Read the data as an appropriate Time Series data and plot the data (Rose)

Out[4]:

	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

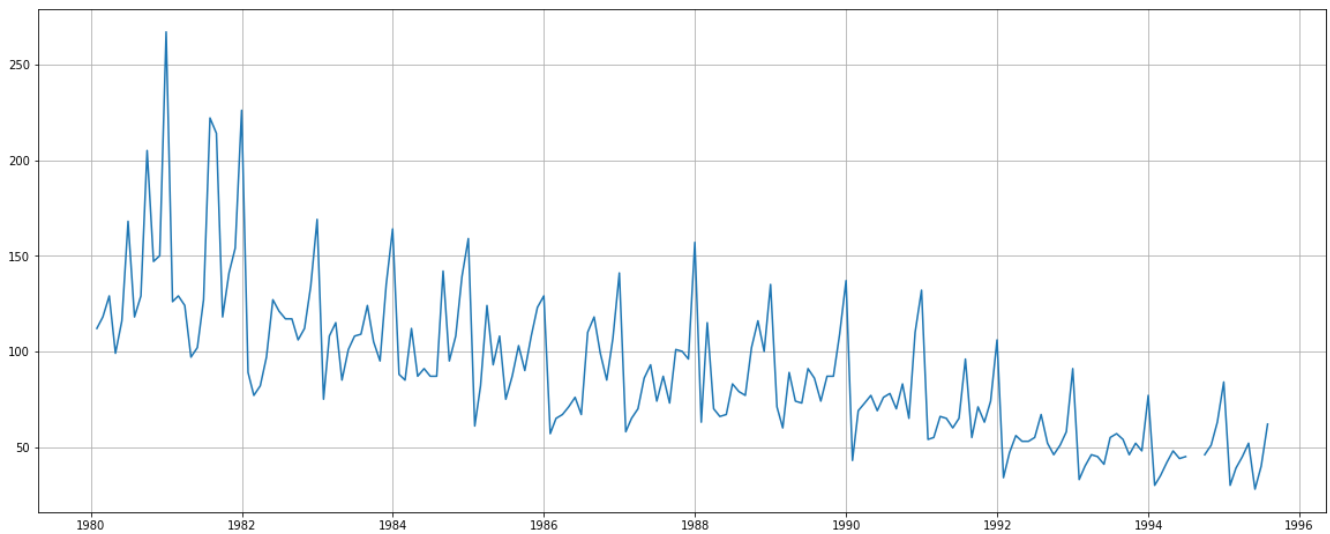
Out[6]:

```
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')
```

Out[7]:

	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

Plot the data(Rose)



Read the data as an appropriate Time Series data and plot the data (Sparkling)

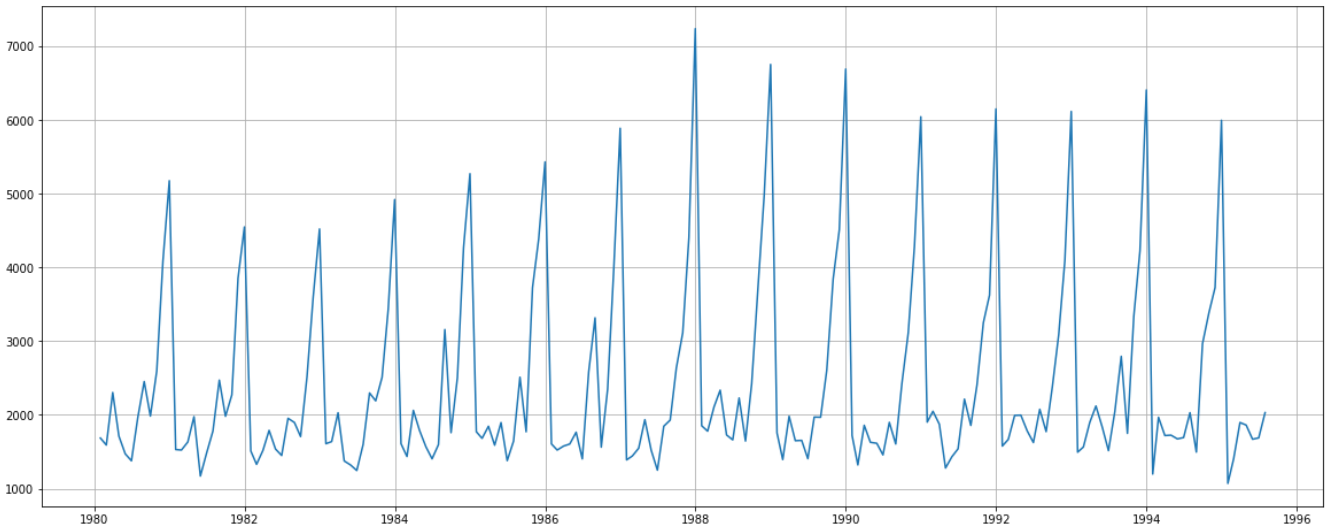
Out[10]:

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

Out[12]:

	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

Plot the data(Sparkling)



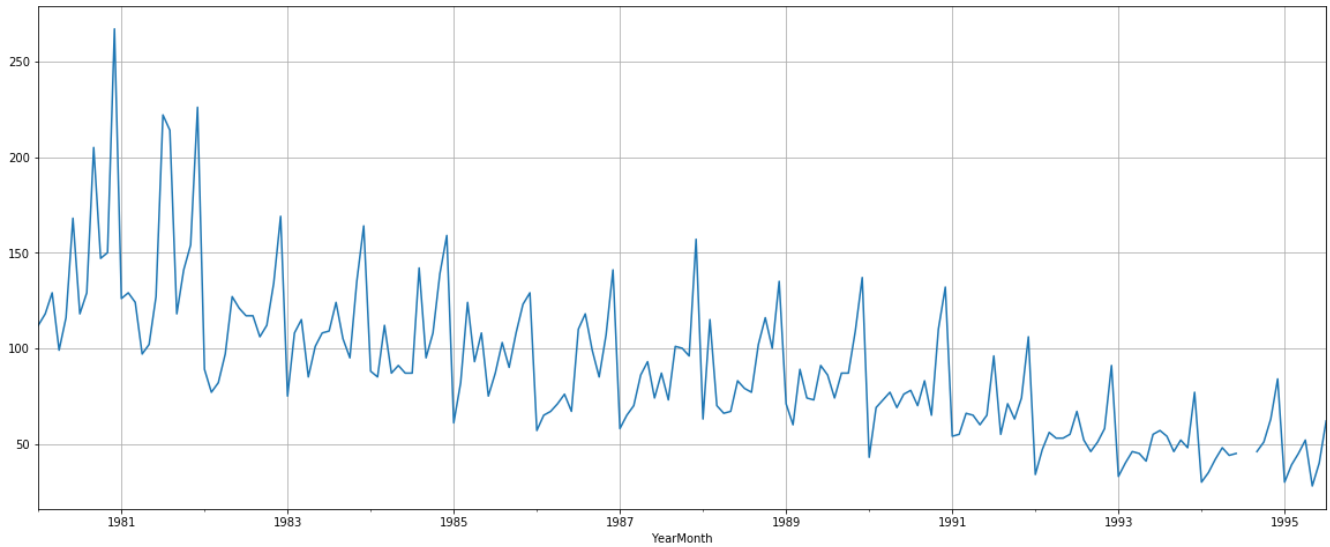
2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

Out[16]:

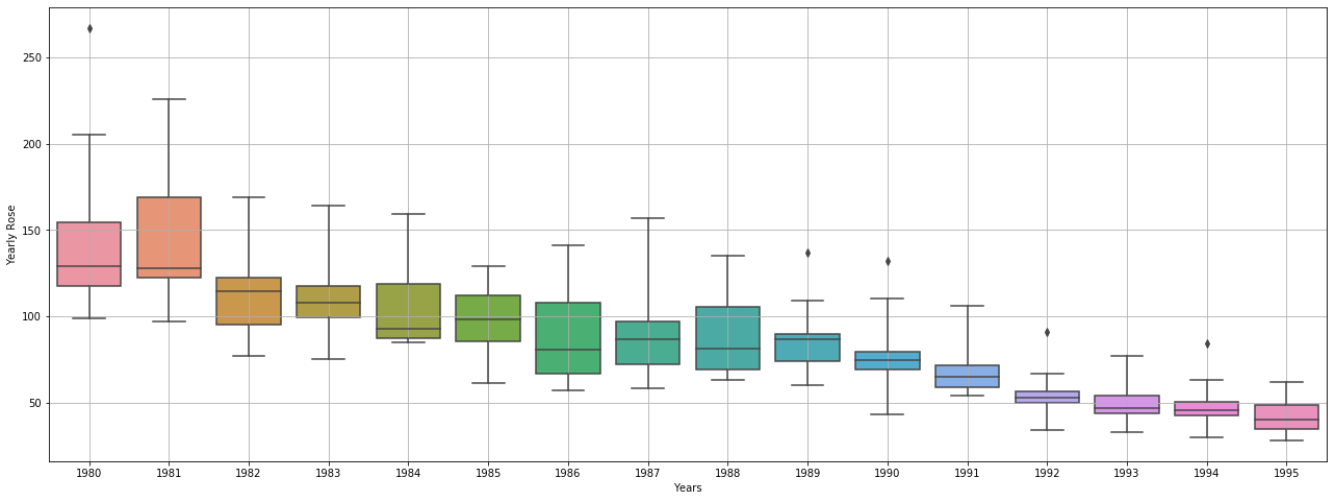
```
YearMonth
1980-01-01    112.0
1980-02-01    118.0
1980-03-01    129.0
1980-04-01     99.0
1980-05-01    116.0
Name: Rose, dtype: float64
```

Out[17]:
```
count    185.000000
mean      90.394595
std       39.175344
min       28.000000
25%       63.000000
50%       86.000000
75%      112.000000
max      267.000000
Name: Rose, dtype: float64
```

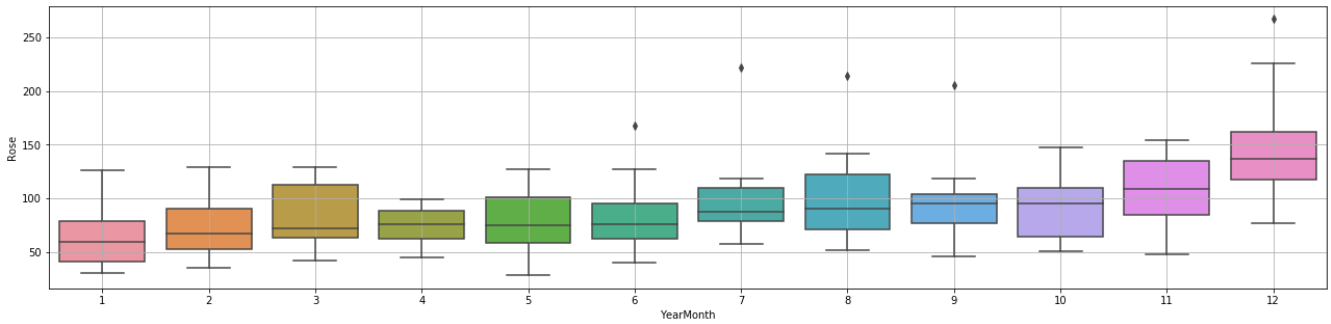
Plot the data(Rose)



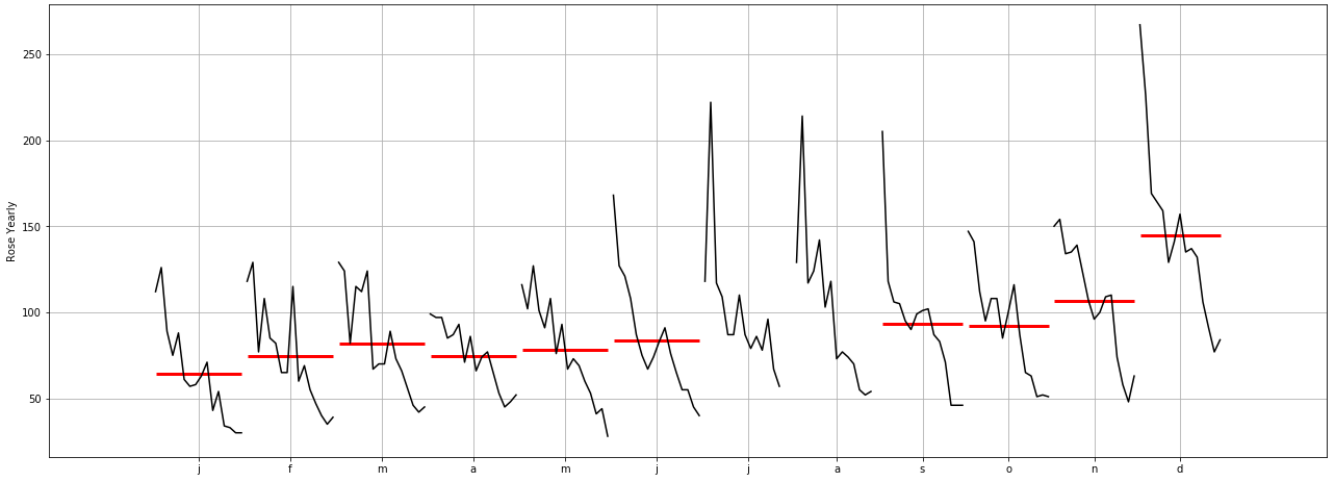
Plot a year on year boxplot(Rose)



Plot a monthly boxplot(Rose)



Plot a monthly plot of the Time Series(Rose)



Defining the Time Series(Rose)

Reading the top records (Rose)

Out[22]:

	YearMonth	Rose	Time_Stamp
0	1980-01	112.0	1980-01-31
1	1980-02	118.0	1980-02-29

Making the TimeStamp as Index (Rose)

Out[23]:

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

Dropping the YearMonth Column

Out[24]:

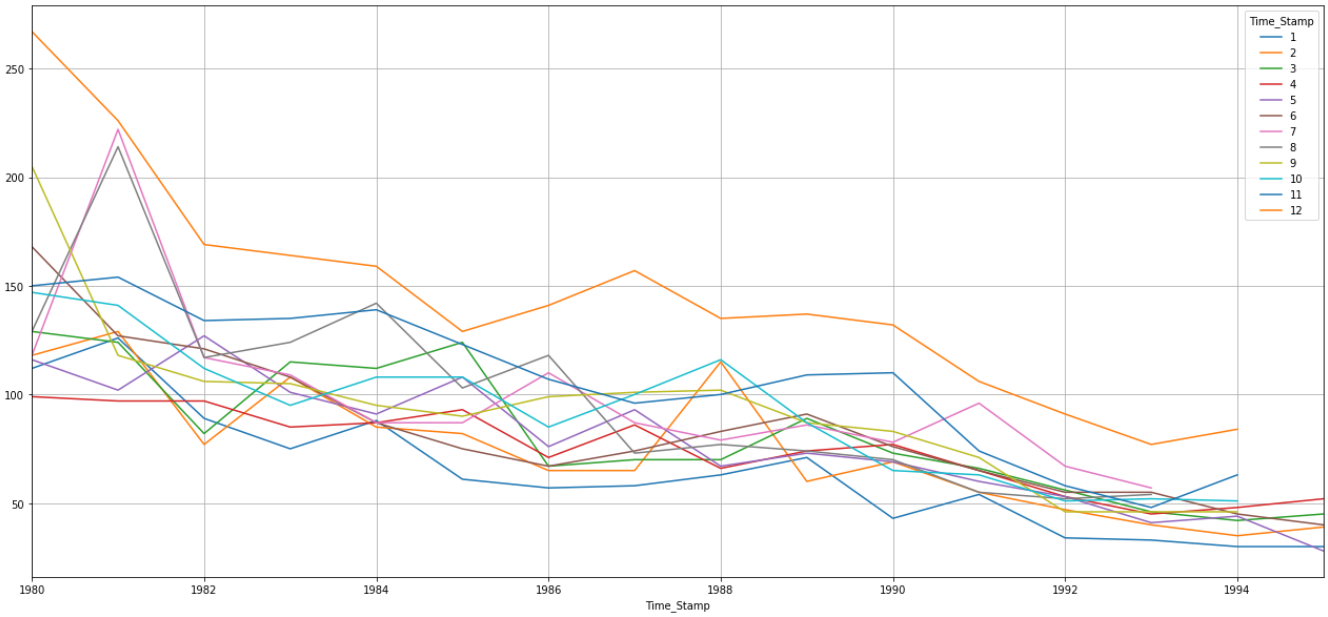
Time_Stamp	Rose
1980-01-31	112.0
1980-02-29	118.0
1980-03-31	129.0
1980-04-30	99.0
1980-05-31	116.0

Yearly Rose Sales across the year

Out[26]:

Time_Stamp	1	2	3	4	5	6	7	8	9	10	11	12
Time_Stamp												
1980	112.0	118.0	129.0	99.0	116.0	168.0	118.0	129.0	205.0	147.0	150.0	267.0
1981	126.0	129.0	124.0	97.0	102.0	127.0	222.0	214.0	118.0	141.0	154.0	226.0
1982	89.0	77.0	82.0	97.0	127.0	121.0	117.0	117.0	106.0	112.0	134.0	169.0
1983	75.0	108.0	115.0	85.0	101.0	108.0	109.0	124.0	105.0	95.0	135.0	164.0
1984	88.0	85.0	112.0	87.0	91.0	87.0	87.0	142.0	95.0	108.0	139.0	159.0
1985	61.0	82.0	124.0	93.0	108.0	75.0	87.0	103.0	90.0	108.0	123.0	129.0
1986	57.0	65.0	67.0	71.0	76.0	67.0	110.0	118.0	99.0	85.0	107.0	141.0
1987	58.0	65.0	70.0	86.0	93.0	74.0	87.0	73.0	101.0	100.0	96.0	157.0
1988	63.0	115.0	70.0	66.0	67.0	83.0	79.0	77.0	102.0	116.0	100.0	135.0
1989	71.0	60.0	89.0	74.0	73.0	91.0	86.0	74.0	87.0	87.0	109.0	137.0
1990	43.0	69.0	73.0	77.0	69.0	76.0	78.0	70.0	83.0	65.0	110.0	132.0
1991	54.0	55.0	66.0	65.0	60.0	65.0	96.0	55.0	71.0	63.0	74.0	106.0
1992	34.0	47.0	56.0	53.0	53.0	55.0	67.0	52.0	46.0	51.0	58.0	91.0
1993	33.0	40.0	46.0	45.0	41.0	55.0	57.0	54.0	46.0	52.0	48.0	77.0
1994	30.0	35.0	42.0	48.0	44.0	45.0	NaN	NaN	46.0	51.0	63.0	84.0
1995	30.0	39.0	45.0	52.0	28.0	40.0	62.0	NaN	NaN	NaN	NaN	NaN

