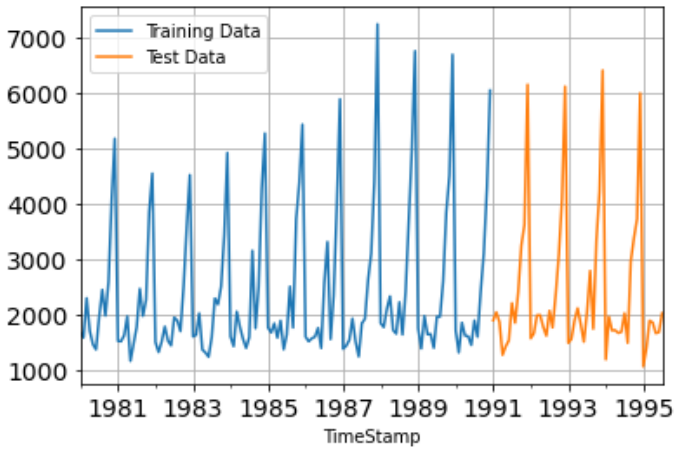
First few rows of Test Data

TimeStamp	Sparkling
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

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

- Checking the plots of the train and test data.



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

- Building the Simple Exponential Smoothing model (SES), Double Exponential Smoothing model (DES), Triple Exponential Smoothing model (TES), Linear Regression model, Naive model, Simple Average model and Moving Average model.

SIMPLE EXPONENTIAL SMOOTHING (SES):

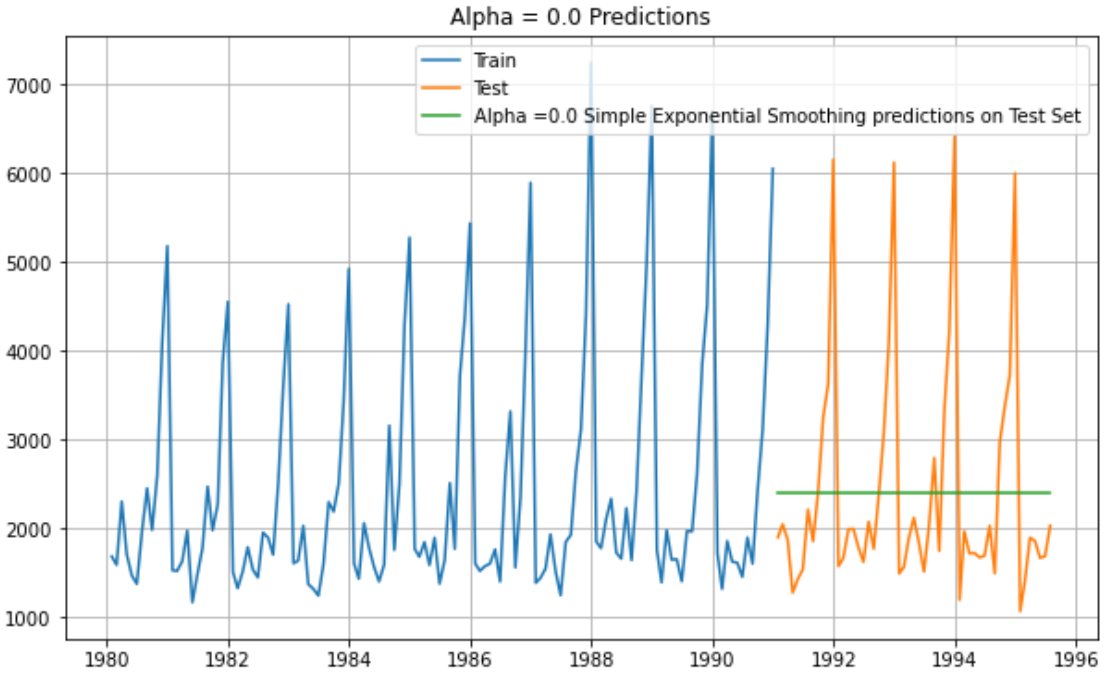
- Apply SES model on the Time Series data when there is no trend or seasonality is present in the data. Though case is almost non-available, we try this to understand how the smoothing parameter (α) controls the performance of the method.
- We create the SES model using the train data and fitting the model to maximize the log-likelihood.
- We check the optimal parameters returned by the model.

```
{'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}
```

- Using the fitted model on the training data, we forecast on the test data.
- We set the parameter as the number of out of sample forecasts from the end of the sample (test 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
```

- We can check the forecasted values along with train and test values in the below plot.



- We check the RMSE of the SES model on the test data.

SES RMSE: 1275.0817392898339

DOUBLE EXPONENTIAL SMOOTHING (DES):

- DES is an extension of SES, which is applicable when the data has trend but no seasonality.
- The Level and Trend component are controlled by α and β smoothing parameter respectively.
- DES model can be initialised by the train data and setting the exponential parameter accordingly and DES is fitted.
- We check the smoothing parameters for DES as below.

When Exponential= False,

```
{'smoothing_level': 0.64781229341859,
 'smoothing_slope': 0.0,
 'smoothing_seasonal': nan,
 'damping_slope': nan,
 'initial_level': 1686.0826076865374,
 'initial_slope': 27.062658731812196,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

When Exponential= True,

```
{'smoothing_level': 0.7307090782489399,
 'smoothing_slope': 0.0,
 'smoothing_seasonal': nan,
 'damping_slope': nan,
 'initial_level': 1687.4654804942545,
 'initial_slope': 0.9409001417447868,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- We forecast using DES model for the duration of the test data.

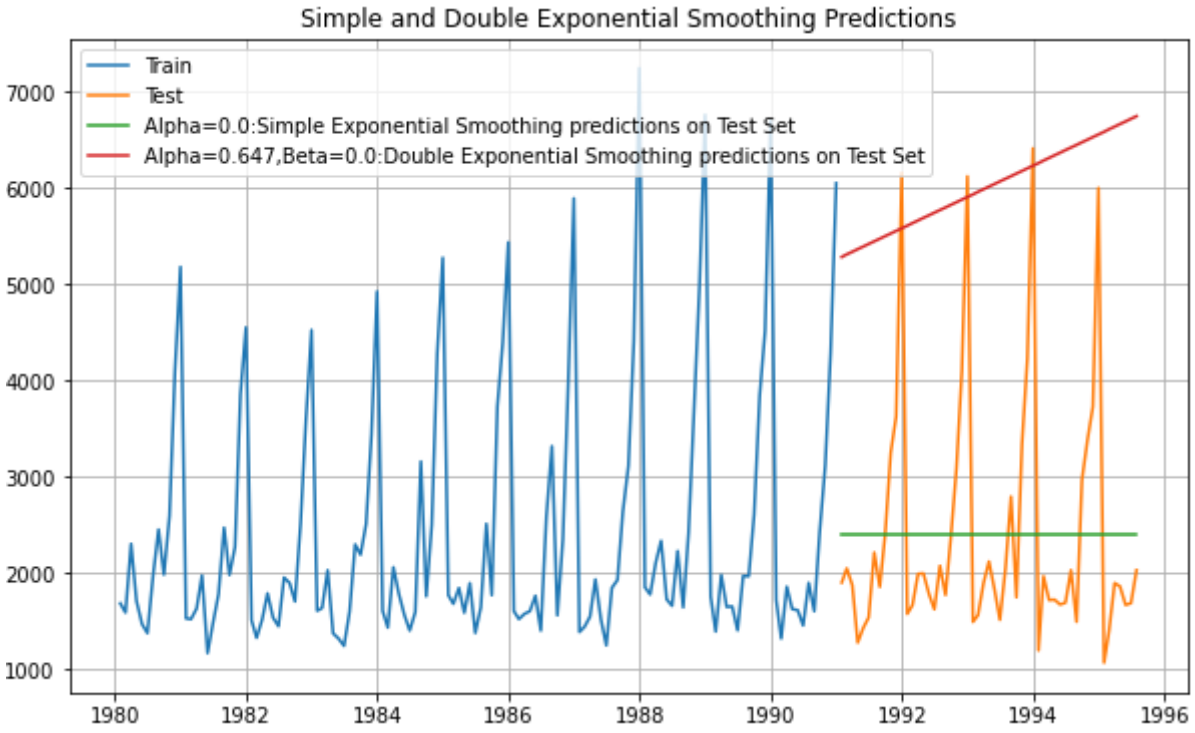
When Exponential= False,

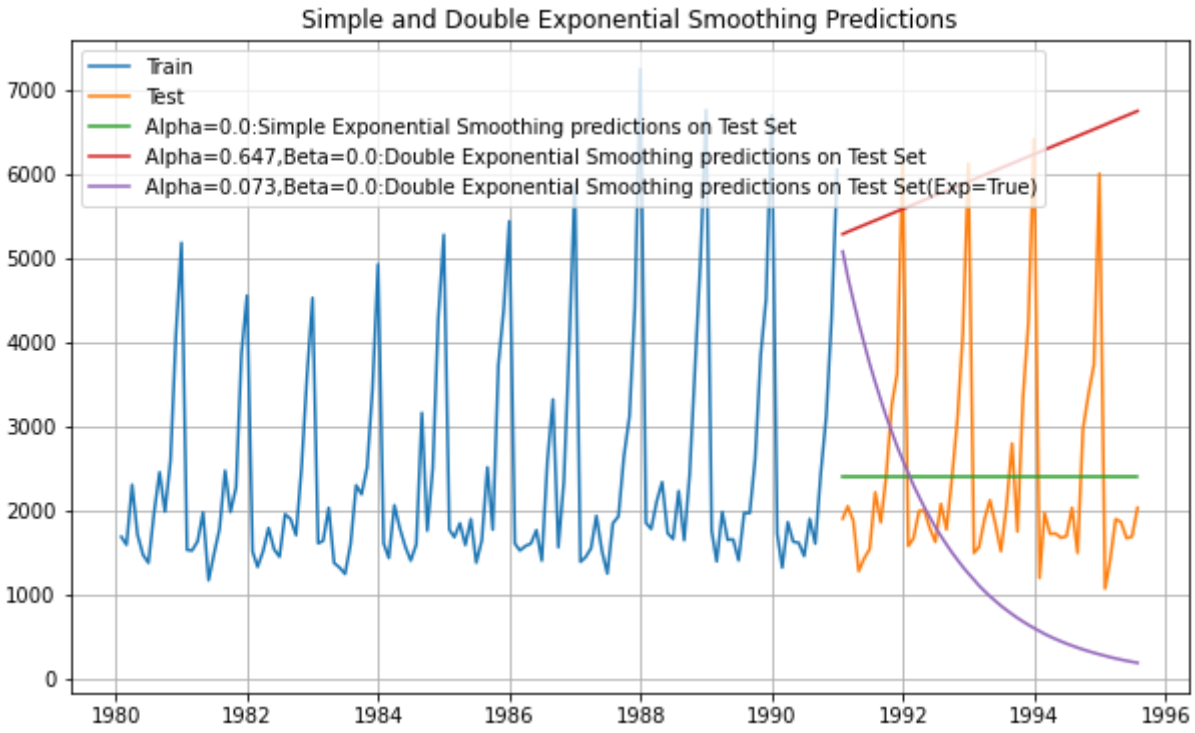
1991-01-31	5281.502122
1991-02-28	5308.564780
1991-03-31	5335.627439
1991-04-30	5362.690098
1991-05-31	5389.752757
1991-06-30	5416.815415
1991-07-31	5443.878074
1991-08-31	5470.940733
1991-09-30	5498.003392
1991-10-31	5525.066050
1991-11-30	5552.128709
1991-12-31	5579.191368

When Exponential= True,

1991-01-31	5075.011579
1991-02-28	4775.079114
1991-03-31	4492.872615
1991-04-30	4227.344480
1991-05-31	3977.509021
1991-06-30	3742.438801
1991-07-31	3521.261199
1991-08-31	3313.155161
1991-09-30	3117.348161
1991-10-31	2933.113326
1991-11-30	2759.766744
1991-12-31	2596.664921

- We can view the SES and DES forecast values along with the train data and test data from the plot below.





- We can see that the DES performs better than SES as the trend is considered for forecasting.
- We check the RMSE of the test data using both the DES models as below.
- **When Exponential= False,**

```
DES RMSE: 3851.1294387304633
```

- **When Exponential= True,**

```
DES RMSE: 2135.0566757901315
```

- We can see that the RMSE is lower when the Exponential parameter is set as True in DES model.

TRIPLE EXPONENTIAL MODEL (TES) or HOLT-WINTER'S METHOD:

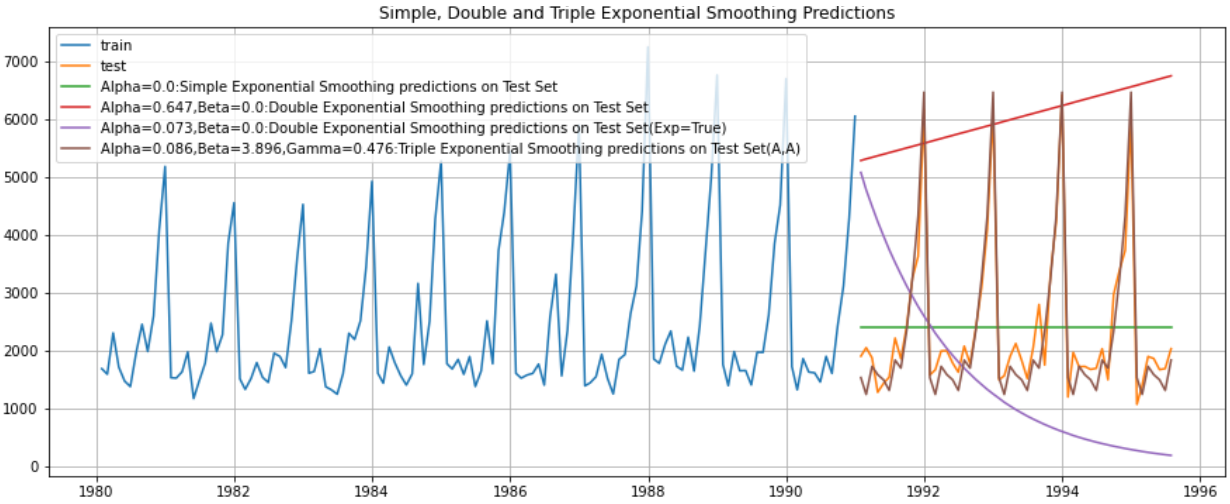
- Considering the Level, Trend, Seasonality components.
- As the seasonality can be additive or multiplicative, the TES model can be additive or multiplicative.
- We initialize the TES model using the train data and setting both the trend and seasonality as additive.
- We fit the TES model and check on the parameters as below.

```
{'smoothing_level': 0.08623089766184816,
 'smoothing_slope': 3.896462894854104e-10,
 'smoothing_seasonal': 0.4763368992674647,
 'damping_slope': nan,
 'initial_level': 1685.0465533546023,
 'initial_slope': 6.956424224149114e-10,
 'initial_seasons': array([  38.52881932,  -37.57358256,  464.69631723,  205.93129973,
        -140.6536742 , -156.86592282,  338.19548509,  856.39848276,
         402.99998598,  971.16197989, 2401.40352534, 3426.27237449]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- We forecast the model for the duration of the test data.

```
1991-01-31    1532.348599
1991-02-28    1241.314340
1991-03-31    1726.703013
1991-04-30    1584.248244
1991-05-31    1493.941871
1991-06-30    1311.395885
1991-07-31    1834.784340
1991-08-31    1696.096104
1991-09-30    2338.834507
1991-10-31    3249.213487
1991-11-30    4324.362476
1991-12-31    6461.265884
```

- We visualize the SES, DES, and TES models together in the plot below.



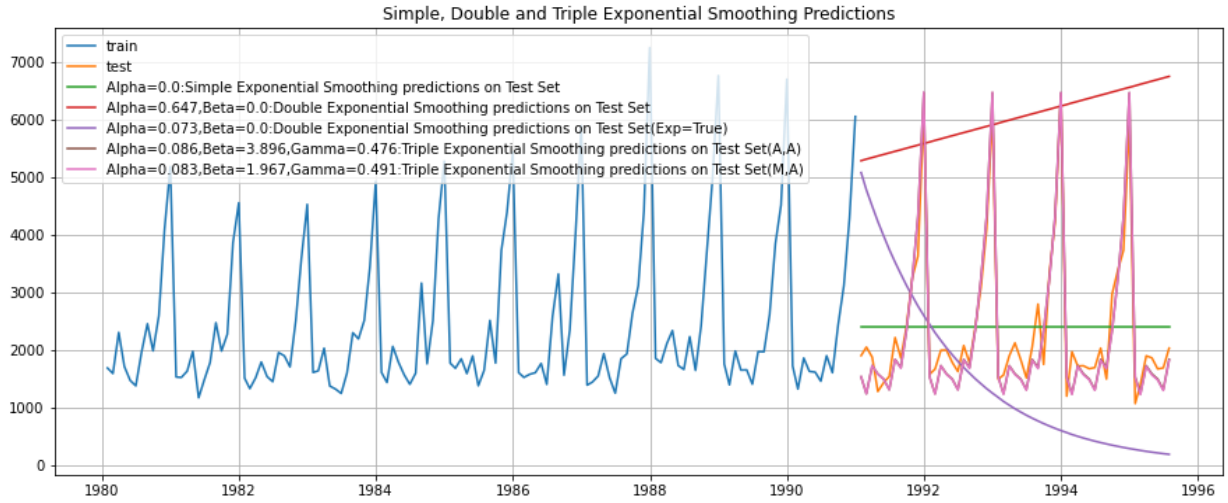
- We check on the RMSE of the test data using the TES model.

```
TES RMSE: 362.79502627634287
```

- The TES is initialized using the train data and setting the Trend as multiplicative and Seasonality as additive.
- The model is fitted and checked for the parameters.

```
{'smoothing_level': 0.08314532478008606,
 'smoothing_slope': 1.967568533823466e-09,
 'smoothing_seasonal': 0.4910344440693423,
 'damping_slope': nan,
 'initial_level': 1640.0000032568673,
 'initial_slope': 0.9998252081989077,
 'initial_seasons': array([  45.99999833,  -48.99999886,  663.99996952,   72.00001907,
       -168.99999657, -262.99998467,  326.00000136,  813.00000633,
        344.00000906,  956.00000263, 2446.99999404, 3538.99998397]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- The TES model is forecasted for the duration of the test data.
- We can visualize the forecast from the below plot.



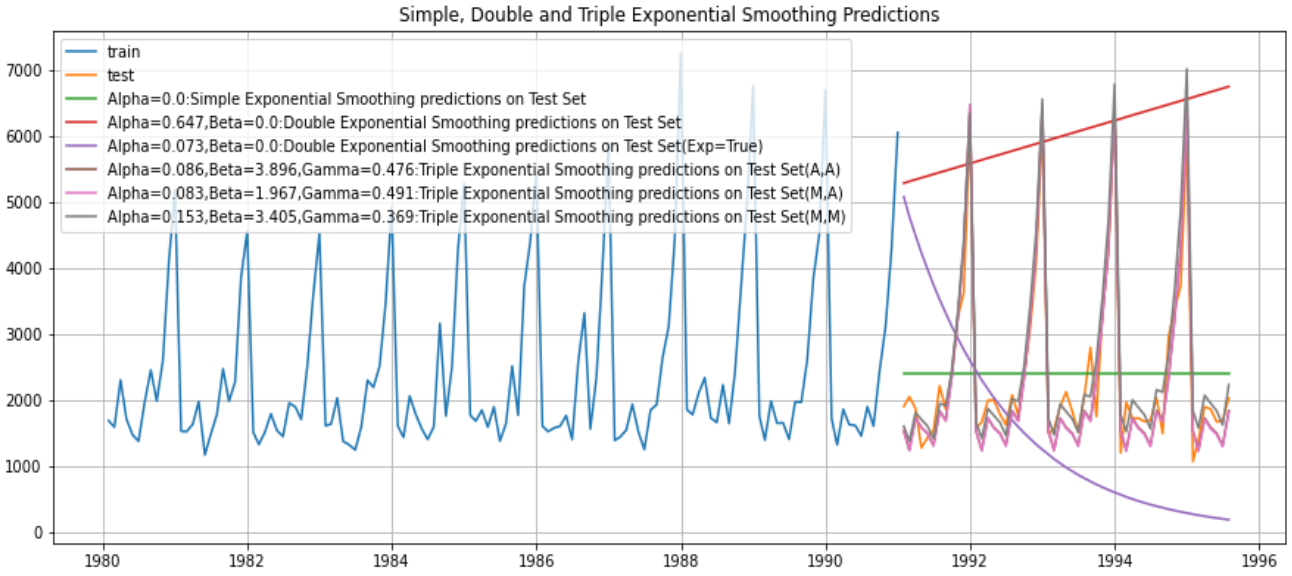
- The RMSE value of the test data using this TES model is as below.

```
TES RMSE: 366.94877416555914
```

- We initialize the TES model using the train data and setting the Trend as multiplicative and Seasonality as multiplicative.
- We fit the model and check on the parameters.

```
{'smoothing_level': 0.15341611498968094,
 'smoothing_slope': 3.405100155267661e-22,
 'smoothing_seasonal': 0.36909930433038646,
 'damping_slope': nan,
 'initial_level': 1640.0000825494478,
 'initial_slope': 1.002823637788699,
 'initial_seasons': array([1.0088922 , 0.96948306, 1.24312435, 1.13282486, 0.94014402,
        0.93860405, 1.22597071, 1.54639001, 1.27515569, 1.63524434,
        2.48873458, 3.1270512 ]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

- We forecast the model for the duration of the test data.
- We can visualize the model forecast ranges from the below plot.