Monthly Rose Trend



Checking and handling missing values

Out[28]:
```
YearMonth
1994-01-01    30.0
1994-02-01    35.0
1994-03-01    42.0
1994-04-01    48.0
1994-05-01    44.0
1994-06-01    45.0
1994-07-01     NaN
1994-08-01     NaN
1994-09-01    46.0
1994-10-01    51.0
1994-11-01    63.0
1994-12-01    84.0
Name: Rose, dtype: float64
```

Imputing the Missing Values(Rose)

Checking the Missing Values after Imputing (Rose)

Out[30]:

Time_Stamp	Rose
1994-01-31	30.000000
1994-02-28	35.000000
1994-03-31	42.000000
1994-04-30	48.000000
1994-05-31	44.000000
1994-06-30	45.000000
1994-07-31	43.693064
1994-08-31	44.326877
1994-09-30	46.000000
1994-10-31	51.000000
1994-11-30	63.000000
1994-12-31	84.000000

Plot after imputing the Missing values

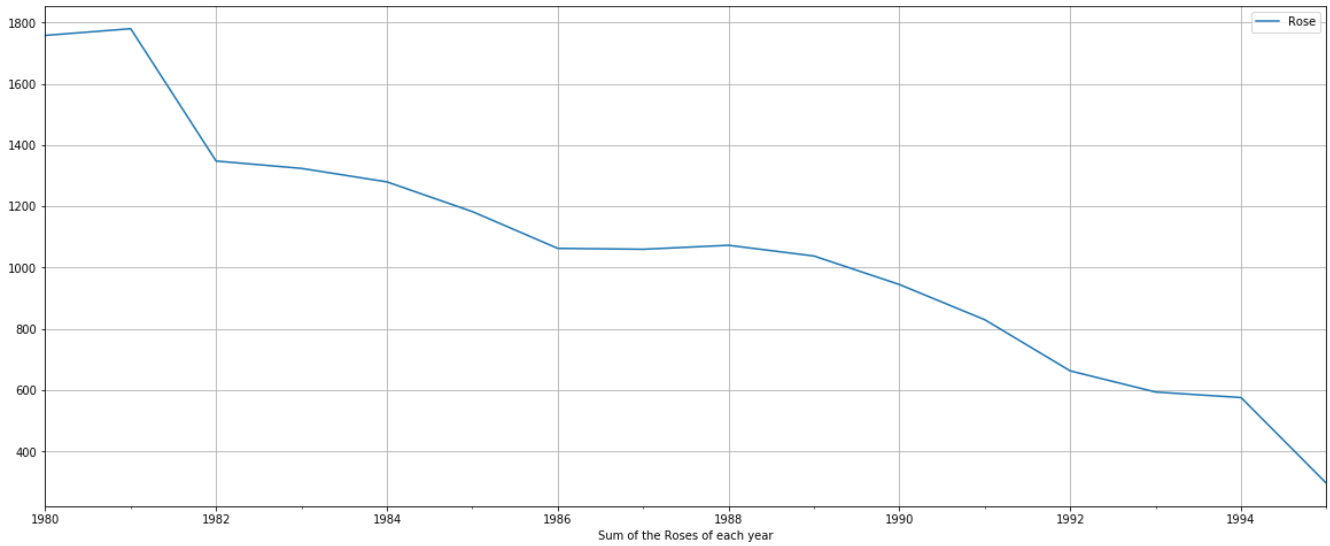
Out[31]:

Time_Stamp	1	2	3	4	5	6	7	8	9	10	11	12
1980	112.0	118.0	129.0	99.0	116.0	168.0	118.000000	129.000000	205.0	147.0	150.0	267.0
1981	126.0	129.0	124.0	97.0	102.0	127.0	222.000000	214.000000	118.0	141.0	154.0	226.0
1982	89.0	77.0	82.0	97.0	127.0	121.0	117.000000	117.000000	106.0	112.0	134.0	169.0
1983	75.0	108.0	115.0	85.0	101.0	108.0	109.000000	124.000000	105.0	95.0	135.0	164.0
1984	88.0	85.0	112.0	87.0	91.0	87.0	87.000000	142.000000	95.0	108.0	139.0	159.0
1985	61.0	82.0	124.0	93.0	108.0	75.0	87.000000	103.000000	90.0	108.0	123.0	129.0
1986	57.0	65.0	67.0	71.0	76.0	67.0	110.000000	118.000000	99.0	85.0	107.0	141.0
1987	58.0	65.0	70.0	86.0	93.0	74.0	87.000000	73.000000	101.0	100.0	96.0	157.0
1988	63.0	115.0	70.0	66.0	67.0	83.0	79.000000	77.000000	102.0	116.0	100.0	135.0
1989	71.0	60.0	89.0	74.0	73.0	91.0	86.000000	74.000000	87.0	87.0	109.0	137.0
1990	43.0	69.0	73.0	77.0	69.0	76.0	78.000000	70.000000	83.0	65.0	110.0	132.0
1991	54.0	55.0	66.0	65.0	60.0	65.0	96.000000	55.000000	71.0	63.0	74.0	106.0
1992	34.0	47.0	56.0	53.0	53.0	55.0	67.000000	52.000000	46.0	51.0	58.0	91.0
1993	33.0	40.0	46.0	45.0	41.0	55.0	57.000000	54.000000	46.0	52.0	48.0	77.0
1994	30.0	35.0	42.0	48.0	44.0	45.0	43.693064	44.326877	46.0	51.0	63.0	84.0
1995	30.0	39.0	45.0	52.0	28.0	40.0	62.000000	NaN	NaN	NaN	NaN	NaN

Rose Yearly plot

Out[32]:

Time_Stamp	Rose
1980-12-31	1758.0
1981-12-31	1780.0
1982-12-31	1348.0
1983-12-31	1324.0
1984-12-31	1280.0

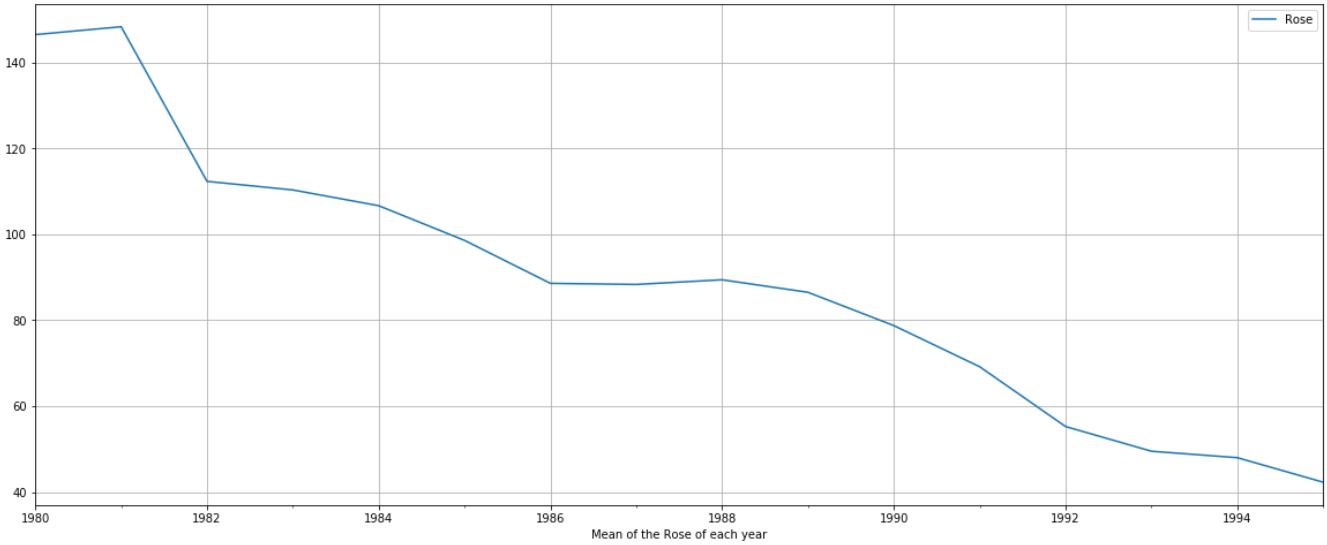


Rose yearly mean

Out[34]:

Time_Stamp	Rose
1980-12-31	146.500000
1981-12-31	148.333333
1982-12-31	112.333333
1983-12-31	110.333333
1984-12-31	106.666667

Rose yearly mean plot

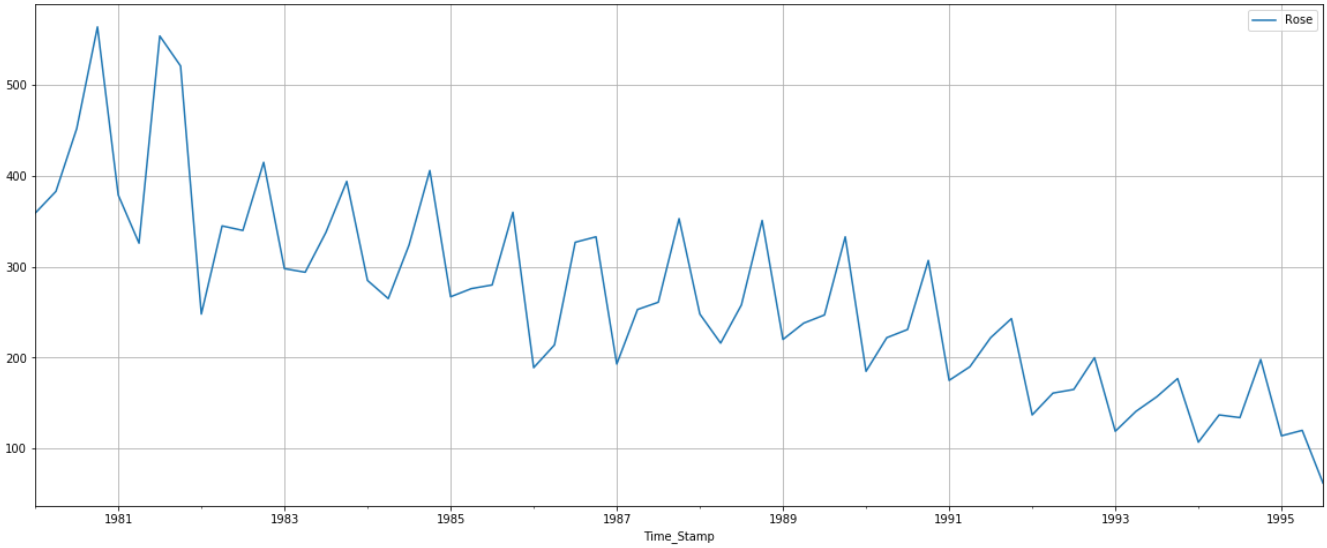


Rose Quarterly Sum

Out[36]:

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

Rose Quarterly plot

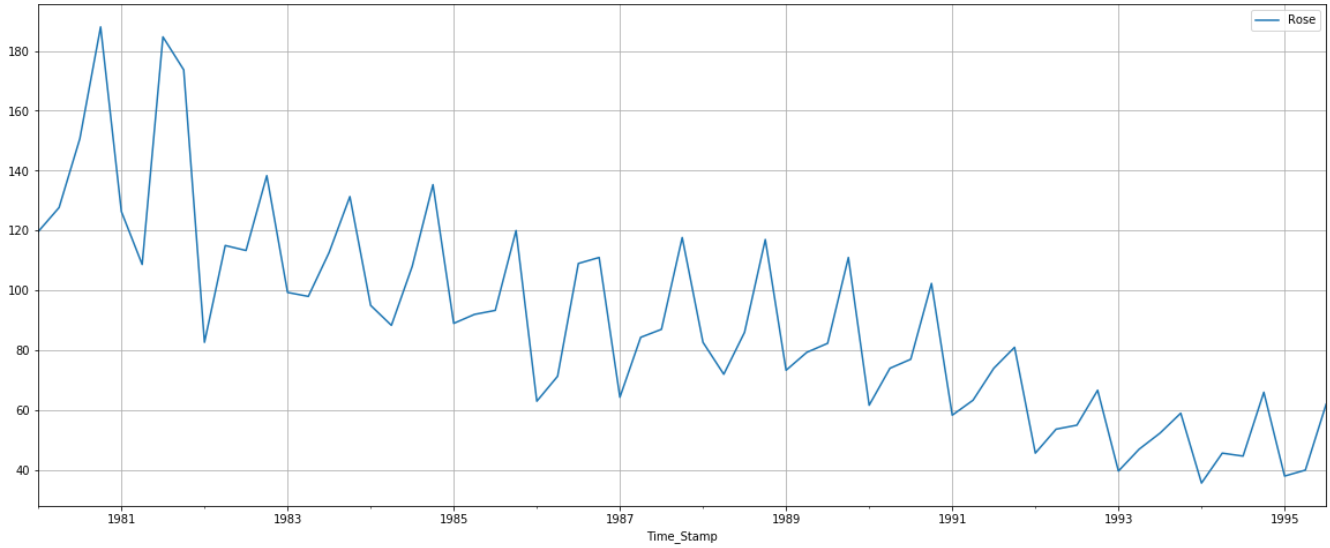


Rose quartely mean

Out[38]:

Time_Stamp	Rose
1980-03-31	119.666667
1980-06-30	127.666667
1980-09-30	150.666667
1980-12-31	188.000000
1981-03-31	126.333333

Rose Quarterly Mean Plot

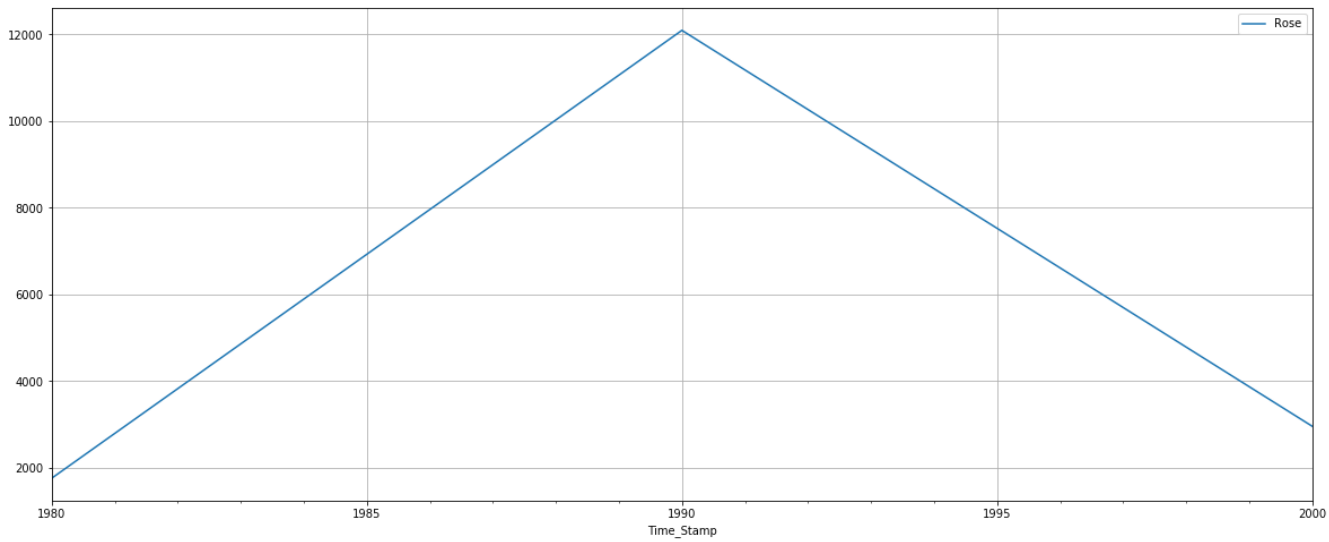


Rose decade plot

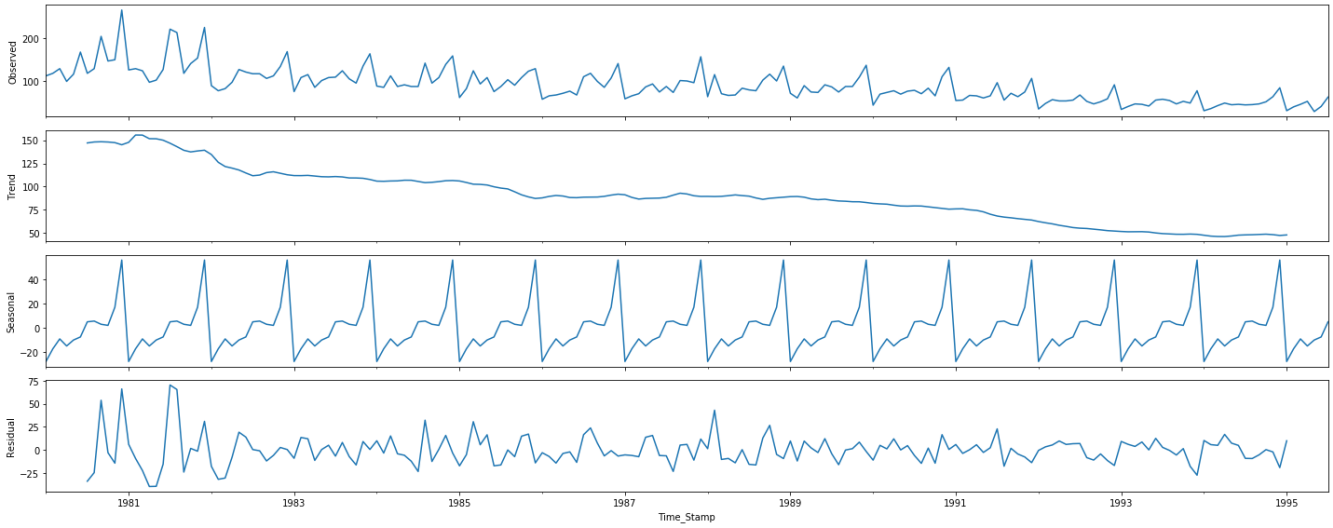
Out[40]:

Time_Stamp	Rose
1980-12-31	1758.000000
1990-12-31	12094.000000
2000-12-31	2959.019941

Rose Decade plot



Decompose the Time Series(Rose)



Trend

Time_Stamp	Rose
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	147.083333
1980-08-31	148.125000
1980-09-30	148.375000
1980-10-31	148.083333
1980-11-30	147.416667
1980-12-31	145.125000

Seasonality

Time_Stamp	Rose
1980-01-31	-27.892252
1980-02-29	-17.422041
1980-03-31	-9.268252
1980-04-30	-15.080752
1980-05-31	-10.178966
1980-06-30	-7.661109
1980-07-31	4.803952
1980-08-31	5.426762
1980-09-30	2.791081
1980-10-31	1.888304
1980-11-30	16.863304
1980-12-31	55.729970

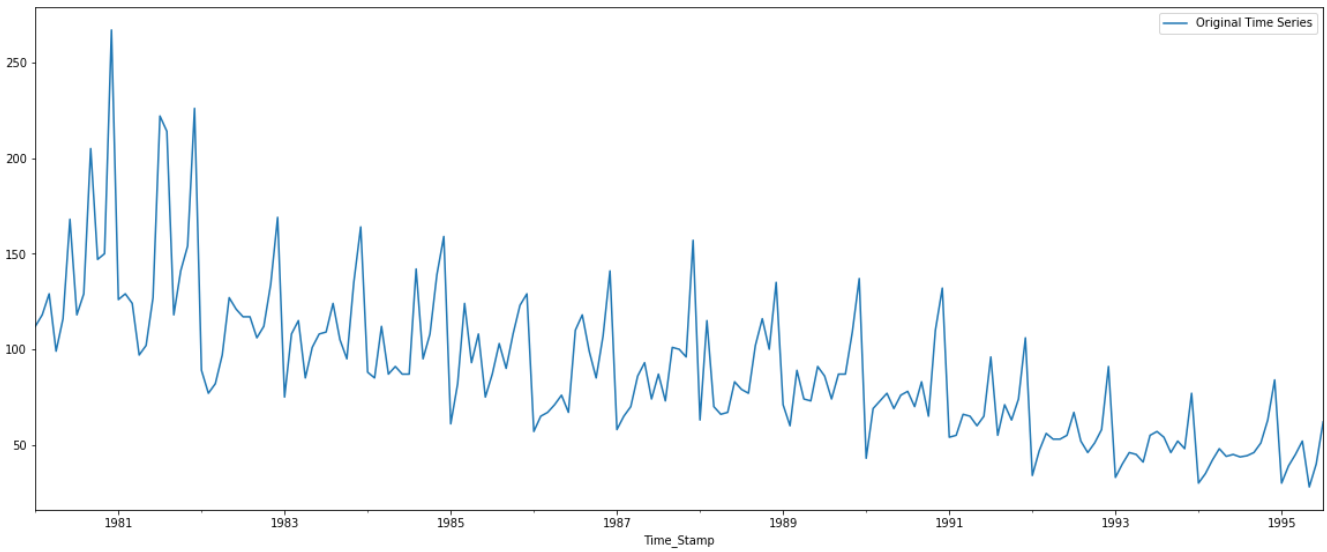
Residual

Time_Stamp	Rose
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	-33.887286
1980-08-31	-24.551762
1980-09-30	53.833919
1980-10-31	-2.971637
1980-11-30	-14.279970
1980-12-31	66.145030

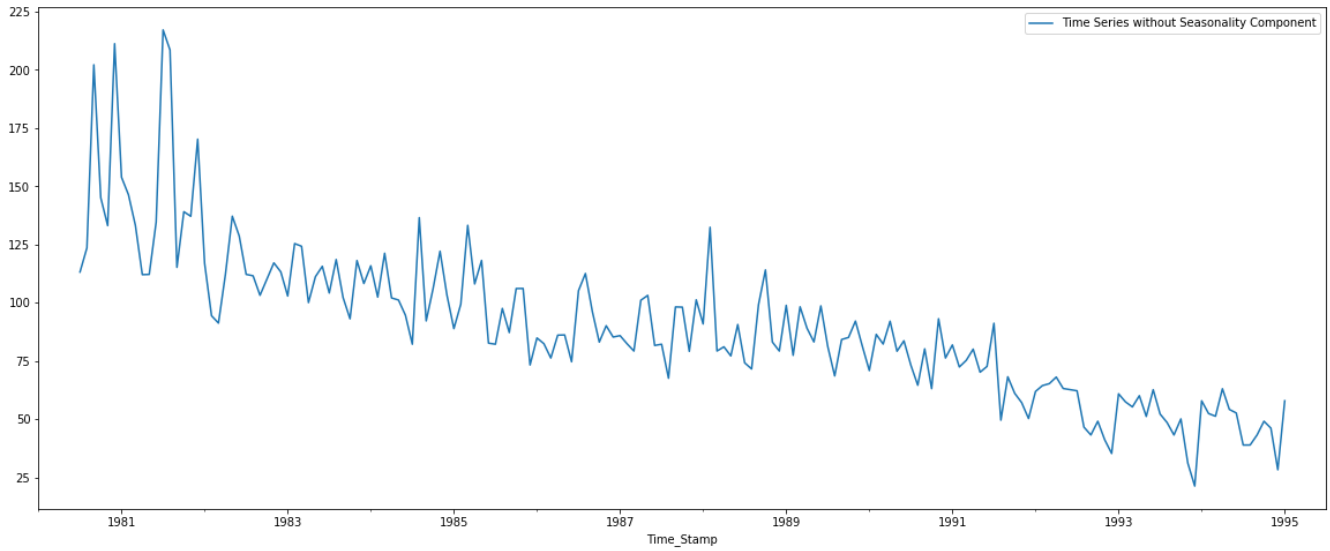
Out[47]:

Time_Stamp	Rose
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	113.196048
1980-08-31	123.573238
1980-09-30	202.208919
1980-10-31	145.111696
1980-11-30	133.136696
1980-12-31	211.270030

Plot Original Series(Rose)



Time Series without Seasonality Component(Rose)



EDA Rose Time Series

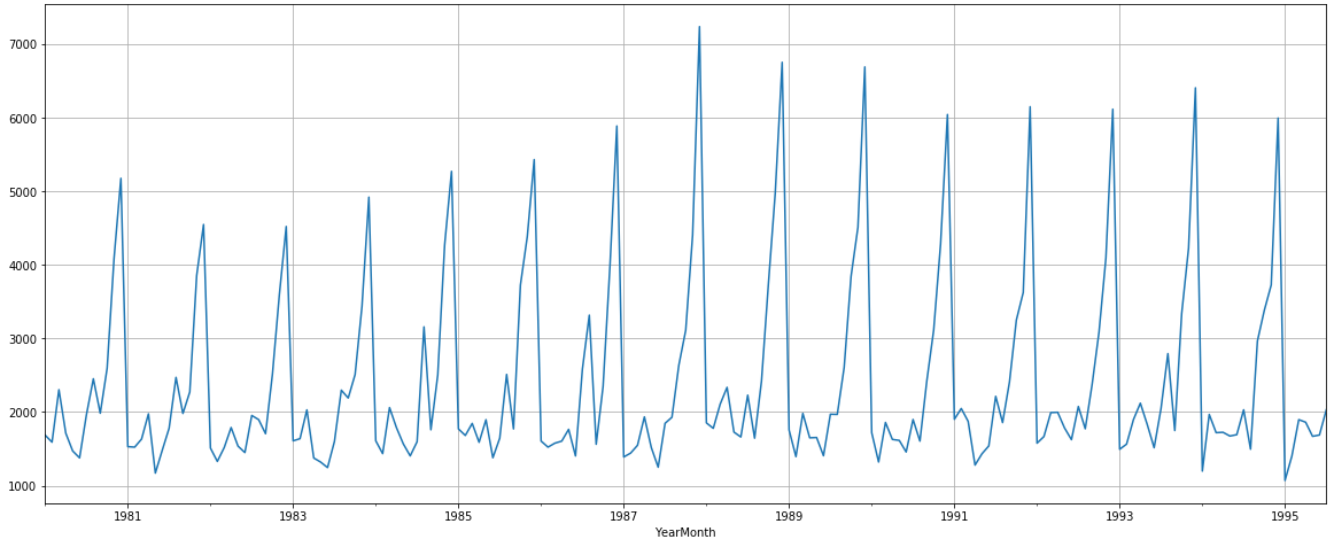
- There is upward trend in every Decemember across the years
- Janurary and April having downword trend rest other months having zig zag trend across the years
- There is Missing Values in the Series

Reading the Time SeriesData(Sparkling)

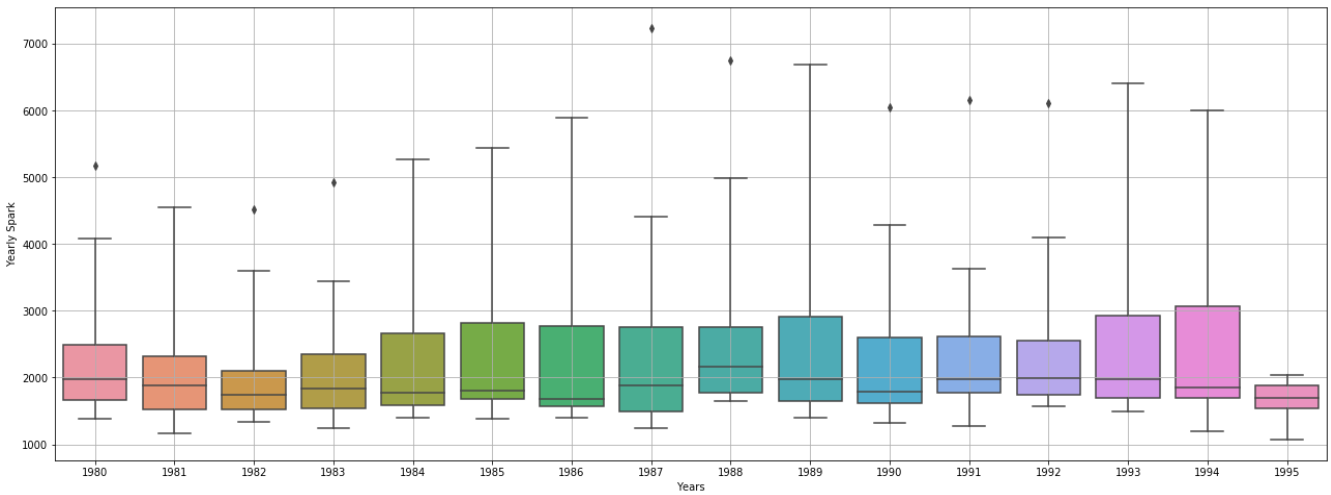
Summary of the Data(Sparkling)

Out[51]:
```
count     187.000000
mean     2402.417112
std      1295.111540
min      1070.000000
25%      1605.000000
50%      1874.000000
75%      2549.000000
max      7242.000000
Name: Sparkling, dtype: float64
```

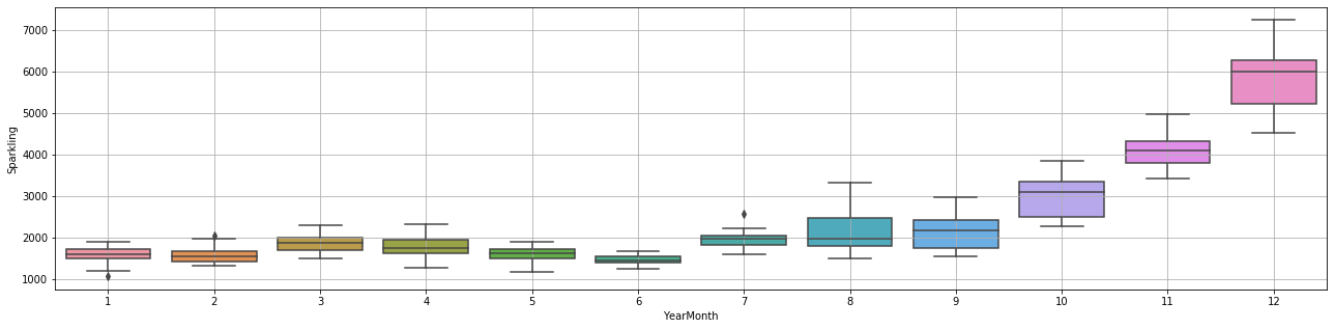
Ploting the data(Sparkling)



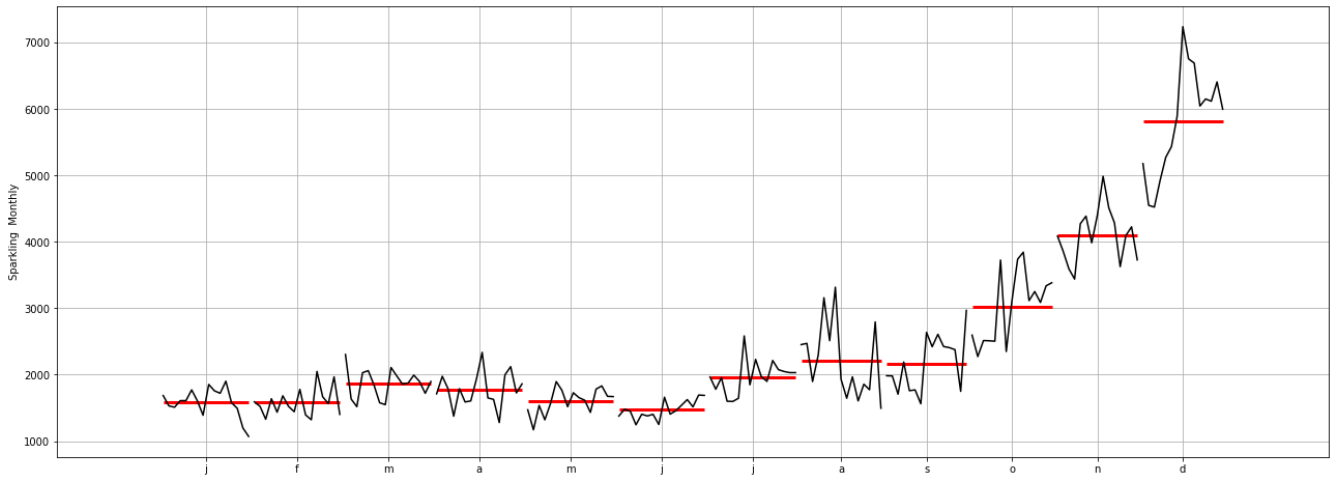
Sparking yearly plot



Sparking Monthly plot



Sparkling Monthly plot



Checking the existing dataset columns(Sparkling)

Out[56]:

	YearMonth	Sparkling	Time_Stamp
0	1980-01	1686	1980-01-31
1	1980-02	1591	1980-02-29

Setting the Timestamp as Index(Sparkling)

Out[57]:

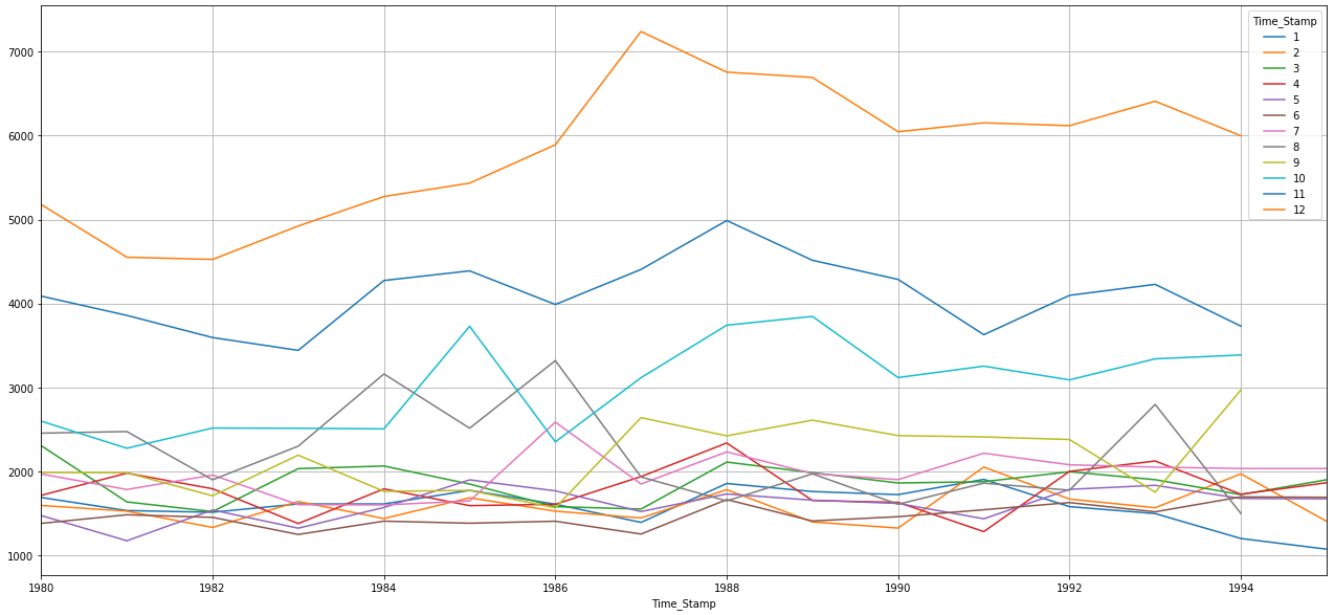
Time_Stamp	YearMonth	Sparkling
1980-01-31	1980-01	1686
1980-02-29	1980-02	1591
1980-03-31	1980-03	2304
1980-04-30	1980-04	1712
1980-05-31	1980-05	1471

Dropping the YearMonth Column(Sparkling)

Out[58]:

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

yearly_plot_across_months(Sparkling)



Yearly data across Month(Sparkling)

Out[60]:

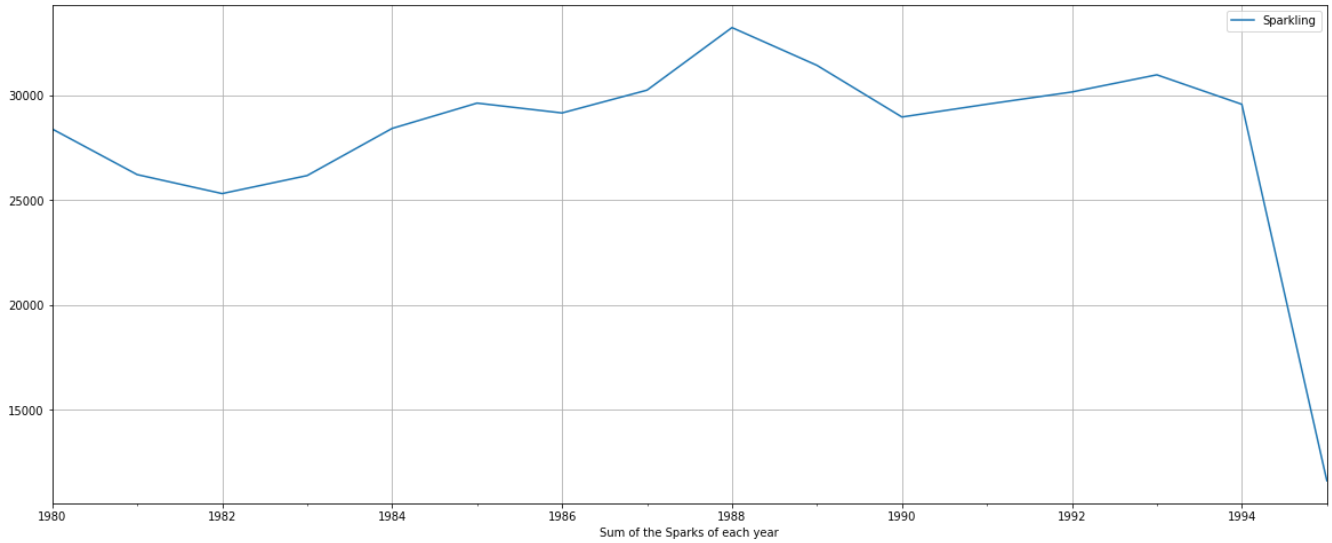
Time_Stamp	1	2	3	4	5	6	7	8	9	10	11	12
Time_Stamp												
1980	1686.0	1591.0	2304.0	1712.0	1471.0	1377.0	1966.0	2453.0	1984.0	2596.0	4087.0	5179.0
1981	1530.0	1523.0	1633.0	1976.0	1170.0	1480.0	1781.0	2472.0	1981.0	2273.0	3857.0	4551.0
1982	1510.0	1329.0	1518.0	1790.0	1537.0	1449.0	1954.0	1897.0	1706.0	2514.0	3593.0	4524.0
1983	1609.0	1638.0	2030.0	1375.0	1320.0	1245.0	1600.0	2298.0	2191.0	2511.0	3440.0	4923.0
1984	1609.0	1435.0	2061.0	1789.0	1567.0	1404.0	1597.0	3159.0	1759.0	2504.0	4273.0	5274.0
1985	1771.0	1682.0	1846.0	1589.0	1896.0	1379.0	1645.0	2512.0	1771.0	3727.0	4388.0	5434.0
1986	1606.0	1523.0	1577.0	1605.0	1765.0	1403.0	2584.0	3318.0	1562.0	2349.0	3987.0	5891.0
1987	1389.0	1442.0	1548.0	1935.0	1518.0	1250.0	1847.0	1930.0	2638.0	3114.0	4405.0	7242.0
1988	1853.0	1779.0	2108.0	2336.0	1728.0	1661.0	2230.0	1645.0	2421.0	3740.0	4988.0	6757.0
1989	1757.0	1394.0	1982.0	1650.0	1654.0	1406.0	1971.0	1968.0	2608.0	3845.0	4514.0	6694.0
1990	1720.0	1321.0	1859.0	1628.0	1615.0	1457.0	1899.0	1605.0	2424.0	3116.0	4286.0	6047.0
1991	1902.0	2049.0	1874.0	1279.0	1432.0	1540.0	2214.0	1857.0	2408.0	3252.0	3627.0	6153.0
1992	1577.0	1667.0	1993.0	1997.0	1783.0	1625.0	2076.0	1773.0	2377.0	3088.0	4096.0	6119.0
1993	1494.0	1564.0	1898.0	2121.0	1831.0	1515.0	2048.0	2795.0	1749.0	3339.0	4227.0	6410.0
1994	1197.0	1968.0	1720.0	1725.0	1674.0	1693.0	2031.0	1495.0	2968.0	3385.0	3729.0	5999.0
1995	1070.0	1402.0	1897.0	1862.0	1670.0	1688.0	2031.0	NaN	NaN	NaN	NaN	NaN

Sparkling yearly Data

Out[61]:

Time_Stamp	Sparkling
1980-12-31	28406
1981-12-31	26227
1982-12-31	25321
1983-12-31	26180
1984-12-31	28431

Sparkling yearly plot

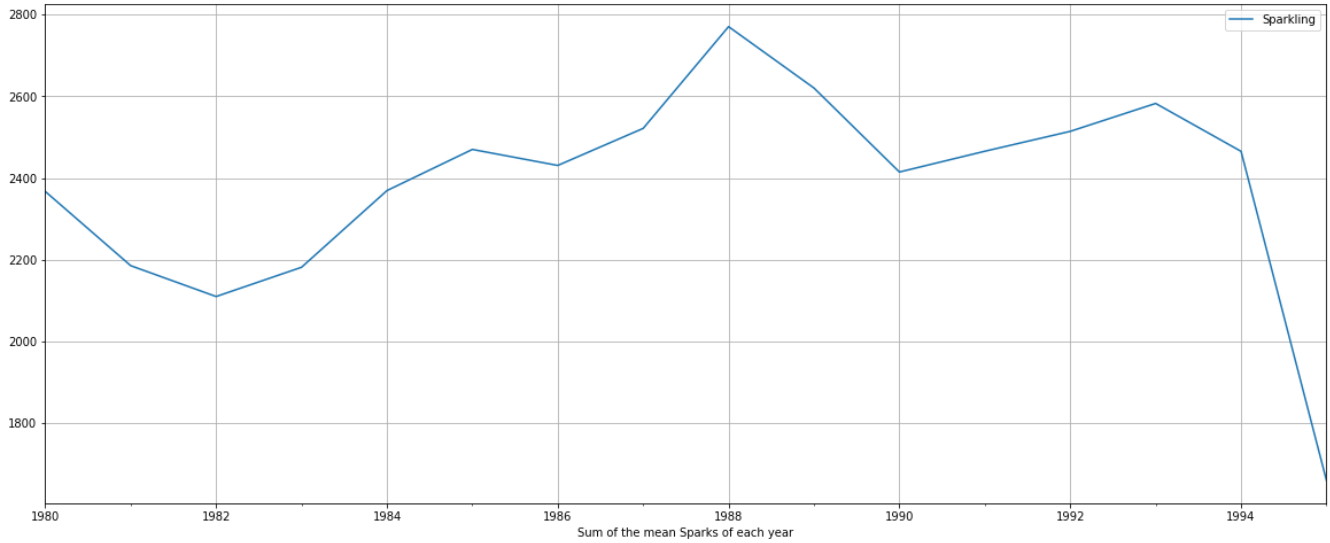


Sparkling yearly mean data

Out[63]:

Time_Stamp	Sparkling
1980-12-31	2367.166667
1981-12-31	2185.583333
1982-12-31	2110.083333
1983-12-31	2181.666667
1984-12-31	2369.250000

Sparkling yearly plot

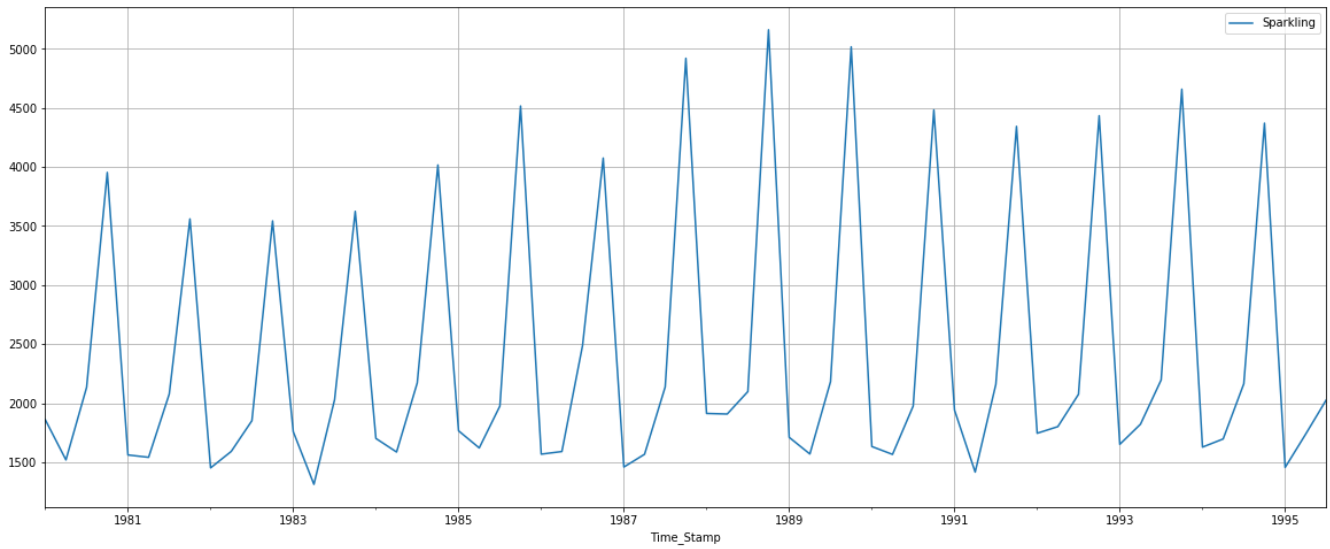


Sparkling Quarterly Data

Out[65]:

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

Sparkling Quarterly plot

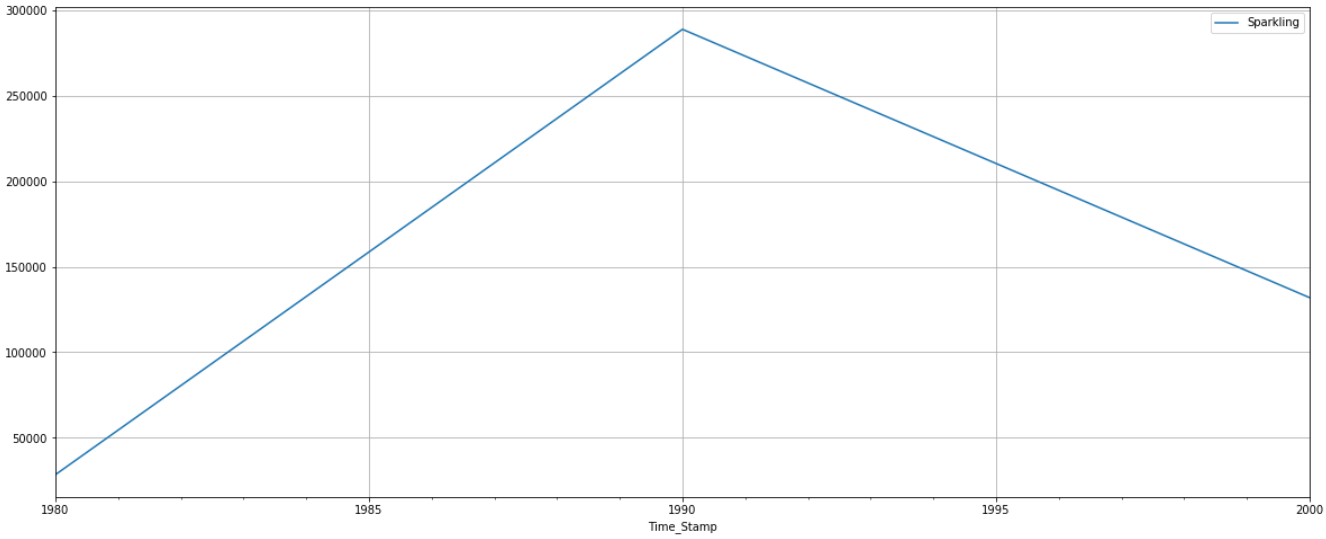


Sparkling Decade data

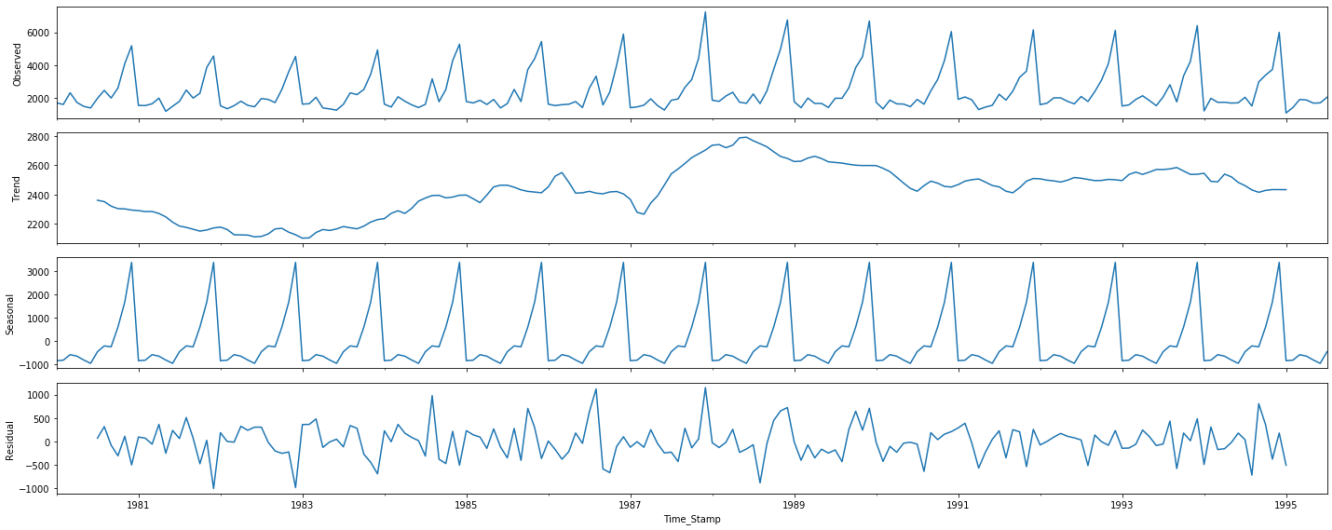
Out[67]:

Time_Stamp	Sparkling
1980-12-31	28406
1990-12-31	288893
2000-12-31	131953

Sparkling Decade plot

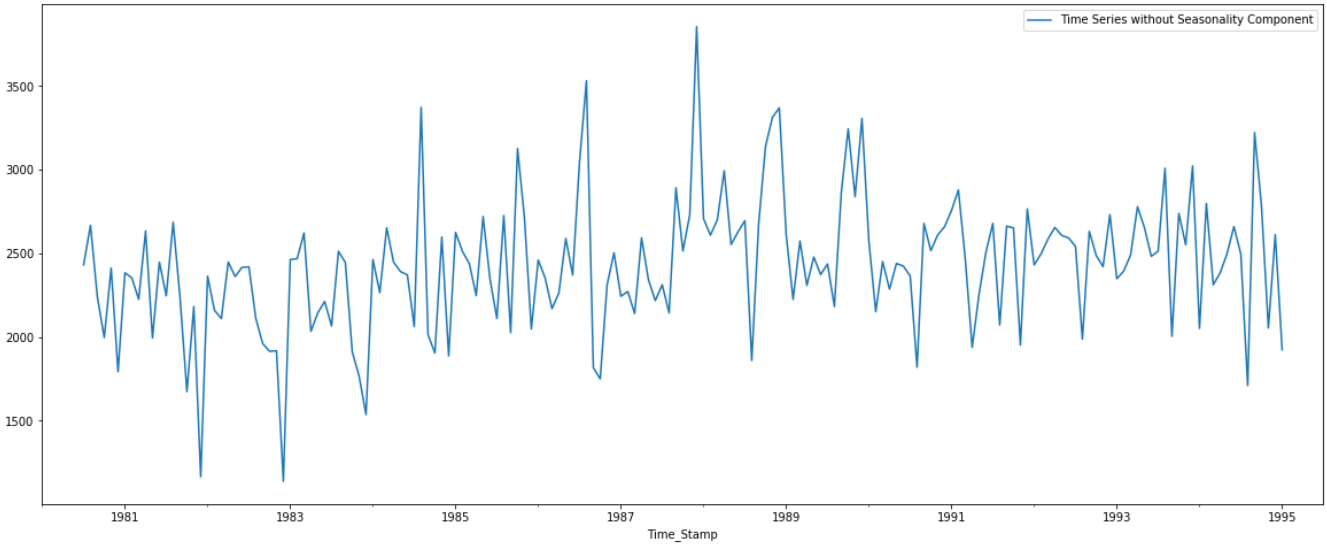


Decomposition the Time Series(Sparkling)