- The RMSE of the Test data using TES model is checked.

```
TES RMSE: 392.95715278648663
```

LINEAR REGRESSION:

- For this particular linear regression, we are going to train the 'Rose' variable against the order of the occurrence. For this we need to modify our training and testing data before fitting it into a linear regression.
- We create 2 series of data called train_time and test_time for the independent variable and Rose as the dependent variable.
- Checking the head and tail rows of the train and test data.

```
First few rows of Training Data
```

TimeStamp	Sparkling	time
1980-01-31	1686	1
1980-02-29	1591	2
1980-03-31	2304	3
1980-04-30	1712	4
1980-05-31	1471	5

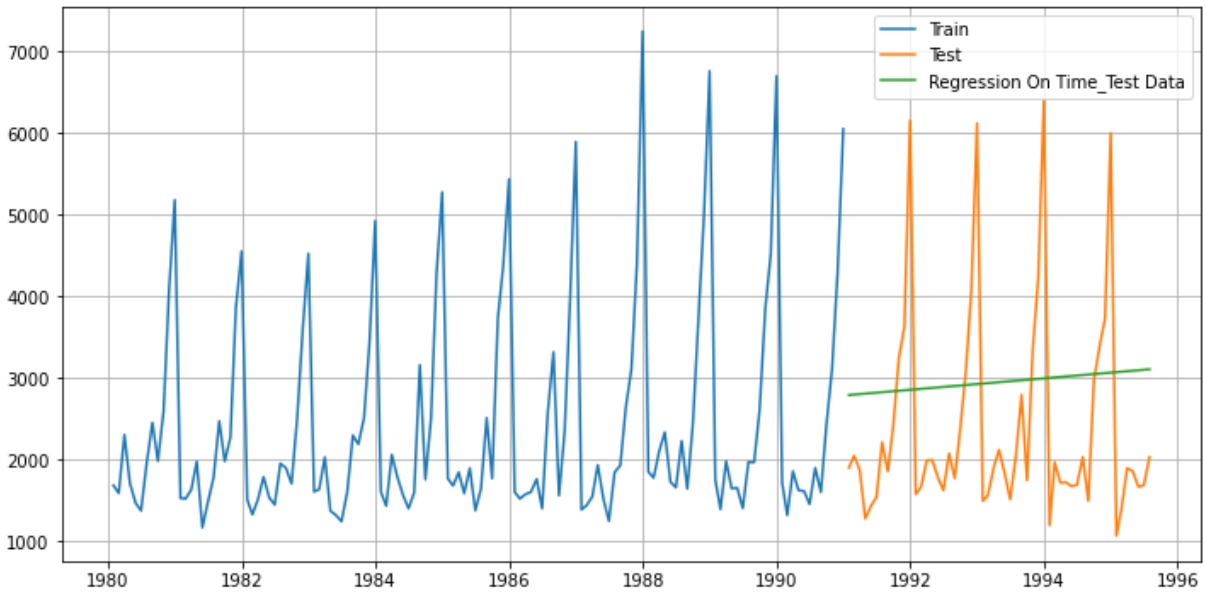
```
First few rows of Test Data
```

TimeStamp	Sparkling	time
1991-01-31	1902	133
1991-02-28	2049	134
1991-03-31	1874	135
1991-04-30	1279	136
1991-05-31	1432	137

- The Linear Regression model is created and fitted on the train data.

```
lr.fit(X, y, sample_weight=None)
```

- The forecast values are visualized in the plot below.



- The RMSE of the test data is calculated using the LR model.

```
Regression on test data:  1389.135174897992
```

NAIVE APPROACH:

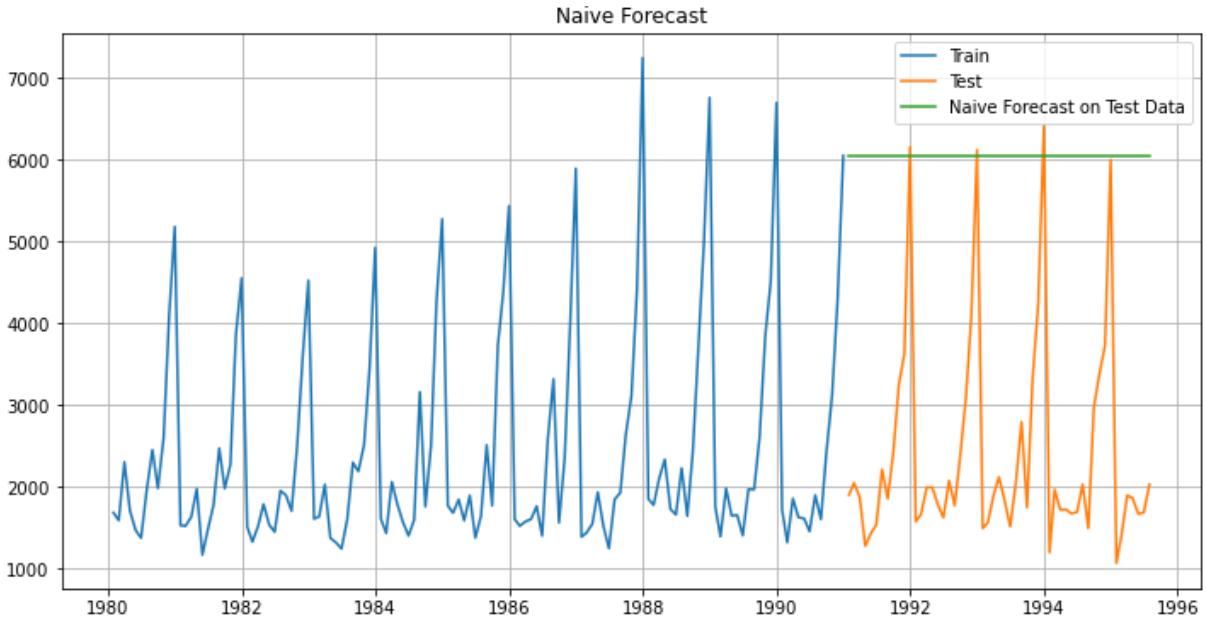
- For this particular naive model, we say that the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.
- Checking the tail of the train data for the Naive model.

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

- The Naive model uses the last value of the train data as the forecast value for the entire test data. We can see the same for the test data as below.

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

- The forecast value with train and test data can be seen in the below plot.



- The RMSE value for the test data can be checked.

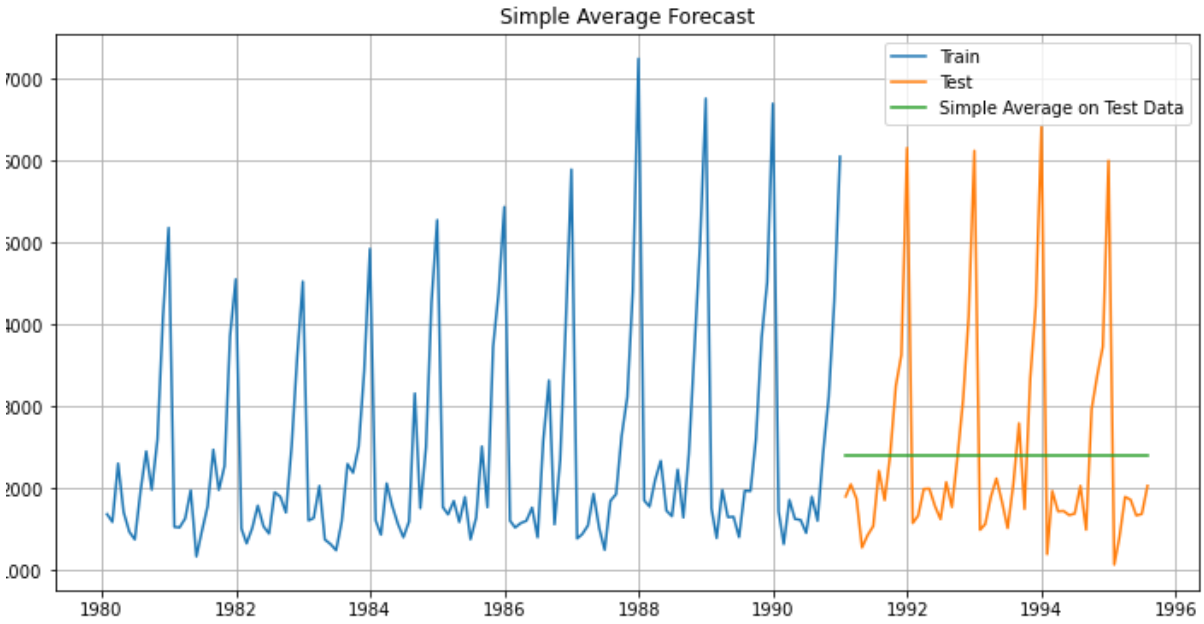
```
For Naive forecast on the Test Data,  RMSE is 3864.279
```

SIMPLE AVERAGE MODEL:

- For this simple average method, we will forecast by using the average of the training values.
- The test is the mean or average of the train data. We can see the test data.

TimeStamp	Sparkling	mean_forecast
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

- We can visualize the forecast of the model along with train and test values.



- The RMSE of the test data is calculated and seen as below.

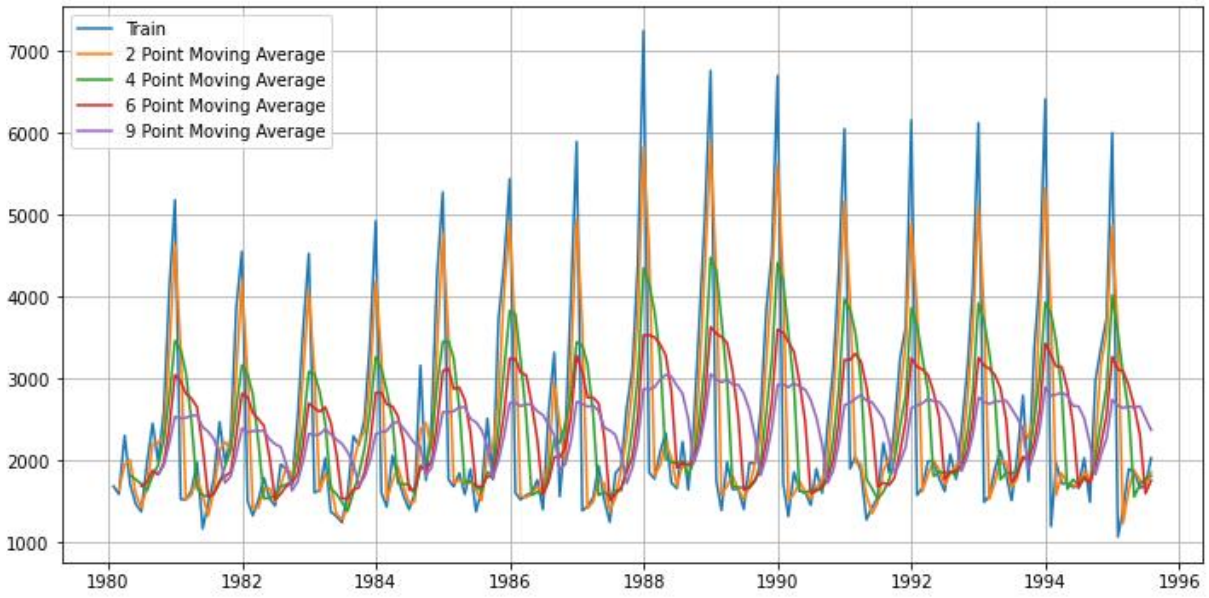
```
For Simple Average forecast on the Test Data,  RMSE is 1275.082
```

MOVING AVERAGE (MA):

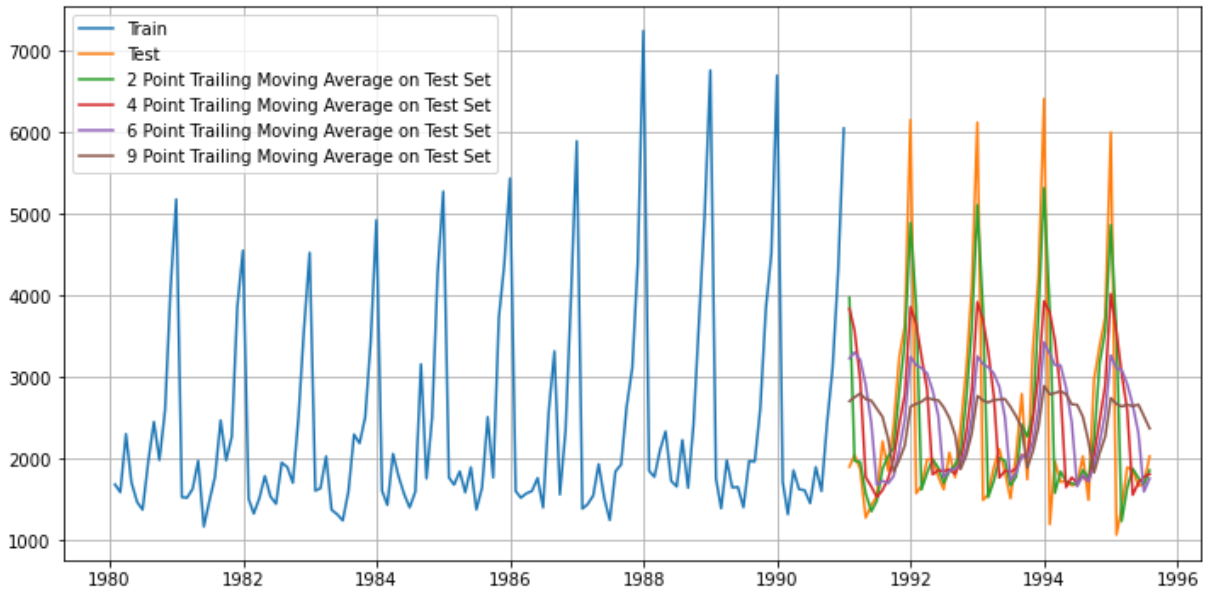
- For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals.
- The best interval can be determined by the maximum accuracy (or the minimum error) over here.
- For Moving Average, we are going to average over the entire data.
- We apply rolling mean of 2,4,6,9 on the data.
- We can see below the original data and rolling mean values.
- NaN is set where there are no values present.