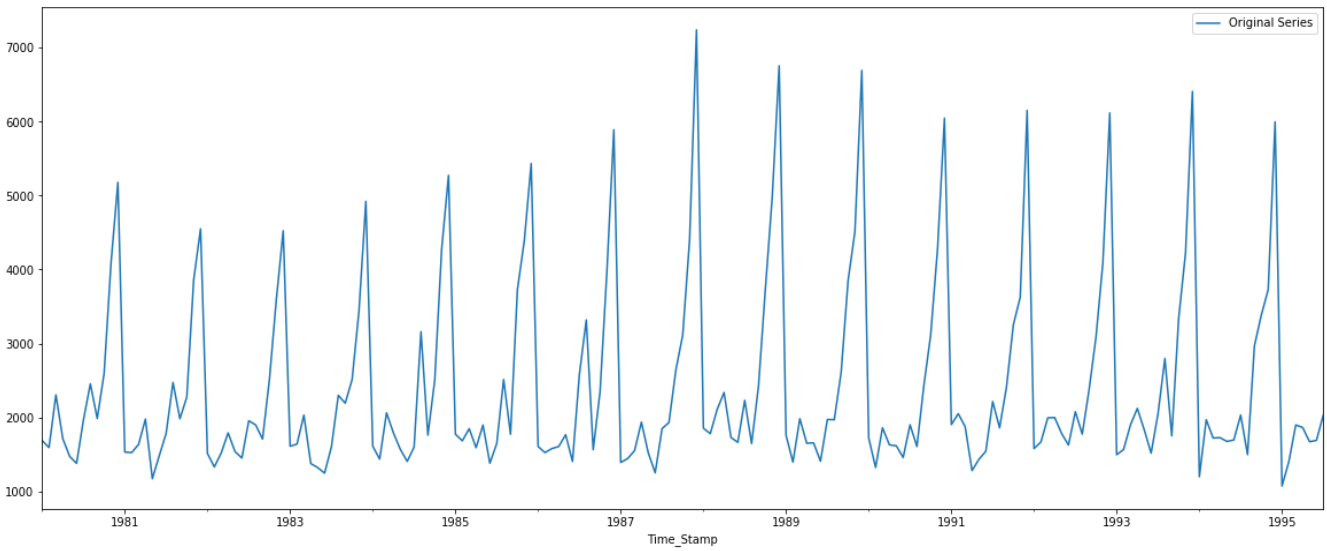


```
Trend
               Sparkling
Time_Stamp
1980-01-31          NaN
1980-02-29          NaN
1980-03-31          NaN
1980-04-30          NaN
1980-05-31          NaN
1980-06-30          NaN
1980-07-31  2360.666667
1980-08-31  2351.333333
1980-09-30  2320.541667
1980-10-31  2303.583333
1980-11-30  2302.041667
1980-12-31  2293.791667

Seasonality
               Sparkling
Time_Stamp
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
1980-11-30  1675.067179
1980-12-31  3386.983846

Residual
              Sparkling
Time_Stamp
1980-01-31         NaN
1980-02-29         NaN
1980-03-31         NaN
1980-04-30         NaN
1980-05-31         NaN
1980-06-30         NaN
1980-07-31   70.835599
1980-08-31  315.999487
1980-09-30  -81.864401
1980-10-31 -307.353290
1980-11-30  109.891154
1980-12-31 -501.775513
```

Time Series without Seasonality Component(Sparkling)



Plotting Original Time Series(Sparkling)



EDA Sparkling Times Series

- In the Case of Sparkling September to Decemember trend is upward and This trend is accross the years . which is good sign .
- January having downward trend accross the years and other Months having zig zag trend across the various years.
- There is no Missing values in the Series

Question 3 Split the data into training and test for Rose case. The test data should start in 1991.

Checking the shape ,Some First and last rows of the Rose TimeSeries

```
Shape of  Rose Training Data
(132, 1)
Shape of  Rose Test Data
(55, 1)
First few rows of Rose Training Data
```

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

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

Time_Stamp	Rose
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 Rose Test Data
```

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

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

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

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

```
Shape of  Spark Training Data
(132, 1)
Shape of  Spark Test Data
(55, 1)
First few rows of Spark  Training Data
```

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

```
Last few rows of Spark Training Data
```

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

```
First few rows ofSpark Test Data
```

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

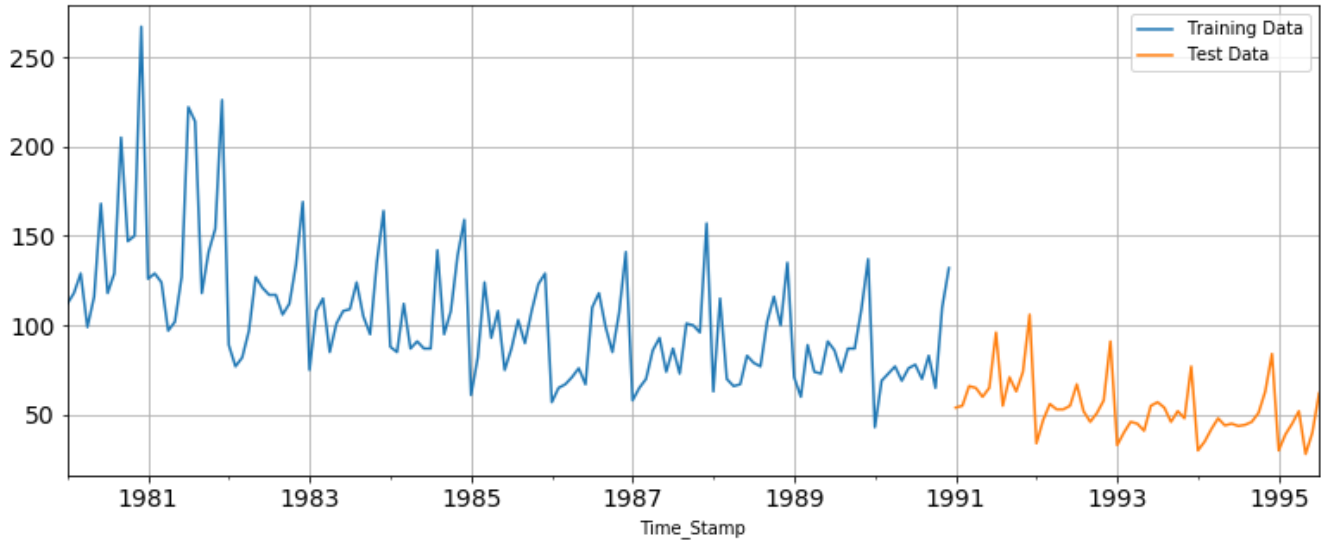
```
Last few rows of Spark Test Data
```

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

Question 4a. 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.
- Please do try to build as many models as possible and as many iterations of models as possible with different parameters.

Plotting Rose train and test plot



Building different models and comparing the accuracy metrics(Rose)

Linear Regression

Checking Training and Test Data(Rose)

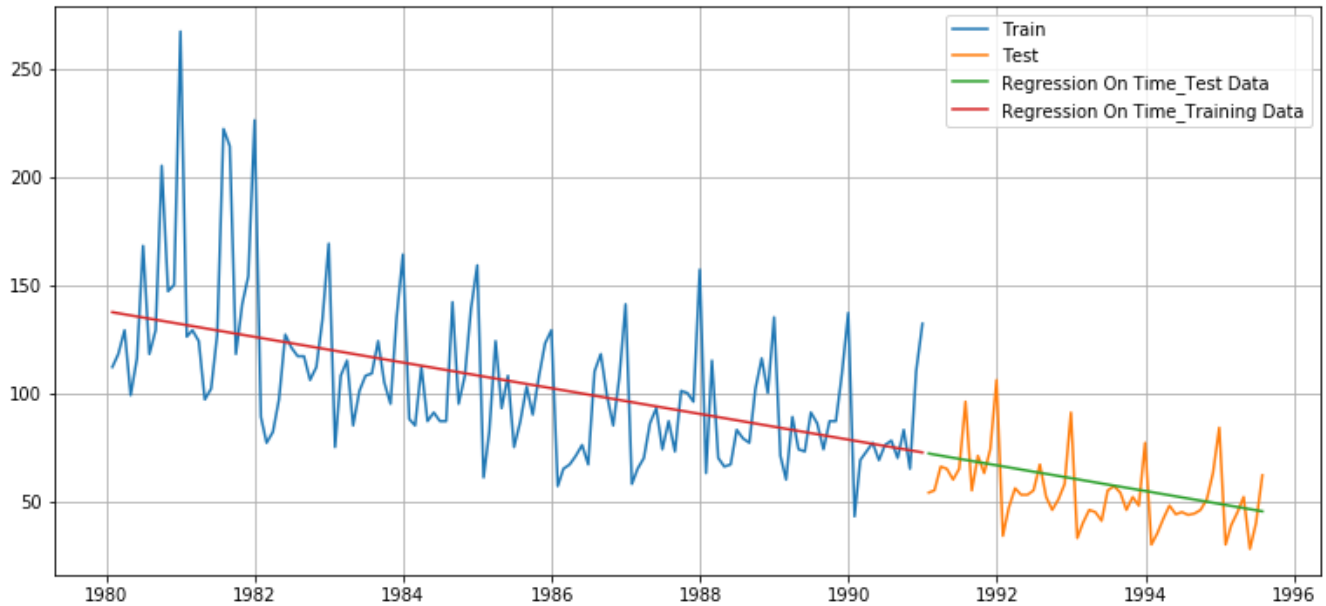
```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
4, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 6
7, 68, 69, 70, 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, 10
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 12
5, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 1
67, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 18
4, 185, 186, 187]
```

Defining the Linear Regression Model and checking the Traing and Test Data(Rose)

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

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

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

Last few rows of Test Data
             Rose  time
Time_Stamp
1995-03-31  45.0   183
1995-04-30  52.0   184
1995-05-31  28.0   185
1995-06-30  40.0   186
1995-07-31  62.0   187
```

Out[87]:
```
LinearRegression()
```

Plotting Linear Regression Train and Test data(Rose)



Defining the functions for calculating the accuracy metrics

Mean Absolute Percentage Error - Function Definition

RegressionOnTime forecast on the Training Data(Rose)

```
For Rose RegressionOnTime forecast on the Training Data,  RMSE is 30.718 MAPE is 21.22
```

RegressionOnTime forecast on the Test Data(Rose)

```
For Rose RegressionOnTime forecast on the Test Data,  RMSE is 15.291 MAPE is 22.94
```

Storing the Rose Regression Result In Dataframe

Out[95]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.29146	22.94

#Model 2(Rose): Naive Approach: $\hat{y}\ t$+1=yt

Defining the Naive Model(Rose)

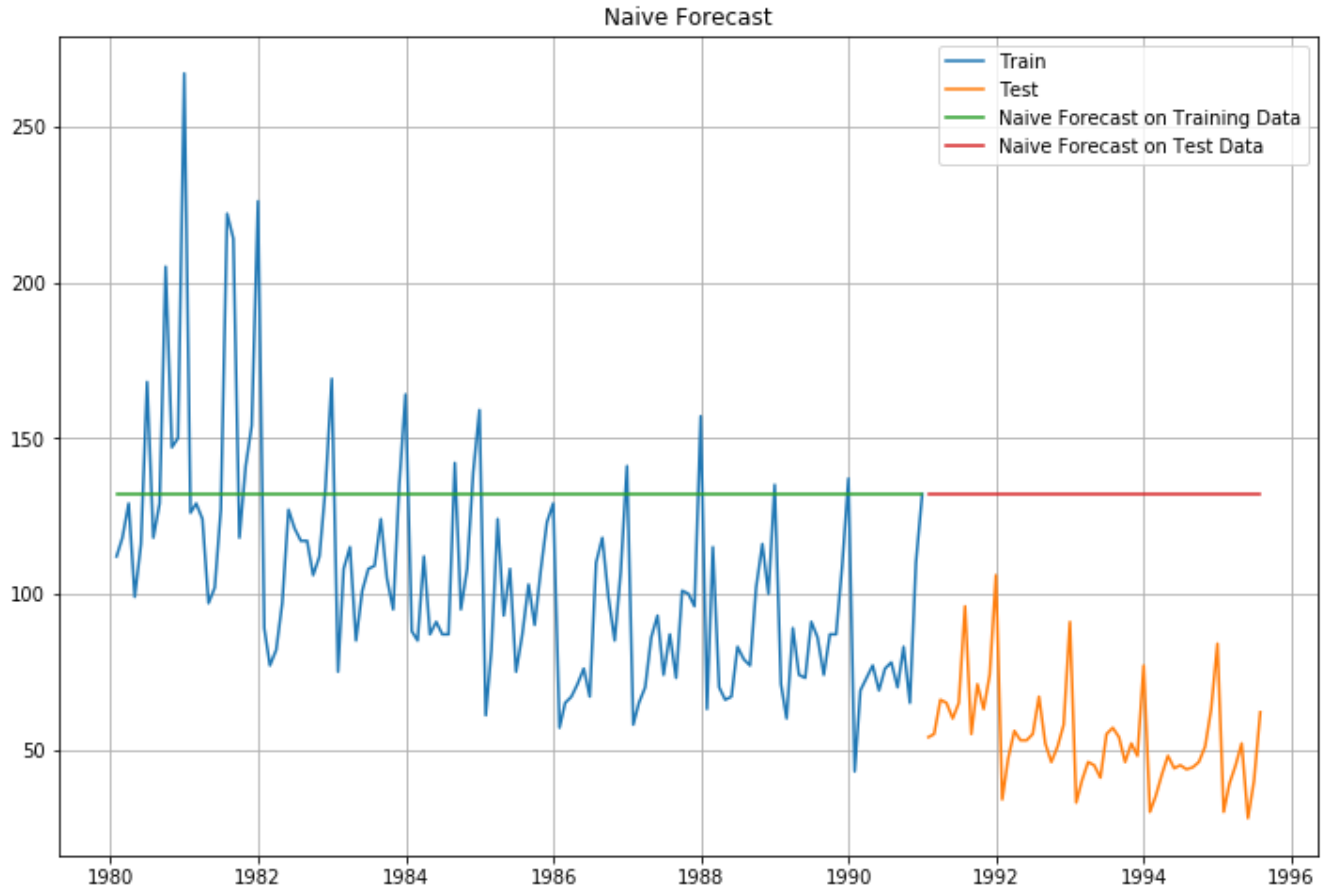
Let us get the last value (or the most recent value) of the training set.

Out[98]:
```
Time_Stamp
1980-01-31    132.0
1980-02-29    132.0
1980-03-31    132.0
1980-04-30    132.0
1980-05-31    132.0
Name: naive, dtype: float64
```

Let us get the last value (or the most recent value) of the Test set.

Out[99]:
```
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
```

Naive Forecast on Train and Test Data(Rose)



RoseModel Evaluation(Training Data - RMSE and MAPE)

```
For Rose Naive Model forecast on the Training Data,  RMSE is 45.064 MAPE is 36.38
```

Rose Model Evaluation(Test Data - RMSE and MAPE)

```
For Rose RegressionOnTime forecast on the Test Data,  RMSE is 79.778 MAPE is 145.35
```

Storing the Naive Result In Dataframe

Out[103]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35

Method 3(Rose): Simple Average

Defining the Simple Average Train Model(Rose)

Out[105]:

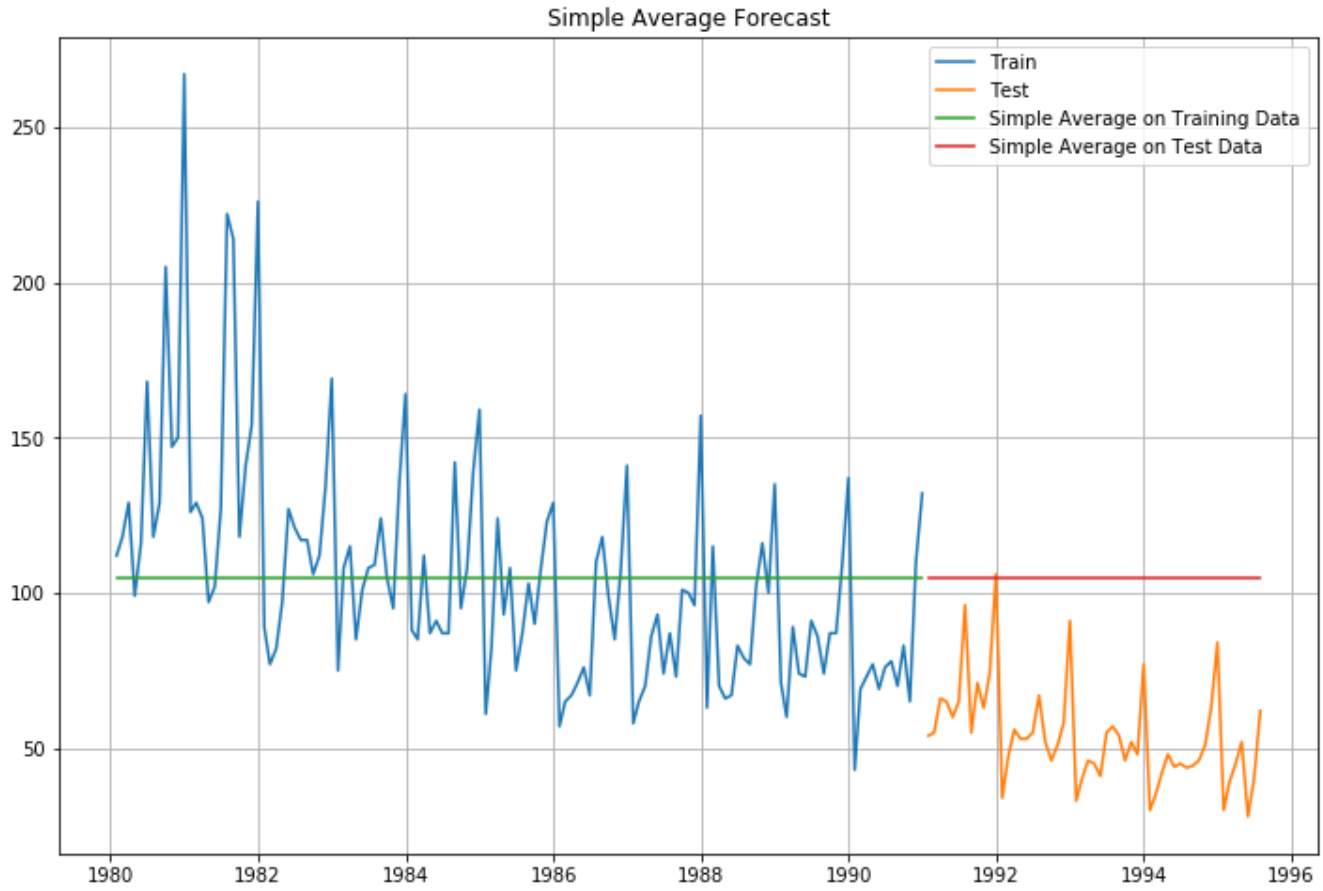
Time_Stamp	Rose	mean_forecast
1980-01-31	112.0	104.939394
1980-02-29	118.0	104.939394
1980-03-31	129.0	104.939394
1980-04-30	99.0	104.939394
1980-05-31	116.0	104.939394