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

- We can visualize the plot below with original data and the rolling means.



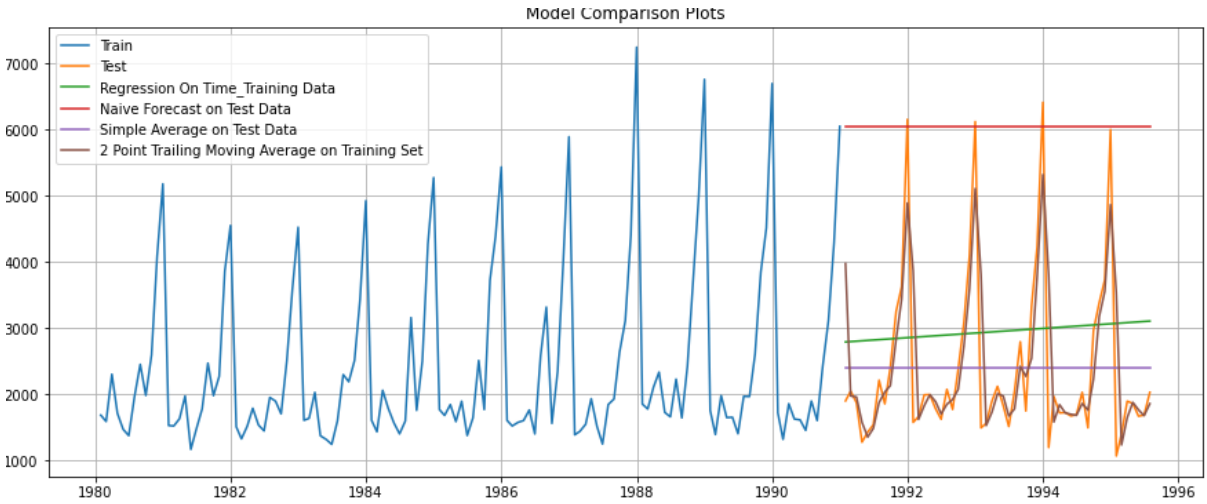
- Splitting the data as train and test data (from 1991 year). We apply the Moving average model on the test data for different trailing points.
- We can visualize the trailing values along with train and test data.



- Checking the RMSE values for the test data on the Moving average model.

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

- We can see that the 2 point Moving Average model forecast has least RMSE value comparatively.
- We plot all the models together and visualize them.



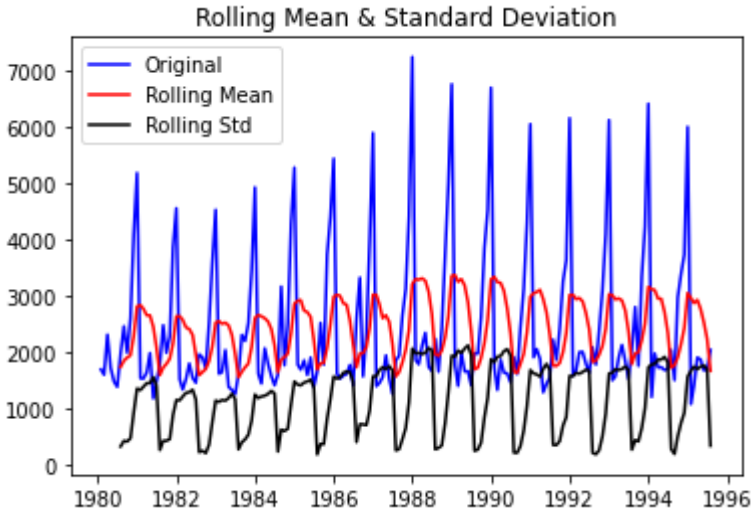
5. **Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment. Note: Stationarity should be checked at alpha = 0.05.**
   - The stationarity of the data is checked using the Augmented Dickey Fuller (ADF) test.
   - The hypothesis for the ADFuller test is as follows:

```
STEP 1: H0 : Time Series is non-stationary
        H1 : Time Series is stationary

STEP 2: Consider the level of significanace (α) as 0.05

STEP 3: Using the Augmented Dickey Fuller (ADF) test we test for stationarity.
```

   - Applying the original data on the ADFuller test and we can see the results below.

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

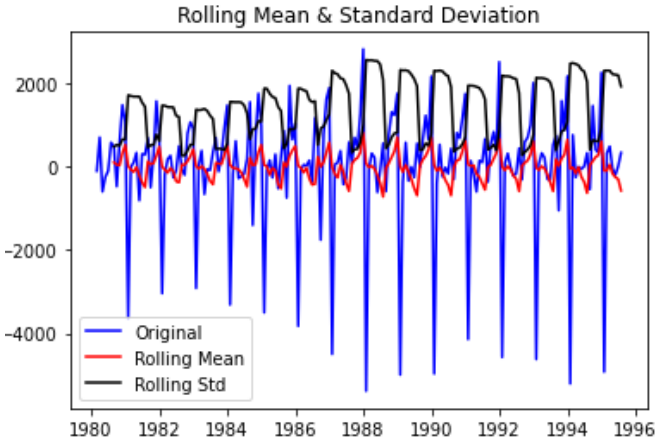
- We can check for the mean and standard deviation of the data for stationarity.



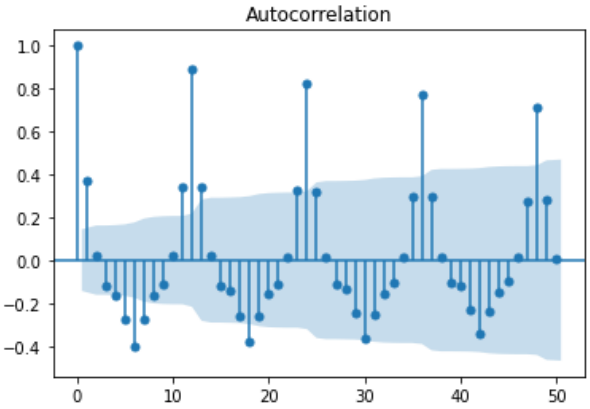
- We can see that the p-value is greater than 0.05 (Level of Significance) hence we cannot reject the Null Hypothesis.
- The given model is non-stationary and we have to make the model as stationary.
- To make a model as stationary we can take appropriate levels of differencing or apply mathematical transformations.
- Here we apply differencing with various levels and initial level with period of 1.
- Applying the differenced data to the ADFuller test and checking the results.

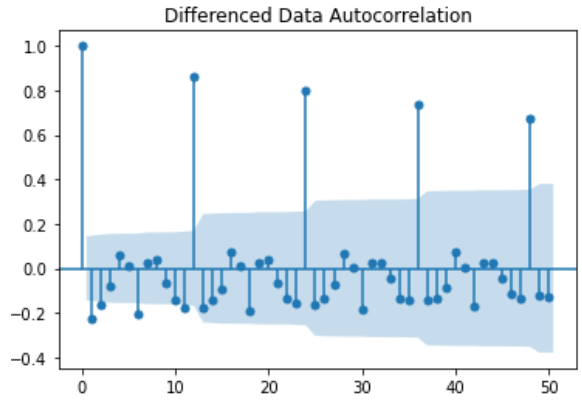
```
Results of Dickey-Fuller Test:
Test Statistic                  -45.050301
p-value                           0.000000
#Lags Used                       10.000000
Number of Observations Used     175.000000
Critical Value (1%)              -3.468280
Critical Value (5%)              -2.878202
Critical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

- We get the p-value less than 0.05, hence rejecting the Null hypothesis and the model stationary now.
- We see that after taking a difference of order 1 the series have become stationary at $\alpha = 0.05$.
- We can plot the mean and standard deviation of the differenced data.

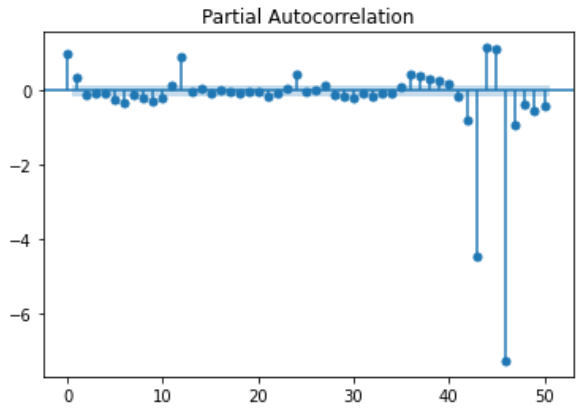


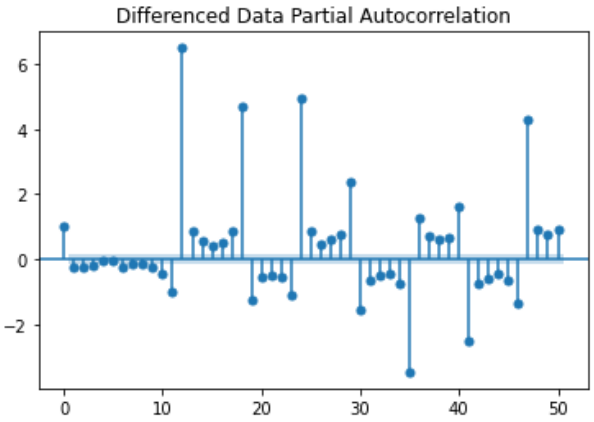
- We built the ACF plots for the original data and the data after differencing.





- We built the PACF plots for the original data and the data after differencing.





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.**

- The data has some seasonality so ideally we should build a SARIMA model. But for demonstration purposes we are building an ARIMA model by looking at the minimum AIC criterion.

AUTOMATED ARIMA MODEL:

- We create a loop to get different combination of parameters of p (for AR) and q (for MA) in the range of 0 and 2, while the order of differencing is kept as 1 for stationarity.

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

- Creating a dataframe for the parameter combinations and their respective AIC values and sort them to get minimum AIC value.

	param	AIC
5	(2, 1, 2)	2210.616569
4	(2, 1, 1)	2232.360490
1	(0, 1, 2)	2232.783098
3	(1, 1, 2)	2233.597647
2	(1, 1, 1)	2235.013945
0	(0, 1, 1)	2264.906438

- Building the automated ARIMA model with the train data and order of (2, 1, 2) where AIC is low for the series.

ARIMA Model Results

Dep. Variable:	D.Sparkling	No. Observations:	131
Model:	ARIMA(2, 1, 2)	Log Likelihood	-1099.308
Method:	css-mle	S.D. of innovations	1011.622
Date:	Sun, 21 Feb 2021	AIC	2210.617
Time:	09:04:32	BIC	2227.868
Sample:	02-29-1980 - 12-31-1990	HQIC	2217.626

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.5854	0.516	10.819	0.000	4.574	6.597
ar.L1.D.Sparkling	1.2704	0.074	17.053	0.000	1.124	1.416
ar.L2.D.Sparkling	-0.5608	0.074	-7.627	0.000	-0.705	-0.417
ma.L1.D.Sparkling	-1.9999	0.042	-47.059	0.000	-2.083	-1.917
ma.L2.D.Sparkling	0.9999	0.042	23.564	0.000	0.917	1.083

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.1326	-0.7073j	1.3354	-0.0888
AR.2	1.1326	+0.7073j	1.3354	0.0888
MA.1	1.0000	-0.0005j	1.0000	-0.0001
MA.2	1.0000	+0.0005j	1.0000	0.0001

- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the ARIMA (2, 1, 2) model.

ARIMA(2,1,2)	1375.191113

AUTOMATED SARIMA MODEL:

- We see that there can be a seasonality of 6 and 12 from the ACF plot. We will run our auto SARIMA models by setting seasonality as 6 and 12.

Seasonality as 6 of the Auto SARIMA Model:

- We create a loop to get different combination of parameters of p (for AR) and q (for MA) in the range of 0 and 2, while the order of differencing d is kept as 1 for stationarity and the order of differencing D is kept as 0 for the seasonal stationarity.

```
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)
```

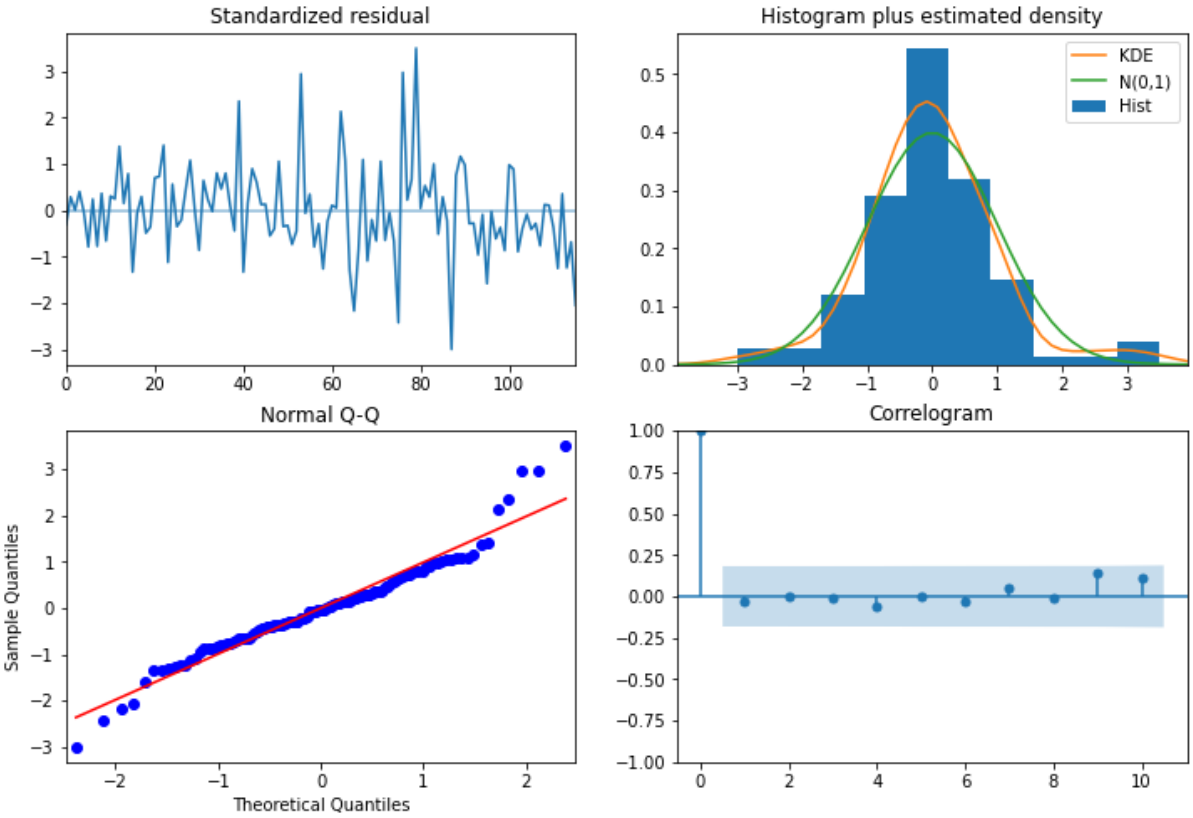
- Running the SARIMA model with all the possible parameter combinations and select the minimum AIC value.

	param	seasonal	AIC
53	(1, 1, 2)	(2, 0, 2, 6)	1727.510411
26	(0, 1, 2)	(2, 0, 2, 6)	1727.888803
80	(2, 1, 2)	(2, 0, 2, 6)	1729.335504
17	(0, 1, 1)	(2, 0, 2, 6)	1741.641479
44	(1, 1, 1)	(2, 0, 2, 6)	1743.374727

- Taking the order (1, 1, 2) (2, 0, 2, 6) as the parameter and building the automated SARIMA model for the series on the train data.

```
                                Statespace Model Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  132
Model:             SARIMAX(1, 1, 2)x(2, 0, 2, 6)   Log Likelihood                -855.755
Date:                           Sun, 21 Feb 2021   AIC                           1727.510
Time:                                   09:06:36   BIC                           1749.539
Sample:                                        0   HQIC                          1736.453
                                             - 132
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6457      0.287     -2.252      0.024      -1.208      -0.084
ma.L1         -0.1062      0.251     -0.423      0.672      -0.598       0.386
ma.L2         -0.7010      0.202     -3.462      0.001      -1.098      -0.304
ar.S.L6       -0.0048      0.027     -0.177      0.860      -0.058       0.048
ar.S.L12       1.0362      0.018     56.089      0.000       1.000       1.072
ma.S.L6        0.4752      0.143      3.321      0.001       0.195       0.756
ma.S.L12      -0.9176      0.180     -5.097      0.000      -1.270      -0.565
sigma2      9.655e+04    2.3e+04      4.193      0.000    5.14e+04    1.42e+05
===================================================================================
Ljung-Box (Q):                       28.98   Jarque-Bera (JB):                 25.23
Prob(Q):                              0.90   Prob(JB):                          0.00
Heteroskedasticity (H):               2.65   Skew:                              0.46
Prob(H) (two-sided):                  0.00   Kurtosis:                          5.09
===================================================================================

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

- Visualizing the Diagnostic plots for standardized residuals of one endogenous variable.



- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the SARIMA (1, 1, 2) (2, 0, 2, 6) model.

SARIMA(1,1,2)(2,0,2,6)	629.354379

Seasonality as 12 of the Auto SARIMA Model:

- We create a loop to get different combination of parameters of p (for AR) and q (for MA) in the range of 0 and 2, while the order of differencing d is kept as 1 for stationarity and the order of differencing D is kept as 0 for the seasonal stationarity.

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (1, 1, 0)(1, 0, 0, 12)
Model: (1, 1, 1)(1, 0, 1, 12)
Model: (1, 1, 2)(1, 0, 2, 12)
Model: (2, 1, 0)(2, 0, 0, 12)
Model: (2, 1, 1)(2, 0, 1, 12)
Model: (2, 1, 2)(2, 0, 2, 12)
```

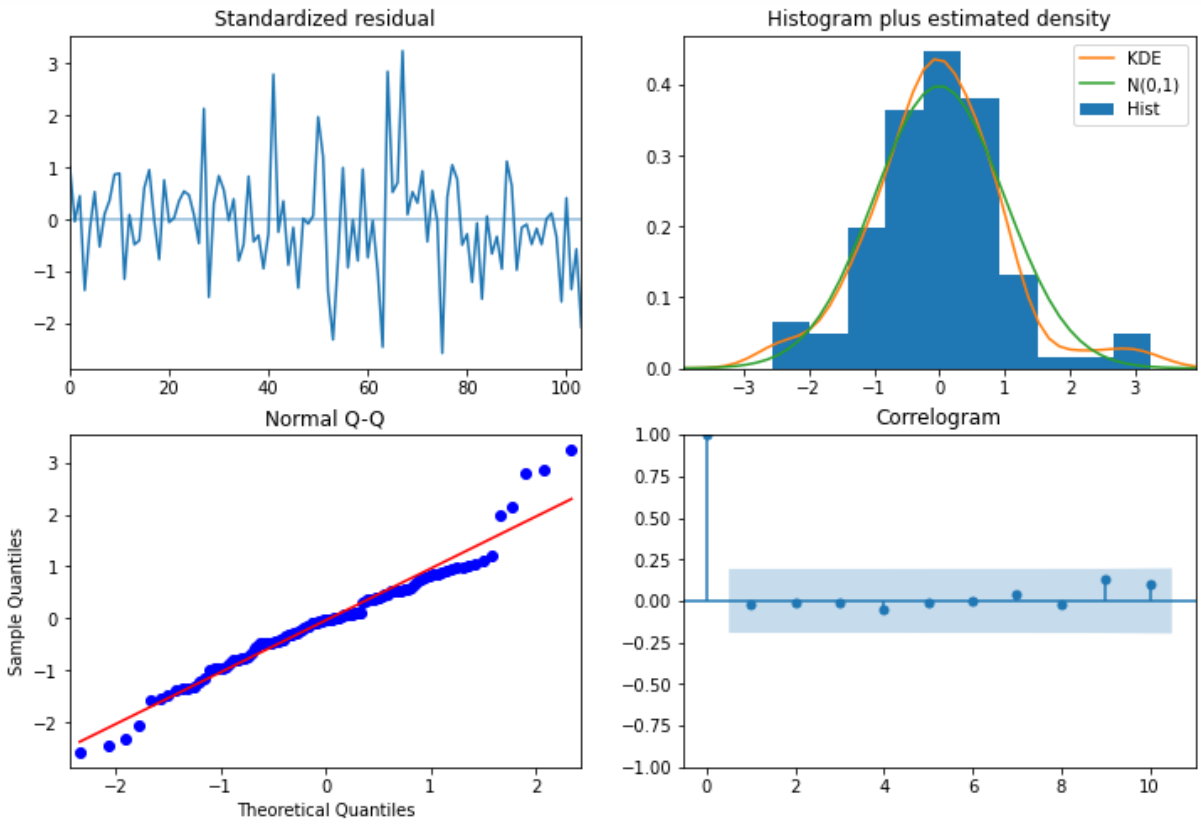
- Running the SARIMA model with all the possible parameter combinations and select the minimum AIC value.

	param	seasonal	AIC
50	(1, 1, 2)	(1, 0, 2, 12)	1555.584247
53	(1, 1, 2)	(2, 0, 2, 12)	1556.080255
26	(0, 1, 2)	(2, 0, 2, 12)	1557.121564
23	(0, 1, 2)	(1, 0, 2, 12)	1557.160507
77	(2, 1, 2)	(1, 0, 2, 12)	1557.340405

- Taking the order (1, 1, 2) (1, 0, 2, 12) as the parameter and building the automated SARIMA model for the series on the train data.