Defining the Simple Average Test Model(Rose)

Out[106]:

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

Plot on Simple Average on Training and Test Data(Rose)



Model Evaluation(Training Data - RMSE and MAPE)

```
For Rose Simple Average Model forecast on the Training Data,  RMSE is 36.034 MAPE is 2
5.39
```

Model Evaluation(Test Data - RMSE and MAPE)

```
For Rose Simple Average forecast on the Test Data,  RMSE is 53.522 MAPE is 95.13
```

Storing the Roe Simple Moving Average Result In Dataframe

Out[111]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13

Rose Method 4: Moving Average(MA)

Out[112]:

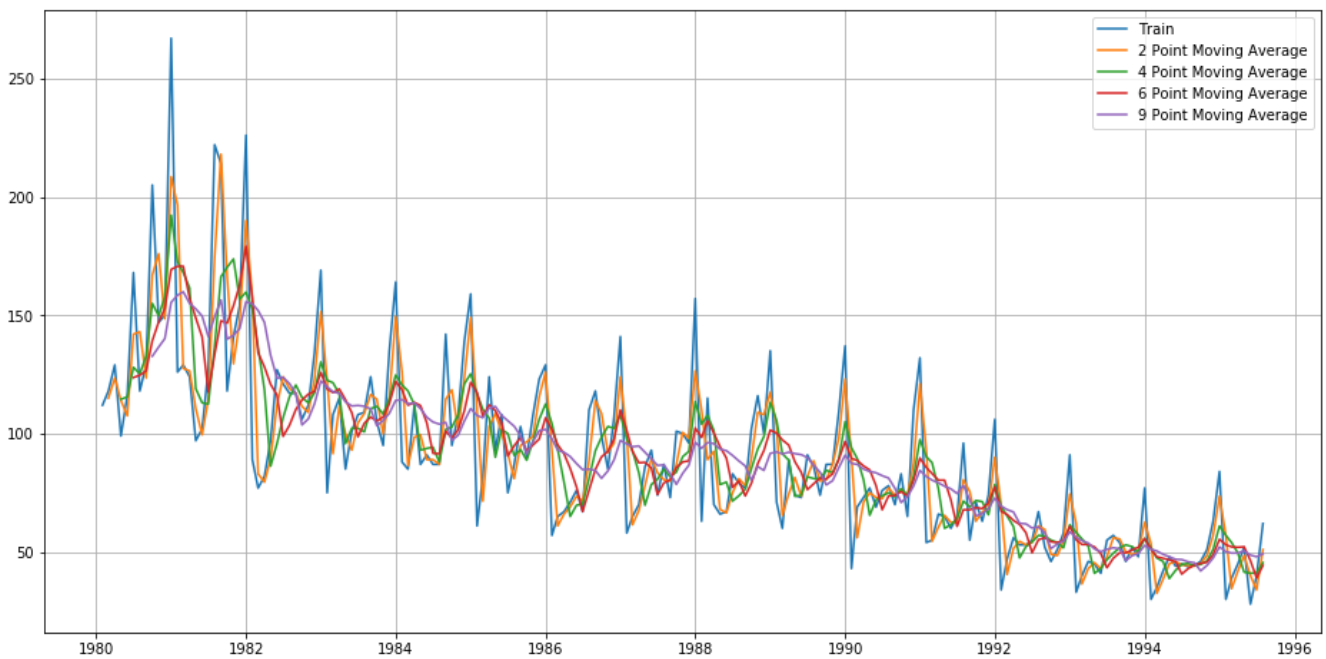
Time_Stamp	Rose
1980-01-31	112.0
1980-02-29	118.0
1980-03-31	129.0
1980-04-30	99.0
1980-05-31	116.0

Rose (Defining the Moveing Average)

Out[113]:

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

Plotting on the whole data(Rose)



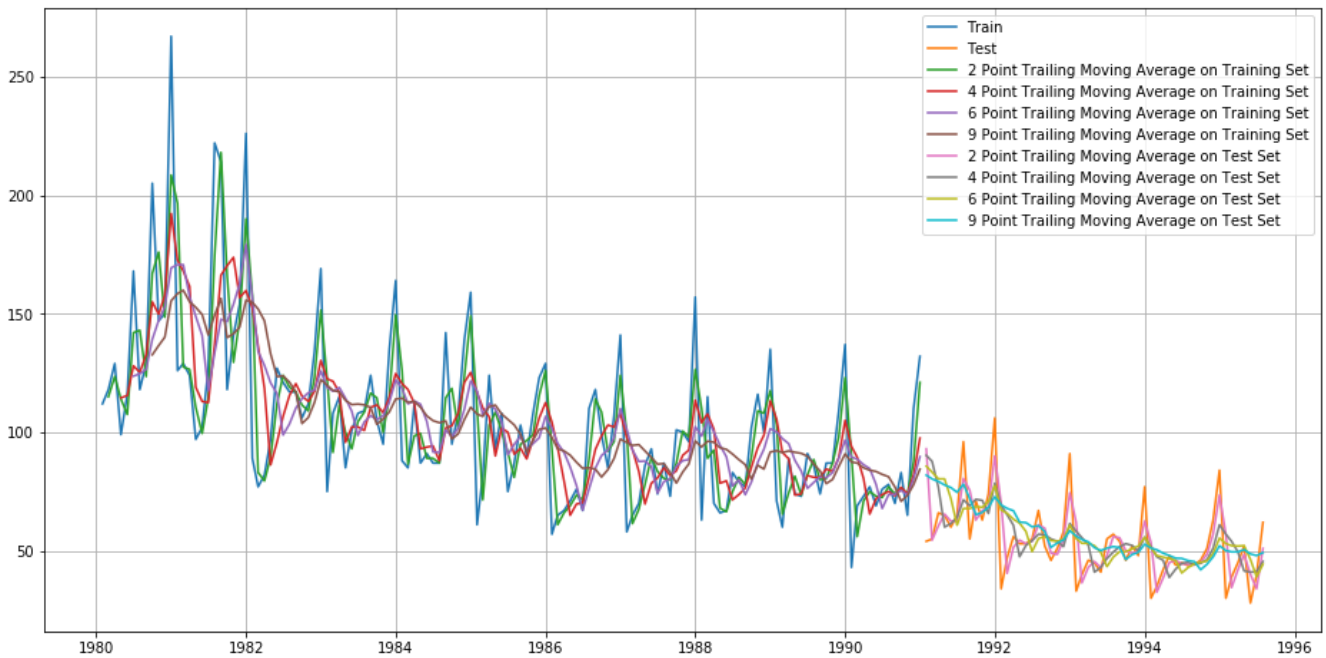
Creating train and test set(Rose)

Checking the top record of test set(Rose)

Out[116]:

Time_Stamp	Rose	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1991-01-31	54.0	93.0	90.25	85.666667	81.888889
1991-02-28	55.0	54.5	87.75	83.166667	80.333333
1991-03-31	66.0	60.5	76.75	80.333333	79.222222
1991-04-30	65.0	65.5	60.00	80.333333	77.777778
1991-05-31	60.0	62.5	61.50	72.000000	76.666667

Plotting on both the Training and Test data



Model Evaluation of various Moving averages(Test Data - RMSE and MAPE)

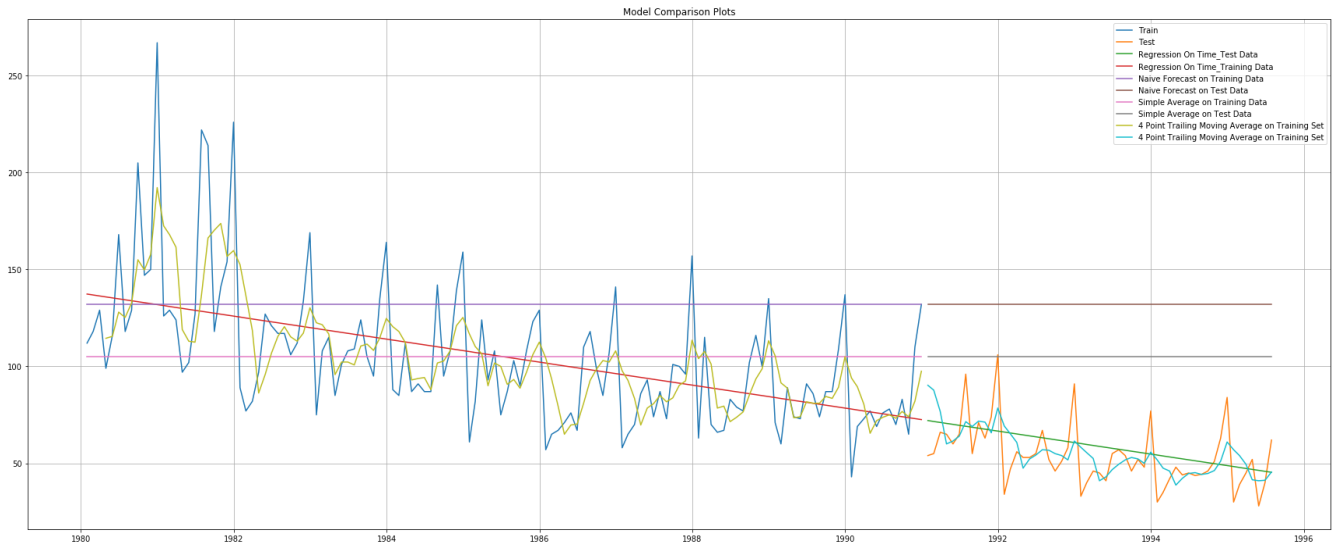
```
For Rose 2 point Moving Average Model forecast on the Test Data,  RMSE is 11.530 MAPE
is 13.60
For Rose 4 point Moving Average Model forecast on the Test Data,  RMSE is 14.462 MAPE
is 19.59
For Rose 6 point Moving Average Model forecast on the Test Data,  RMSE is 14.587 MAPE
is 20.83
For Rose 9 point Moving Average Model forecast on the Test Data,  RMSE is 14.740 MAPE
is 21.13
```

Storing the Moving average Result In Dataframe

Out[120]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13

Rose(plot all the models and compare the Time Series plots)



Method 5(Rose): Simple Exponential Smoothing

(Rose): SES autofit parameters

Out[126]:

```
{'smoothing_level': 0.09874989805931945,
 'smoothing_slope': nan,
 'smoothing_seasonal': nan,
 'damping_slope': nan,
 'initial_level': 134.38695595667537,
 'initial_slope': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

(Rose): SES Predicting Training top 5 autofit values

Out[127]:

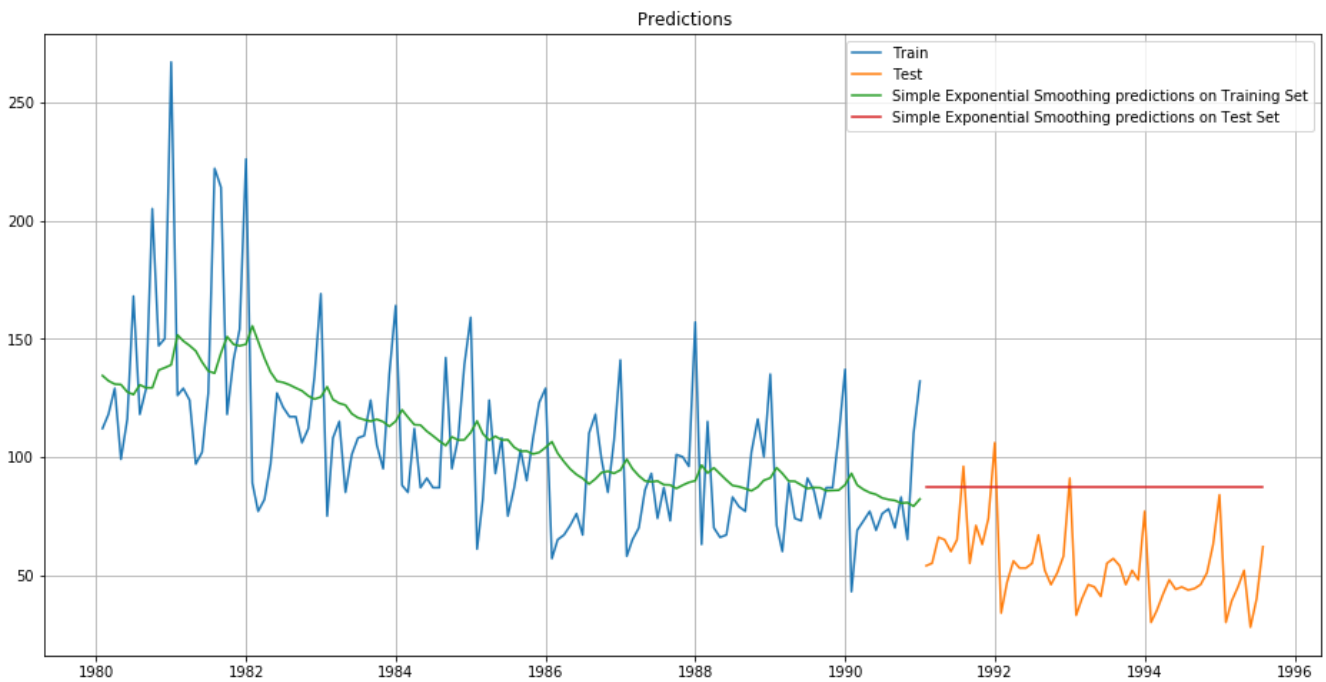
Time_Stamp	Rose	predict
1980-01-31	112.0	134.386956
1980-02-29	118.0	132.176246
1980-03-31	129.0	130.776343
1980-04-30	99.0	130.600930
1980-05-31	116.0	127.480341

(Rose): SES Predicting Test top 5 autofit values

Out[128]:

Time_Stamp	Rose	predict
1991-01-31	54.0	87.104999
1991-02-28	55.0	87.104999
1991-03-31	66.0	87.104999
1991-04-30	65.0	87.104999
1991-05-31	60.0	87.104999

Plotting on both the Training and Test data



#Model Evaluation Simple Exponential Smoothing(Training Data)

```
For  Simple Exponential Smoothing Model forecast on the Training Data,  RMSE is 31.501
MAPE is 22.73
```

Model Evaluation Simple Exponential Smoothing(Test Data)