```
                                 Statespace Model Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  132
Model:             SARIMAX(1, 1, 2)x(1, 0, 2, 12)   Log Likelihood                -770.792
Date:                            Sun, 21 Feb 2021   AIC                           1555.584
Time:                                    09:10:13   BIC                           1574.095
Sample:                                         0   HQIC                          1563.083
                                              - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6282      0.255     -2.464      0.014      -1.128      -0.128
ma.L1         -0.1040      0.225     -0.463      0.643      -0.545       0.337
ma.L2         -0.7277      0.154     -4.736      0.000      -1.029      -0.427
ar.S.L12       1.0439      0.014     72.838      0.000       1.016       1.072
ma.S.L12      -0.5550      0.098     -5.663      0.000      -0.747      -0.363
ma.S.L24      -0.1354      0.120     -1.133      0.257      -0.370       0.099
sigma2      1.506e+05   2.03e+04      7.401      0.000    1.11e+05    1.9e+05
===================================================================================
Ljung-Box (Q):                       23.02   Jarque-Bera (JB):                 11.72
Prob(Q):                              0.99   Prob(JB):                          0.00
Heteroskedasticity (H):               1.47   Skew:                              0.36
Prob(H) (two-sided):                  0.26   Kurtosis:                          4.48
===================================================================================

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

- Visualizing the Diagnostic plots for standardized residuals of one endogenous variable.



- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the SARIMA (1, 1, 2) (1, 0, 2, 12) model.

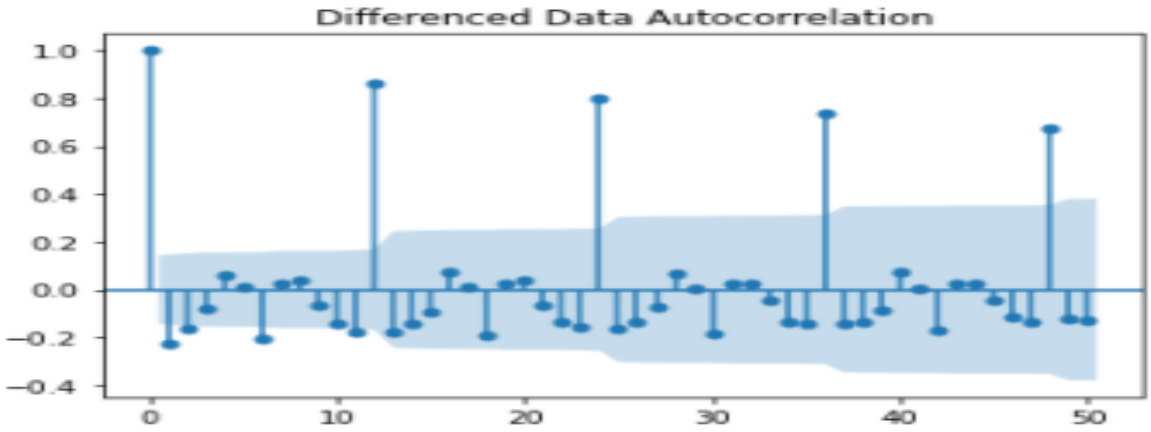
SARIMA(1,1,2)(1,0,2,12)	528.602775

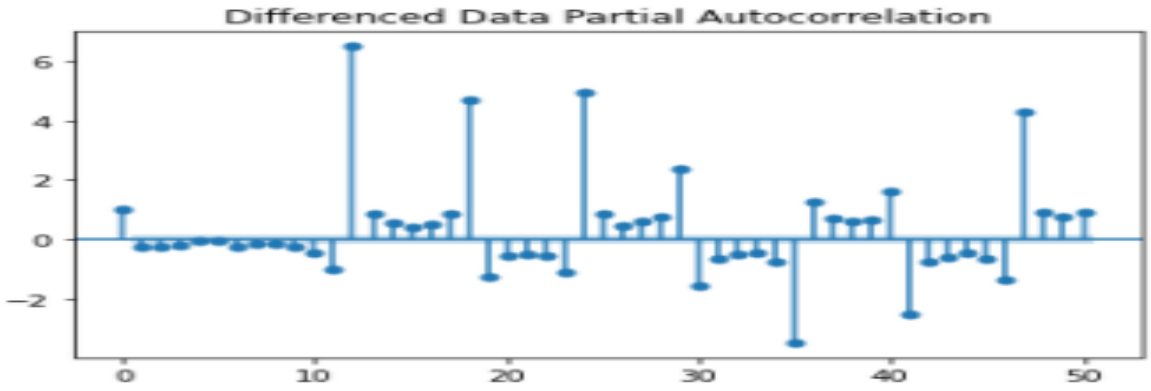
7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.

- The data has some seasonality so ideally we should build a SARIMA model. But for demonstration purposes we are building an ARIMA model by looking at the ACF and the PACF plots.

ARIMA USING CUT-OFF POINTS FROM ACF AND PACF PLOTS:

- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 0.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 0.
- Checking the ACF and PACF plots for the data differencing of level 1.





- By looking at the above plots, we can say that both the PACF has 3 AR terms (p) and ACF plot has 2 MA terms(q).
- Taking the summary of ARIMA (4,1,2) as seen below.

ARIMA Model Results

Dep. Variable:	D.Sparkling	No. Observations:	131
Model:	ARIMA(3, 1, 2)	Log Likelihood	-1107.464
Method:	css-mle	S.D. of innovations	1105.999
Date:	Tue, 23 Feb 2021	AIC	2228.927
Time:	15:39:15	BIC	2249.054
Sample:	02-29-1980 - 12-31-1990	HQIC	2237.105

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.9812	nan	nan	nan	nan	nan
ar.L1.D.Sparkling	-0.4419	nan	nan	nan	nan	nan
ar.L2.D.Sparkling	0.3080	7.68e-06	4.01e+04	0.000	0.308	0.308
ar.L3.D.Sparkling	-0.2501	nan	nan	nan	nan	nan
ma.L1.D.Sparkling	-0.0004	0.031	-0.011	0.991	-0.060	0.060
ma.L2.D.Sparkling	-0.9996	0.031	-32.716	0.000	-1.060	-0.940

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.0000	-0.0000j	1.0000	-0.5000
AR.2	1.1157	-1.6594j	1.9996	-0.1558
AR.3	1.1157	+1.6594j	1.9996	0.1558
MA.1	1.0000	+0.0000j	1.0000	0.0000
MA.2	-1.0004	+0.0000j	1.0004	0.5000

- Hence we select the model of order (3, 1, 2) at be the best model for ARIMA using the cut-off points from the ACF and PACF plots.
- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the ARIMA (3, 1, 2) model.

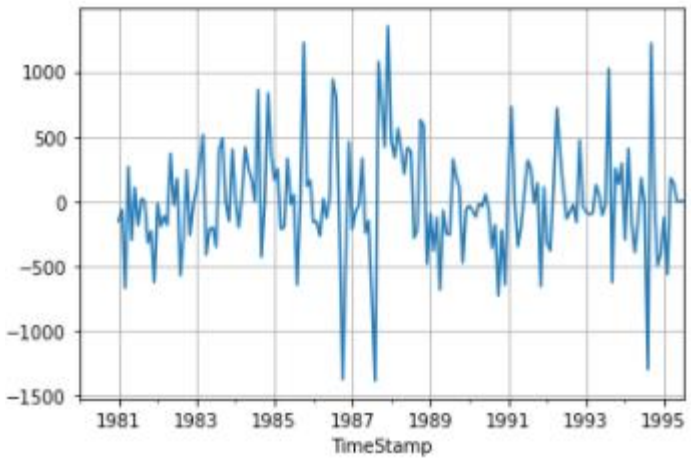
ARIMA(3,1,2)	1378.786317

SARIMA USING CUT-OFF POINTS FROM ACF AND PACF PLOTS:

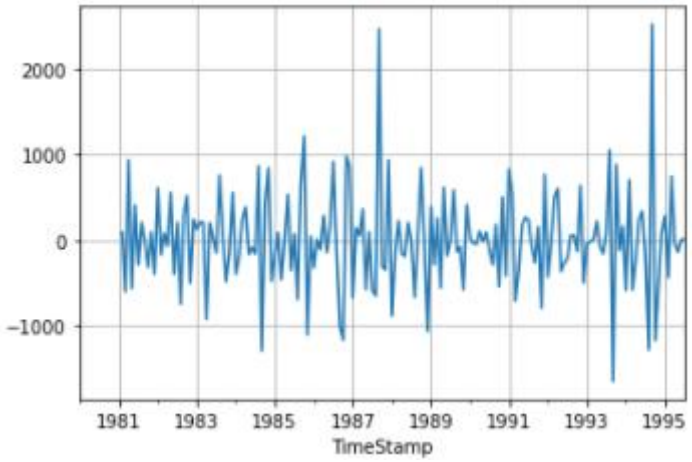
- We see that there can be a seasonality of 6 and 12 from the ACF plot. We will run our auto SARIMA models by setting seasonality as 6 and 12.

Seasonality as 12 of the SARIMA Model:

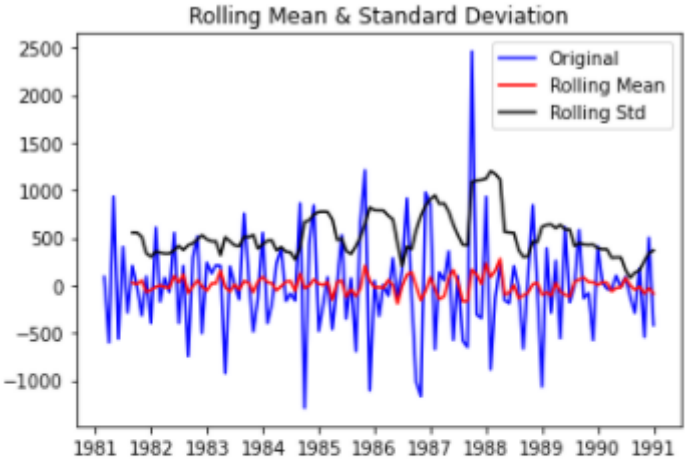
- Taking the seasonality as 12 and checking the plot of the data.



- We see that there might be a slight trend which can be noticed in the data. So we take a differencing of first order on the seasonally differenced series.

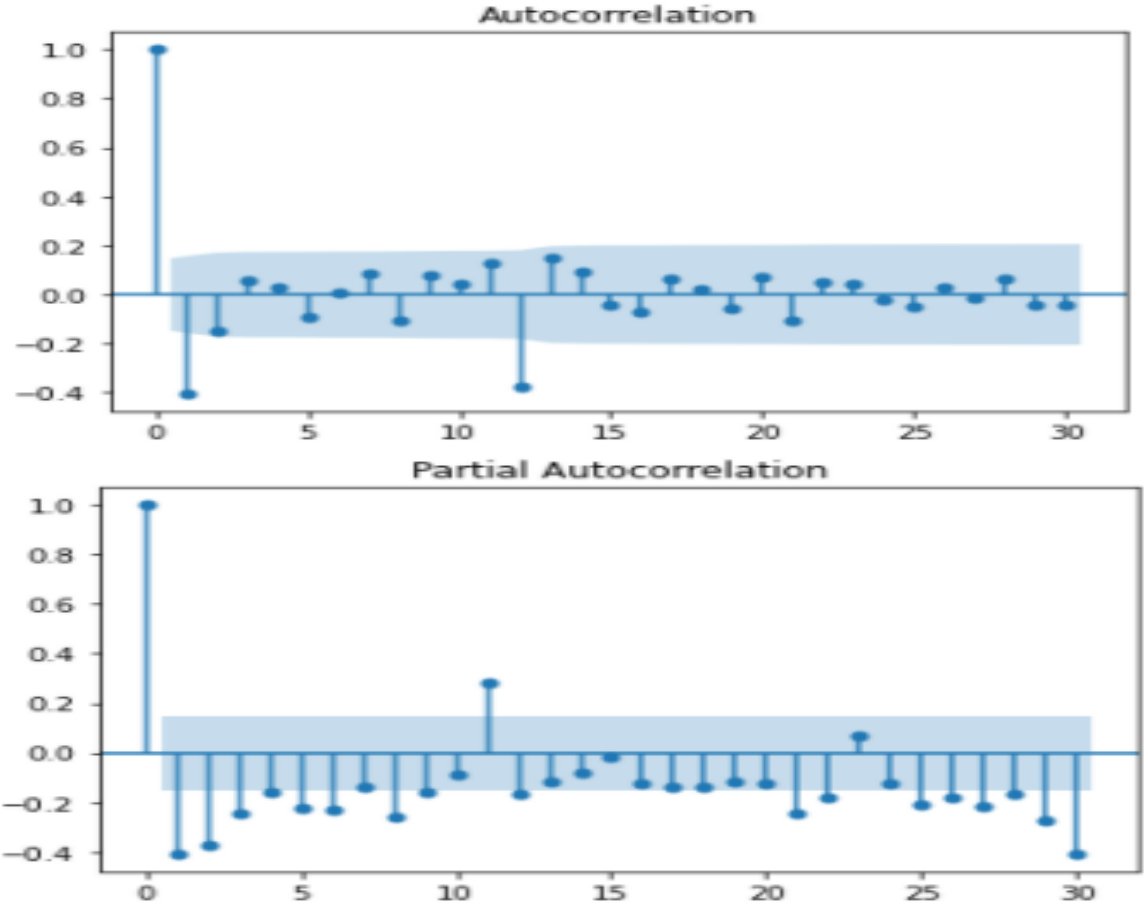


- Checking the stationarity of the seasonally first order differencing series.




```
Results of Dickey-Fuller Test:
Test Statistic                  -3.342905
p-value                          0.013066
#Lags Used                      10.000000
Number of Observations Used    108.000000
Critical Value (1%)             -3.492401
Critical Value (5%)             -2.888697
Critical Value (10%)            -2.581255
dtype: float64
```

- The ACF and PACF plots for the seasonality is plotted and checked for the order.



- The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 0.
- The Moving-Average parameter in an SARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off to 0.
- Remember to check the ACF and the PACF plots only at multiples of 12 (since 12 is the seasonal period).
- The below is summary of the SARIMA with order (3, 1, 2) (6, 1, 2, 12) with lowest AIC possible.

Statespace Model Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)x(6, 1, 1, 12)	Log Likelihood	-323.675
Date:	Tue, 23 Feb 2021	AIC	673.349
Time:	15:41:37	BIC	696.543
Sample:	0	HQIC	681.951
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5243	0.239	-2.190	0.029	-0.993	-0.055
ar.L2	0.3109	0.423	0.735	0.462	-0.518	1.140
ar.L3	0.3247	0.224	1.449	0.147	-0.115	0.764
ma.L1	-3.888e-05	349.378	-1.11e-07	1.000	-684.768	684.768
ma.L2	-1.0000	86.665	-0.012	0.991	-170.860	168.860
ar.S.L12	-0.8795	0.206	-4.277	0.000	-1.282	-0.476
ar.S.L24	-0.3472	0.217	-1.598	0.110	-0.773	0.079
ar.S.L36	-0.1852	0.170	-1.087	0.277	-0.519	0.149
ar.S.L48	-0.2829	0.257	-1.102	0.270	-0.786	0.220
ar.S.L60	-0.5926	0.336	-1.764	0.078	-1.251	0.066
ar.S.L72	-0.2040	0.275	-0.743	0.458	-0.742	0.334
ma.S.L12	0.9980	86.945	0.011	0.991	-169.411	171.407
sigma2	1.046e+05	0.001	7.93e+07	0.000	1.05e+05	1.05e+05

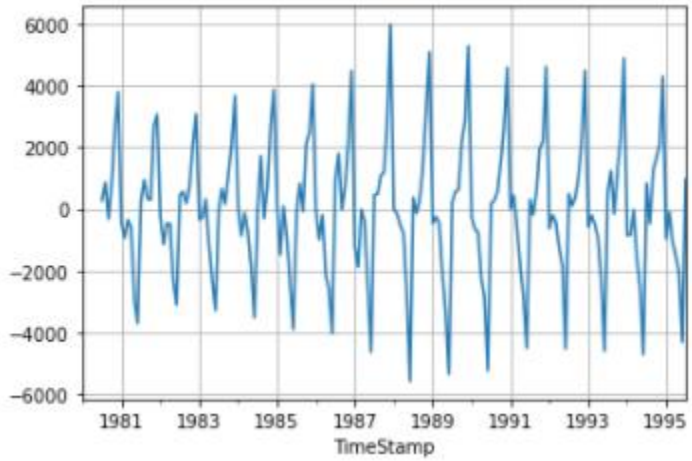
Ljung-Box (Q):	27.42	Jarque-Bera (JB):	3.11
Prob(Q):	0.93	Prob(JB):	0.21
Heteroskedasticity (H):	0.34	Skew:	0.49
Prob(H) (two-sided):	0.04	Kurtosis:	3.85

- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the SARIMA (3, 1, 2) (6, 1, 2, 12) model.

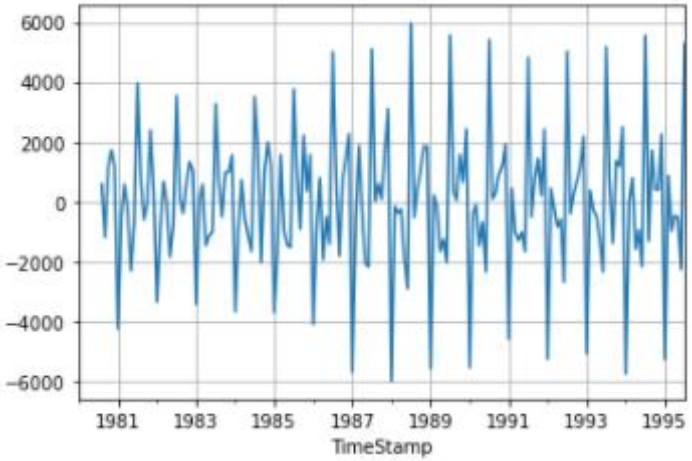
SARIMA(0,1,2)(0,0,0,12)	369.677903

Seasonality as 6 of Auto SARIMA Model:

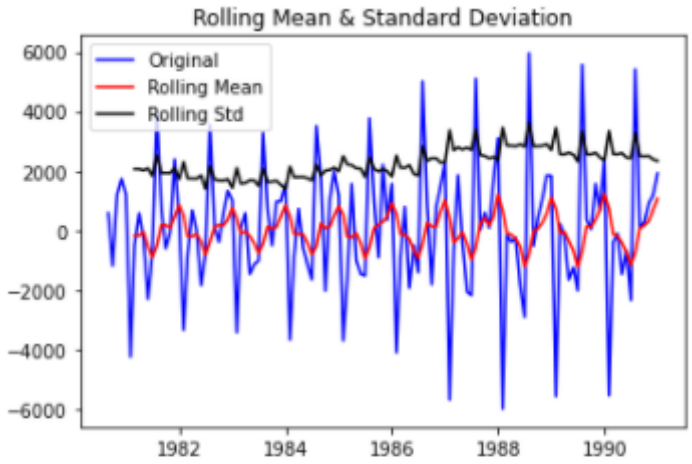
- Taking the seasonality as 6 and checking the plot of the data.



- We see that there might be a slight trend which can be noticed in the data. So we take a differencing of first order on the seasonally differenced series.

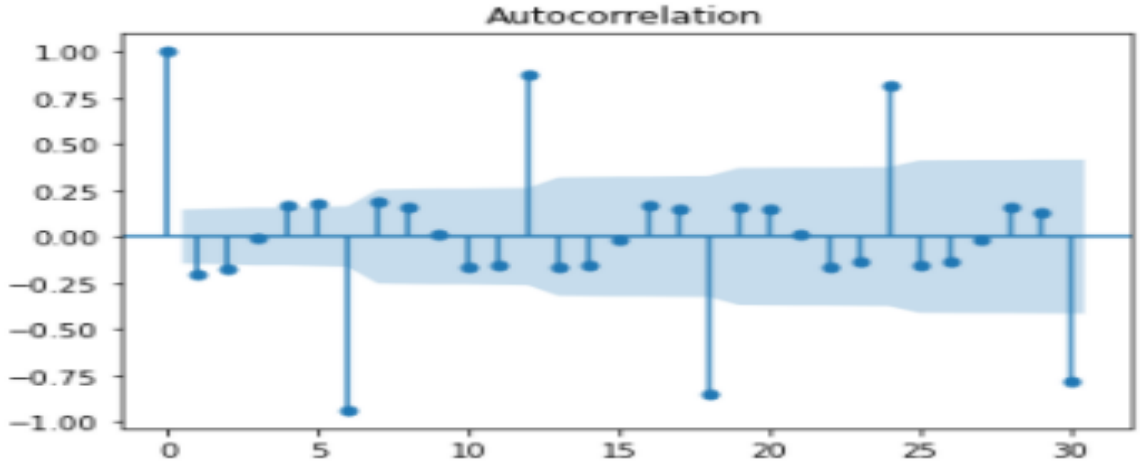


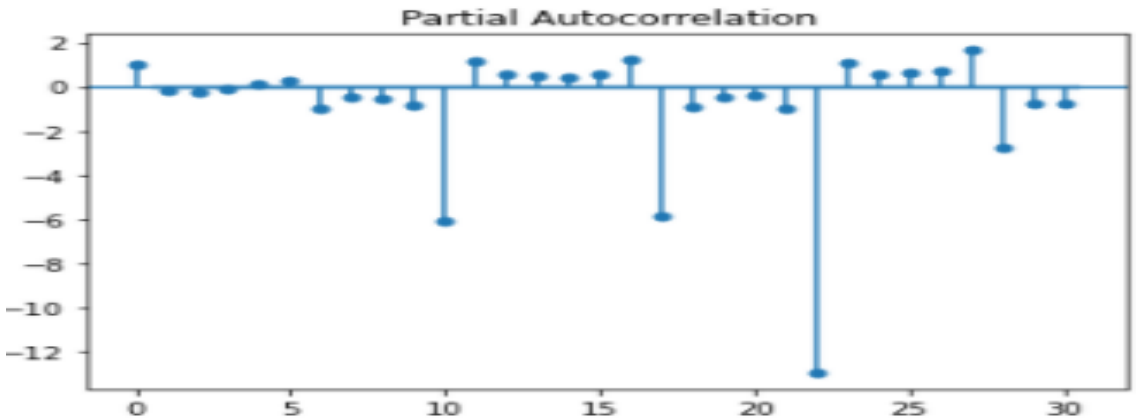
- Checking the stationarity of the seasonally first order differencing series.




```
Results of Dickey-Fuller Test:
Test Statistic                -7.017242e+00
p-value                        6.683657e-10
#Lags Used                     1.300000e+01
Number of Observations Used    1.110000e+02
Critical Value (1%)           -3.490683e+00
Critical Value (5%)           -2.887952e+00
Critical Value (10%)          -2.580857e+00
dtype: float64
```

- The ACF and PACF plots for the seasonality is plotted and checked for the order.