```
For Rose Simple Exponential Smoothing Model forecast on the Test Data,  RMSE is 36.859
MAPE is 64.05
```

Storing the SimpleExponentinalSmoothing Result In Dataframe

Out[132]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13
Rose_SimpleExponentialSmoothing	36.858586	64.05

Define an empty dataframe

Out[133]:

Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

Model Evaluation at various Alpha Values

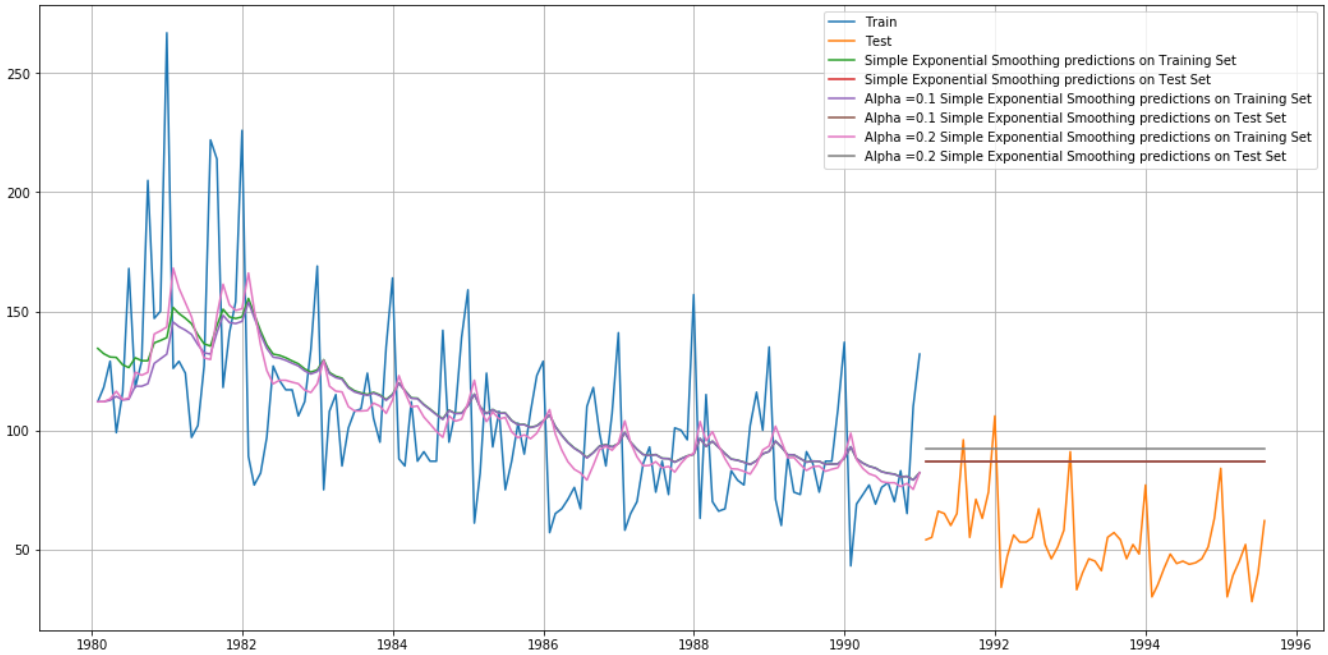
Out[135]:

	Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	31.815610	22.47	36.890375	64.10
1	0.2	31.979391	22.20	41.423877	72.39
2	0.3	32.470164	22.31	47.566302	83.89
3	0.4	33.035130	22.50	53.828369	95.70
4	0.5	33.682839	22.68	59.702301	107.02
5	0.6	34.441171	22.88	65.031433	117.26
6	0.7	35.323261	23.08	69.758009	126.30
7	0.8	36.334596	23.41	73.833601	134.06
8	0.9	37.482782	23.93	77.198702	140.46

Out[136]:

	Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	31.815610	22.47	36.890375	64.10
1	0.2	31.979391	22.20	41.423877	72.39
2	0.3	32.470164	22.31	47.566302	83.89
3	0.4	33.035130	22.50	53.828369	95.70
4	0.5	33.682839	22.68	59.702301	107.02

Plotting on both the Training and Test data



Storing the Simple Exponential Smoothing Result In Dataframe

Out[138]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13
Rose_SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.1,Rose_SimpleExponentialSmoothing	36.890375	64.10
Alpha=0.2,Rose_SimpleExponentialSmoothing	41.423877	72.39

(Rose) Double Exponential Smoothing (Holt's Model)

define an empty dataframe to store our values from the loop

Out[141]:

Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

tryng different values of alpha

Finding the Test RMSE at different values of alpha and Beta

Out[143]:

	Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	34.439111	24.83	36.987695	63.95
1	0.1	0.2	33.450729	24.45	48.757725	83.27
2	0.1	0.3	33.145789	24.46	78.228265	131.48
3	0.1	0.4	33.262191	24.68	99.655757	165.80
4	0.1	0.5	33.688415	24.92	124.342434	205.72
...	...	...	...	...	...	...
95	1.0	0.6	51.831610	34.67	801.752691	1325.29
96	1.0	0.7	54.497039	36.43	841.965081	1390.37
97	1.0	0.8	57.365879	38.27	854.038055	1409.91
98	1.0	0.9	60.474309	40.21	834.783438	1378.75
99	1.0	1.0	63.873454	42.49	780.152032	1290.33

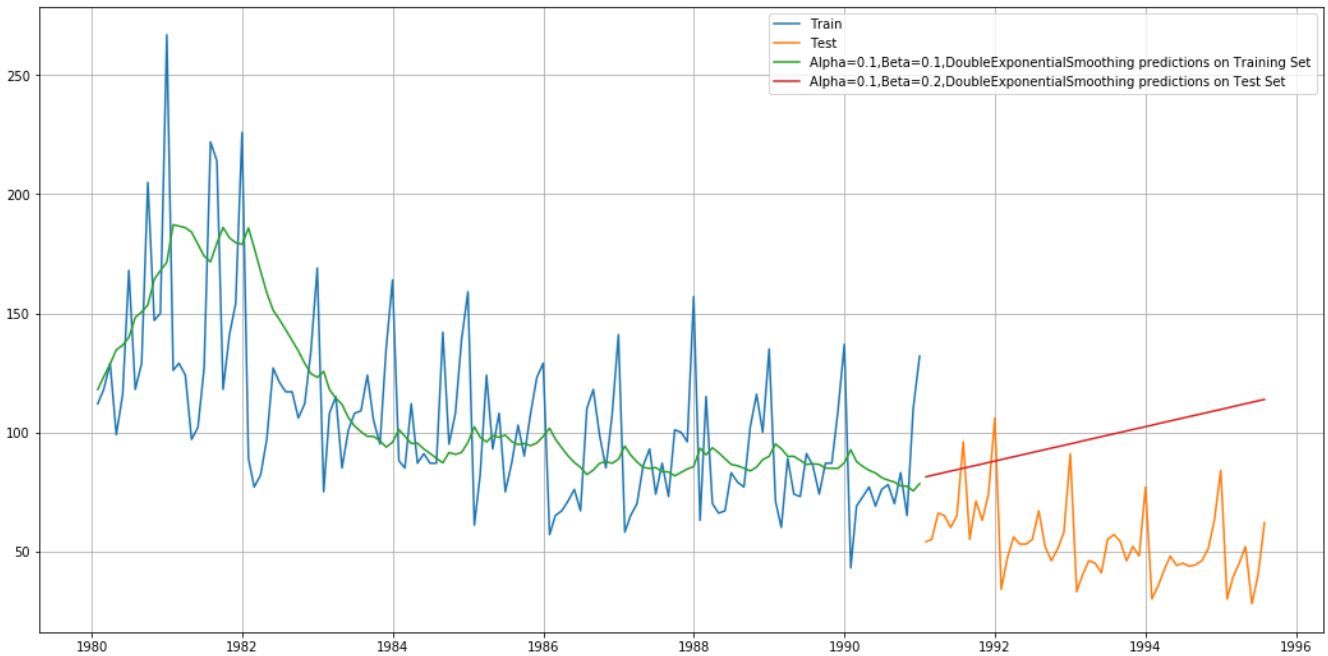
100 rows × 6 columns

Finding the Test RMSE in increasing order

Out[144]:

	Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	34.439111	24.83	36.987695	63.95
1	0.1	0.2	33.450729	24.45	48.757725	83.27
10	0.2	0.1	33.097427	23.69	65.800617	113.41
2	0.1	0.3	33.145789	24.46	78.228265	131.48
20	0.3	0.1	33.611269	23.80	98.723180	170.39

Plotting on both the Training and Test data



Storing the Double Exponential Smoothing Result In Dataframe

Out[146]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13
Rose_SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.1,Rose_SimpleExponentialSmoothing	36.890375	64.10
Alpha=0.2,Rose_SimpleExponentialSmoothing	41.423877	72.39
Alpha=0.1,Beta=0.1,Rose DoubleExponentialSmoothing	36.987695	63.95

Triple Exponential Smoothing (Holt - Winter's Model)

TES Autofit parameter (Rose)

Out[149]:

```
{'smoothing_level': 0.10609628211044227,
 'smoothing_slope': 0.04843853483296353,
 'smoothing_seasonal': 0.0,
 'damping_slope': nan,
 'initial_level': 76.65565233828687,
 'initial_slope': 0.0,
 'initial_seasons': array([1.47550285, 1.65927166, 1.80572675, 1.58888842, 1.77822733,
        1.92604398, 2.11649492, 2.25135237, 2.11690636, 2.08112883,
        2.40927327, 3.30448211]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

TES Prediction on the training data(Rose)

Out[150]:

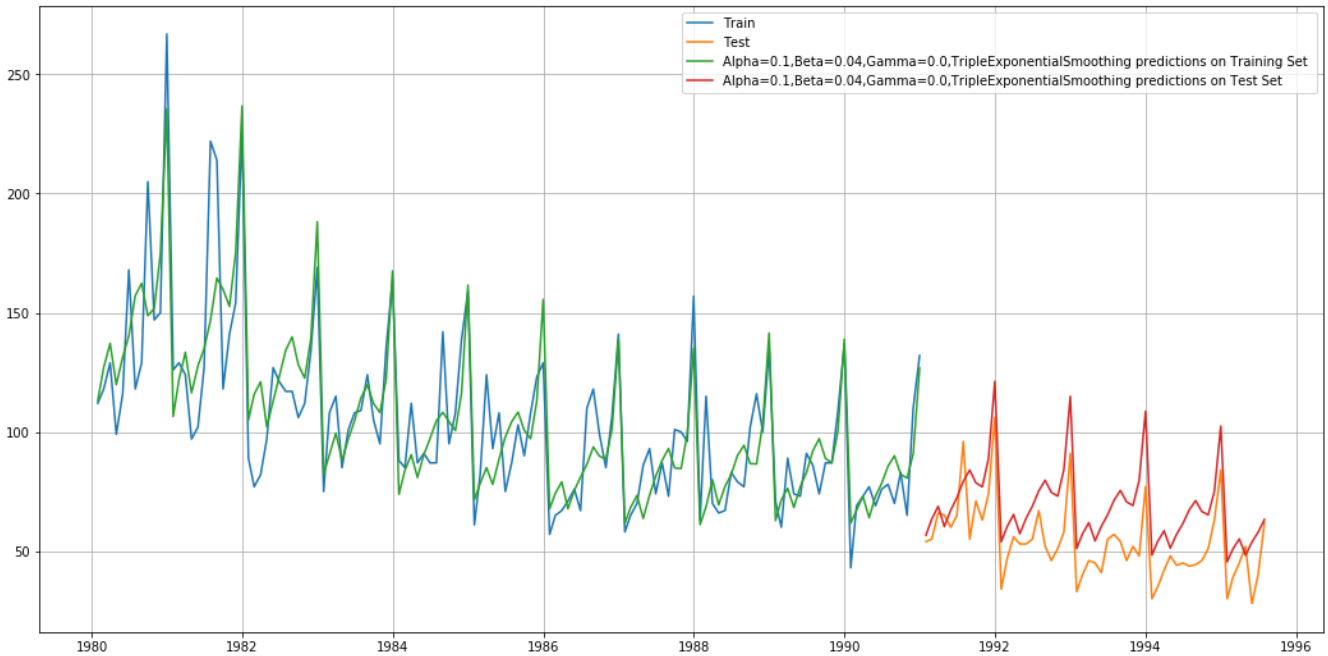
Time_Stamp	Rose	auto_predict
1980-01-31	112.0	113.105633
1980-02-29	118.0	127.054249
1980-03-31	129.0	137.165648
1980-04-30	99.0	119.844379
1980-05-31	116.0	131.432588

TES Prediction on the test data(Rose)

Out[151]:

Time_Stamp	Rose	auto_predict
1991-01-31	54.0	56.674334
1991-02-28	55.0	63.471272
1991-03-31	66.0	68.788792
1991-04-30	65.0	60.277823
1991-05-31	60.0	67.180378

Triple Exponential Smoothing(Plotting on both the Training and Test using autofit)



Triple Exponential Smoothing Model forecast on the Training Data

```
 Rose Triple Exponential Smoothing Model forecast on the Training Data,  RMSE is 18.57
9 MAPE is 13.21
```

Rose Triple Exponential Smoothing Model forecast on the Test Data

```
 Rose Triple Exponential Smoothing Model forecast on the Test Data,  RMSE is 17.445 MA
PE is 29.01
```

Storing the Triple Exponential Smoothing Result In Dataframe

Out[155]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13
Rose_SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.1,Rose_SimpleExponentialSmoothing	36.890375	64.10
Alpha=0.2,Rose_SimpleExponentialSmoothing	41.423877	72.39
Alpha=0.1,Beta=0.1,Rose DoubleExponentialSmoothing	36.987695	63.95
alpha=.1,beta=.4,gamma=0.0,Rose_TripleExponentialSmoothing	17.445169	29.01

Empty dataframe to store our values from the loop

Out[156]:

Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

find the Test RMSE at different values of alpha, beta and gamma

Out[158]:

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	0.1	25.216699	15.88	12.827131	20.28
1	0.1	0.1	0.2	24.057626	15.21	11.955885	18.76
2	0.1	0.1	0.3	23.645037	14.94	11.888892	18.47
3	0.1	0.1	0.4	23.570939	14.82	12.167489	18.87
4	0.1	0.1	0.5	23.718878	14.79	12.675513	19.64
...	...	...	...	...	...	...	...
995	1.0	1.0	0.6	28358.458519	3085.88	9603.641743	7820.66
996	1.0	1.0	0.7	30724.126331	4617.55	23029.955505	11848.08
997	1.0	1.0	0.8	1218.755446	493.75	9626.695041	8589.55
998	1.0	1.0	0.9	14150.253251	2303.41	9691.903861	7924.06
999	1.0	1.0	1.0	1768.254189	614.79	8138.609588	6817.82

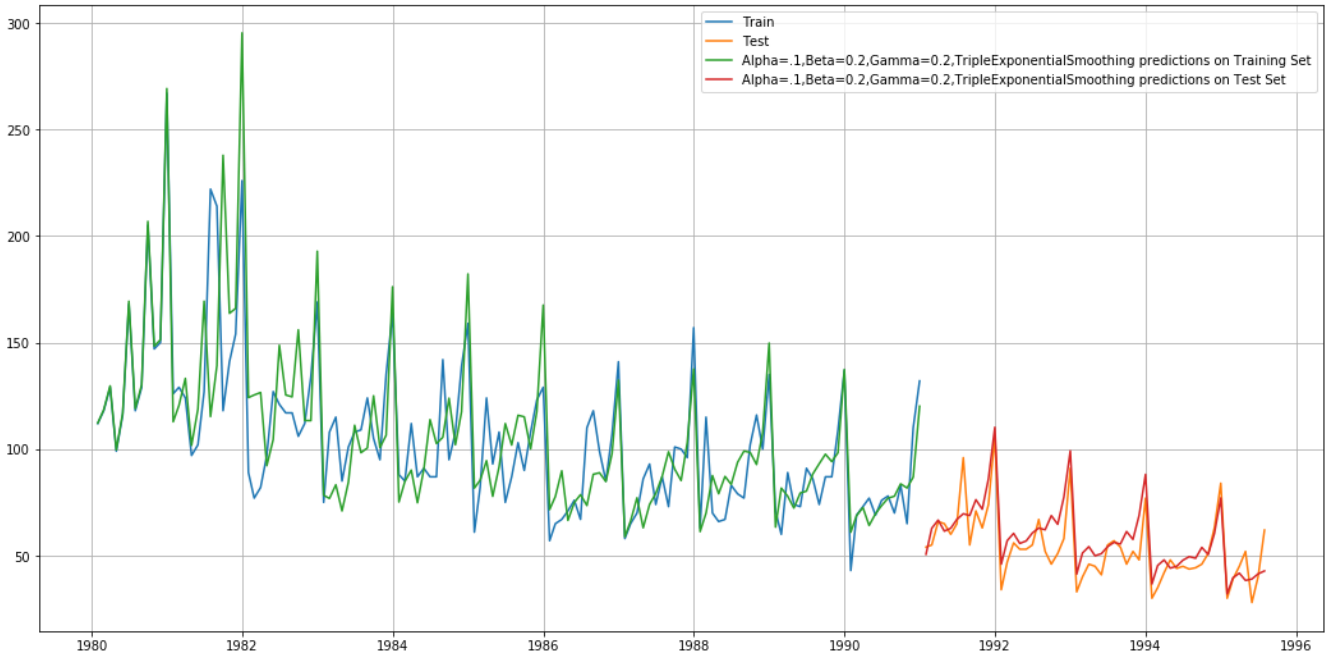
1000 rows × 7 columns

Sorting the Test RMSE in Ascending order

Out[159]:

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
11	0.1	0.2	0.2	24.365597	15.36	9.665739	14.08
10	0.1	0.2	0.1	25.529854	16.06	9.954136	14.50
12	0.1	0.2	0.3	23.969166	15.13	9.961876	14.33
142	0.2	0.5	0.3	27.631767	17.87	10.002930	14.25
151	0.2	0.6	0.2	28.289836	18.09	10.005753	13.53

Ploting the TripleExponentialSmoothing



Storing the TripleExponentialSmoothing In Dataframe

Out[161]:

	Test RMSE	Test MAPE
Rose_RegressionOnTime	15.291460	22.94
Rose_NaiveModel	79.778066	145.35
Rose_SimpleAverageModel	53.521557	95.13
Rose_2pointTrailingMovingAverage	11.530180	13.60
Rose_4pointTrailingMovingAverage	14.462330	19.59
Rose_6pointTrailingMovingAverage	14.586916	20.83
Rose_9pointTrailingMovingAverage	14.740112	21.13
Rose_SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.1,Rose_SimpleExponentialSmoothing	36.890375	64.10
Alpha=0.2,Rose_SimpleExponentialSmoothing	41.423877	72.39
Alpha=0.1,Beta=0.1,Rose DoubleExponentialSmoothing	36.987695	63.95
alpha=.1,beta=.4,gamma=0.0,Rose_TripleExponentialSmoothing	17.445169	29.01
Rose_Alpha=.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.665739	14.08