- The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 0.
- The Moving-Average parameter in an SARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off to 0.
- Remember to check the ACF and the PACF plots only at multiples of 6 (since 6 is the seasonal period).
- The below is summary of the SARIMA with order (3, 1, 2) (6, 1, 2, 6) with lowest AIC possible.

Statespace Model Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)x(2, 1, 2, 6)	Log Likelihood	-812.727
Date:	Tue, 23 Feb 2021	AIC	1645.454
Time:	15:39:30	BIC	1672.458
Sample:	0	HQIC	1656.407
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6201	0.257	-2.413	0.016	-1.124	-0.116
ar.L2	0.0281	0.165	0.171	0.864	-0.294	0.351
ar.L3	0.0845	0.109	0.778	0.437	-0.128	0.298
ma.L1	-0.1429	0.241	-0.593	0.553	-0.616	0.330
ma.L2	-0.7227	0.217	-3.325	0.001	-1.149	-0.297
ar.S.L6	-1.1541	0.206	-5.611	0.000	-1.557	-0.751
ar.S.L12	-0.1371	0.210	-0.652	0.514	-0.549	0.275
ma.S.L6	0.2654	0.220	1.205	0.228	-0.166	0.697
ma.S.L12	-0.5578	0.106	-5.271	0.000	-0.765	-0.350
sigma2	1.501e+05	2.3e+04	6.516	0.000	1.05e+05	1.95e+05

Ljung-Box (Q):	26.89	Jarque-Bera (JB):	13.34
Prob(Q):	0.94	Prob(JB):	0.00
Heteroskedasticity (H):	1.66	Skew:	0.48
Prob(H) (two-sided):	0.13	Kurtosis:	4.41

- We forecast on the duration of the test data.
- Checking the RMSE for the test data using the SARIMA (3, 1, 2) (6, 1, 2, 6) model.

SARIMA(3,1,2)(2,1,2,6)	596.465833

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

In Sparkling the sorted order of Test RMSE values:

	Test RMSE
Alpha=0.086,Beta=3.896,Gamma=0.476:TES	362.795026
Alpha=0.083,Beta=1.967,Gamma=0.491:TES	366.948774
SARIMA(0,1,2)(0,0,0,12)	369.677903
Alpha=0.153,Beta=3.405,Gamma=0.369:TES	392.957153
SARIMA(1,1,2)(1,0,2,12)	528.602775
SARIMA(3,1,2)(2,1,2,6)	596.465833
SARIMA(1,1,2)(2,0,2,6)	629.354379
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.098,SES	1275.081739
SimpleAverageModel	1275.081804
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
ARIMA(2,1,2)	1375.191113
ARIMA(3,1,2)	1378.786317
RegressionOnTime	1389.135175
Alpha=0.073,Beta=0.0:DES	2135.056676
Alpha=0.647,Beta=0.0:DES	3851.129439
NaiveModel	3864.279352

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.**

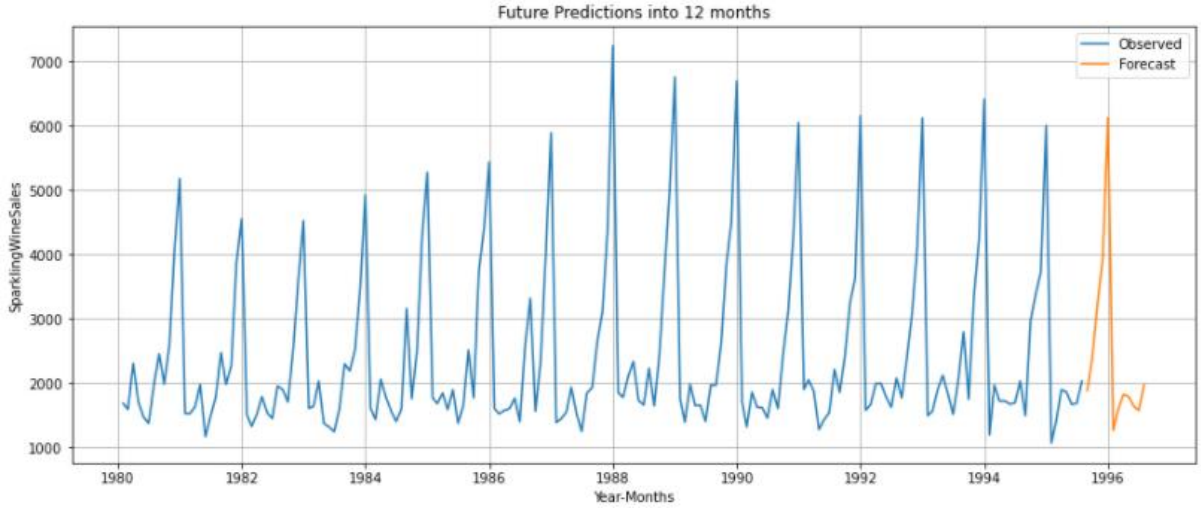
- The most optimum model on the complete data is Triple Exponential Smoothing or Holt - Winter's Method as seen from the above table in the sorted order from the least RMSE value.
- Checking on the TES parameters.

```
{'smoothing_level': 0.05263157894736842,
 'smoothing_slope': 0.05263157894736842,
 'smoothing_seasonal': 0.3684210526315789,
 'damping_slope': nan,
 'initial_level': 1580.0,
 'initial_slope': 0.0,
 'initial_seasons': array([ 106.,   11.,  724.,  132., -109., -203.,  386.,  873.,  404.,
        1016., 2507., 3599.]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

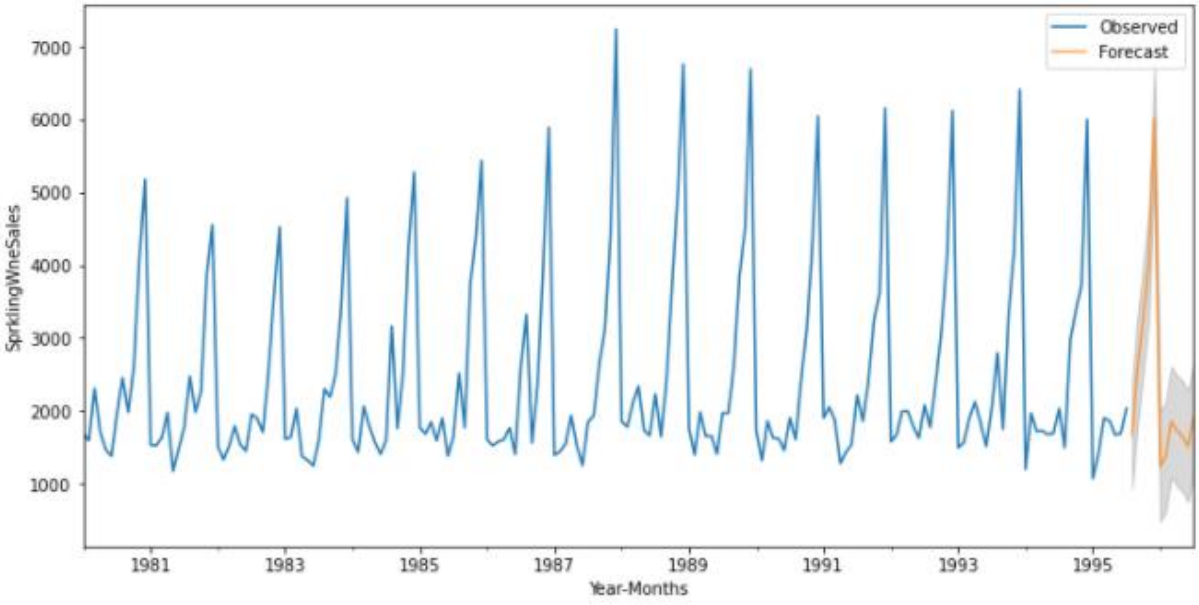
- Using the TES model, forecasting for the duration of 12 months into the future.
- The predicted values for 12 months into the future are as follows.

```
1995-08-31    1884.976788
1995-09-30    2402.258515
1995-10-31    3245.977251
1995-11-30    3932.213221
1995-12-31    6119.724097
1996-01-31    1266.116928
1996-02-29    1583.646654
1996-03-31    1821.829064
1996-04-30    1795.729443
1996-05-31    1643.054827
1996-06-30    1576.941994
1996-07-31    1975.093858
Freq: M, dtype: float64
```

- The plot with the observed data and forecasted data is seen below.

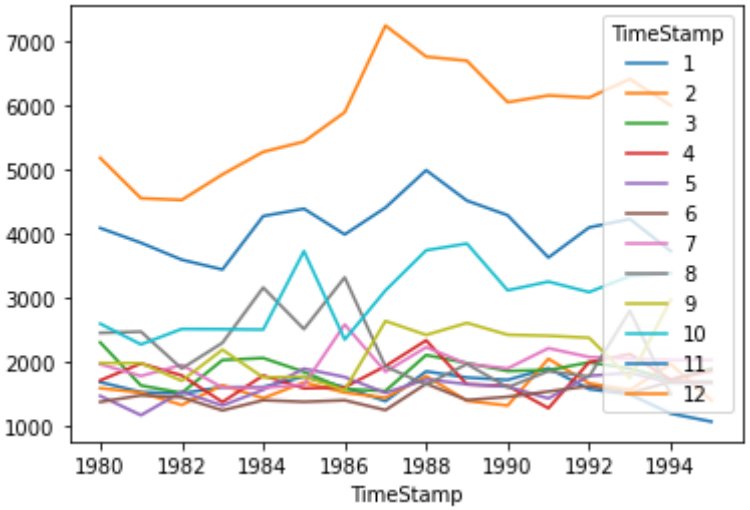


- Using the SARIMA model, we get the below forecast with confidence interval.

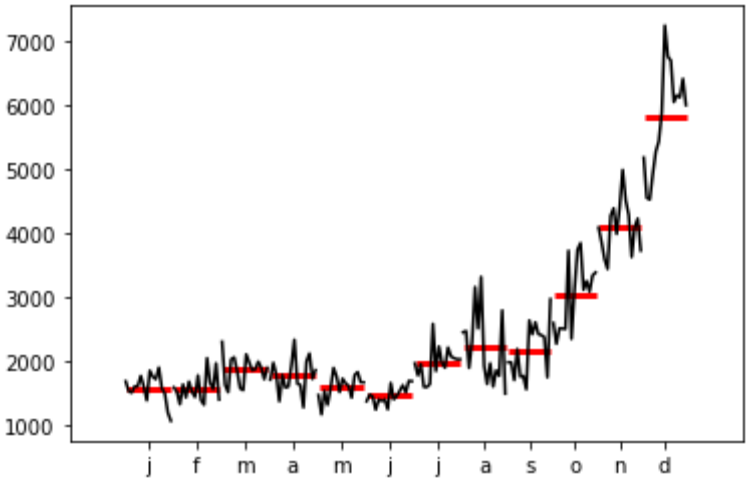


10. Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

- The series seems to be Multiplicative while decomposition of the series.



- From the above monthly plot across the years, we can see that the sales are high in the month of December. This might be due to the festive season of the year.
- The sales across the years seem doing pretty good but are slightly fluctuating.
- The sale is low in the month of January and gradually increases till July. There is good raise in till December. Overall sale is high in the December across the years. We can see in the below plot.



- In the forecast for the future, the sale from previous 4 years is considered. The similar pattern is being followed, January has low sale and December has high sale.
- The company must try to retain the same practises as now to maintain the sales in the near future.
- The company can take new steps as well as risk by constantly improving with the customer's preference to increase the sales effectively.