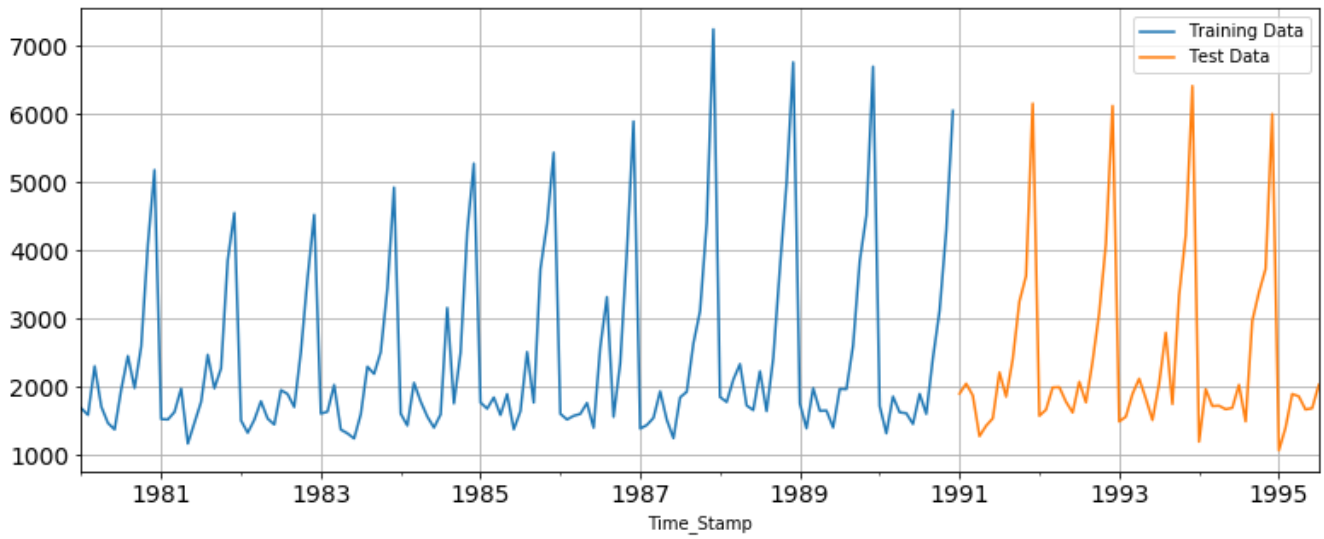
Test RMSE of various Models in acending order(Rose)

```
Sorted by RMSE values on the Test Data:
                                                      Test RMSE  Test MAPE
Rose_Alpha=.1,Beta=0.2,Gamma=0.2,TripleExponent...    9.665739      14.08
Rose_2pointTrailingMovingAverage                     11.530180      13.60
Rose_4pointTrailingMovingAverage                     14.462330      19.59
Rose_6pointTrailingMovingAverage                     14.586916      20.83
Rose_9pointTrailingMovingAverage                     14.740112      21.13
Rose_RegressionOnTime                                15.291460      22.94
alpha=.1,beta=.4,gamma=0.0,Rose_TripleExponenti...   17.445169      29.01
Rose_SimpleExponentialSmoothing                      36.858586      64.05
Alpha=0.1,Rose_SimpleExponentialSmoothing            36.890375      64.10
Alpha=0.1,Beta=0.1,Rose DoubleExponentialSmoothing   36.987695      63.95
Alpha=0.2,Rose_SimpleExponentialSmoothing            41.423877      72.39
Rose_SimpleAverageModel                              53.521557      95.13
Rose_NaiveModel                                      79.778066     145.35
```

Final Model Building (Rose Times Series) Comments

- have built several models and went through a model building exercise. This particular exercise has given us an idea as to which particular model gives us the least error on our test set for this data
- Alpha=.1,Beta=0.2,Gamma=0.2,TripleExponent Model is best Model up to now which gaves us the best accuracy (least RMSE)

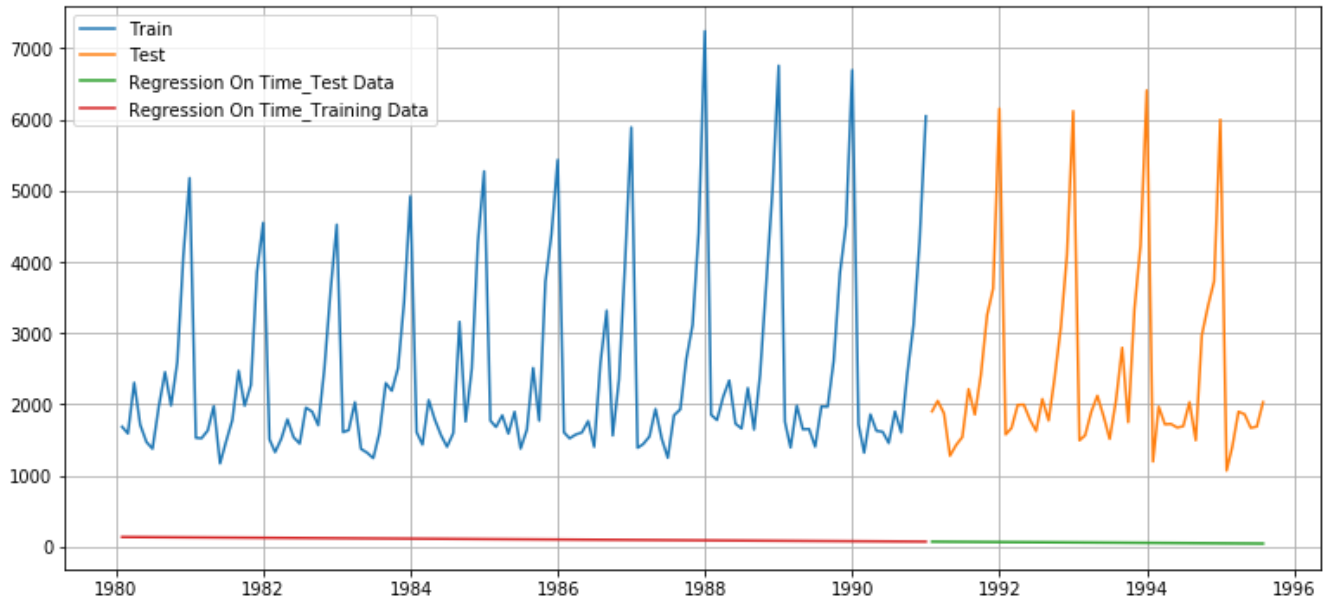
Defining the Linear Regression Model(Sparkling)



Checking the Training and Test Data(Sparkling)

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
4, 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, 6
7, 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, 10
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 12
5, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 1
67, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 18
4, 185, 186, 187]



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

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

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

Last few rows of Test Data
             Sparkling  time
Time_Stamp
1995-03-31       1897   183
1995-04-30       1862   184
1995-05-31       1670   185
1995-06-30       1688   186
1995-07-31       2031   187
```

Plotting Linear Regression Train and Test data(Sparkling)



Defining the functions for calculating the accuracy metrics

Mean Absolute Percentage Error - Function Definition

RegressionOnTime forecast on the Training Data(Sparkling)

```
For RegressionOnTime forecast on the Training Data,  RMSE is 2641.866 MAPE is 95.63
```

RegressionOnTime forecast on the Test Data(Sparkling)

```
For RegressionOnTime forecast on the Test Data,  RMSE is 2665.351 MAPE is 97.55
```

Storing the Sparkling Regression Result In Dataframe

Out[176]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55

#Model 2(Sparkling): Naive Approach: $\hat{y}\ t$+1=yt

Defining the Naive Model(Sparkling)

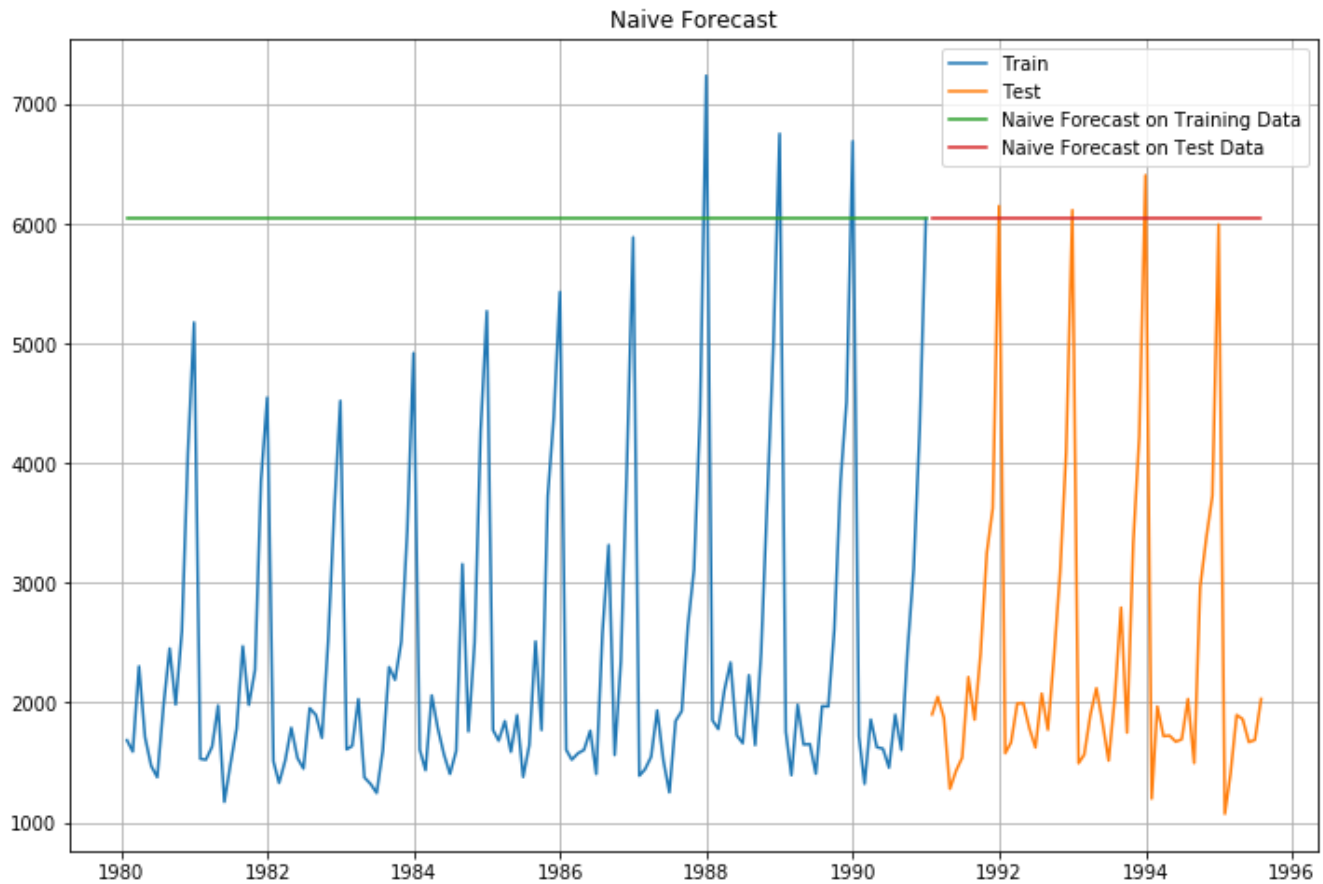
Let us get the last value (or the most recent value) of the training set.

Out[178]:
```
Time_Stamp
1980-01-31    6047
1980-02-29    6047
1980-03-31    6047
1980-04-30    6047
1980-05-31    6047
Name: naive, dtype: int64
```

Let us get the last value (or the most recent value) of the Test set.

Out[179]:
```
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
```

Naive Forecast on Train and Test Data(Sparkling)



SparklingModel Evaluation(Training Data - RMSE and MAPE)

```
For Naive Model forecast on the Training Data,  RMSE is 3867.701 MAPE is 153.17
```

Sparkling Model Evaluation(Test Data - RMSE and MAPE)

```
For RegressionOnTime forecast on the Test Data,  RMSE is 3864.279 MAPE is 152.87
```

Storing the Naive Result In Dataframe

Out[183]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87

Method 3(Sparkling): Simple Average

Defining the Simple Average Train Model(Sparkling)

Out[185]:

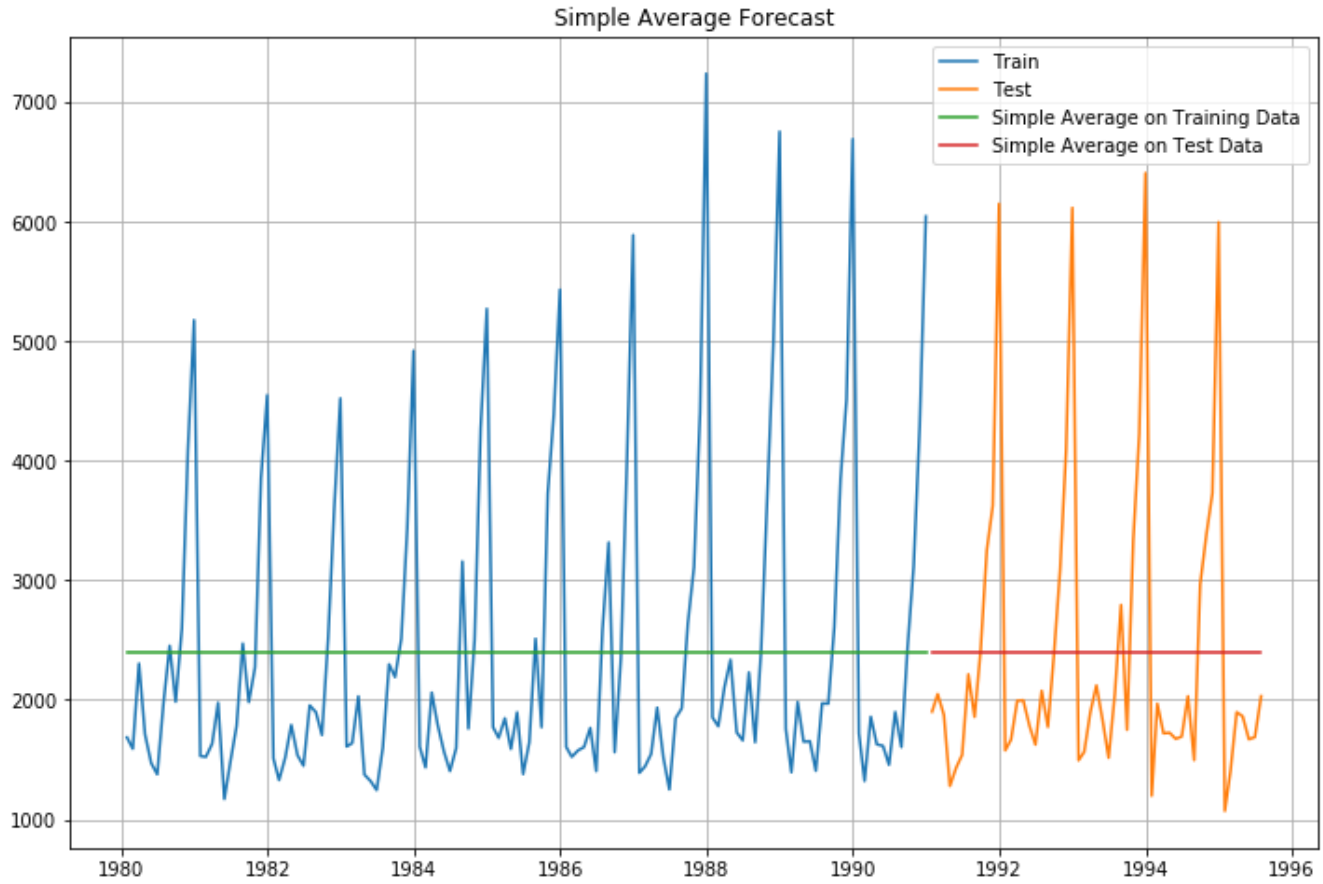
Time_Stamp	Sparkling	mean_forecast
1980-01-31	1686	2403.780303
1980-02-29	1591	2403.780303
1980-03-31	2304	2403.780303
1980-04-30	1712	2403.780303
1980-05-31	1471	2403.780303

Defining the Simple Average Test Model(Sparkling)

Out[186]:

Time_Stamp	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

Plot on Simple Average on Training and Test Data(Sparkling)



Model Evaluation(Training Data - RMSE and MAPE)

```
For Simple Average Model forecast on the Training Data,  RMSE is 1298.484 MAPE is 40.3
6
```

Model Evaluation(Test Data - RMSE and MAPE)

```
For Simple Average forecast on the Test Data,  RMSE is 1275.082 MAPE is 38.90
```

Storing the Sparkling Simple Moving Average Result In Dataframe

Out[190]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90

Sparkling (Defining the Moveing Average)

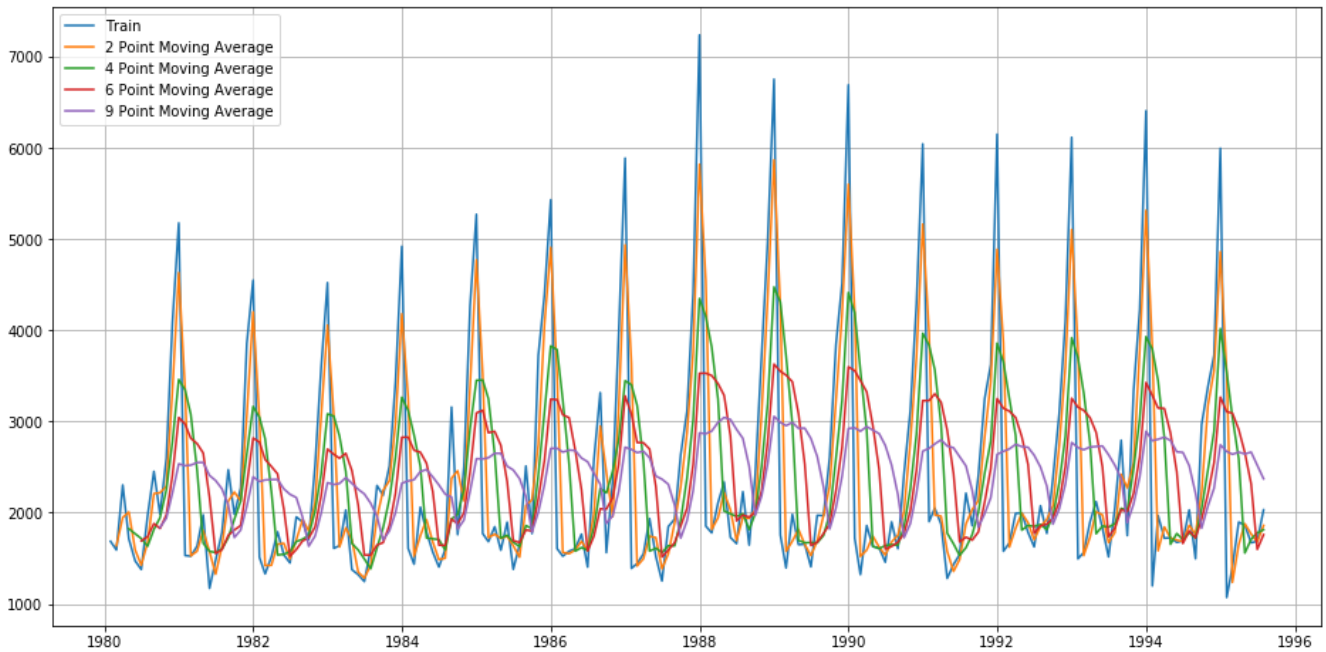
Out[191]:

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

Out[192]:

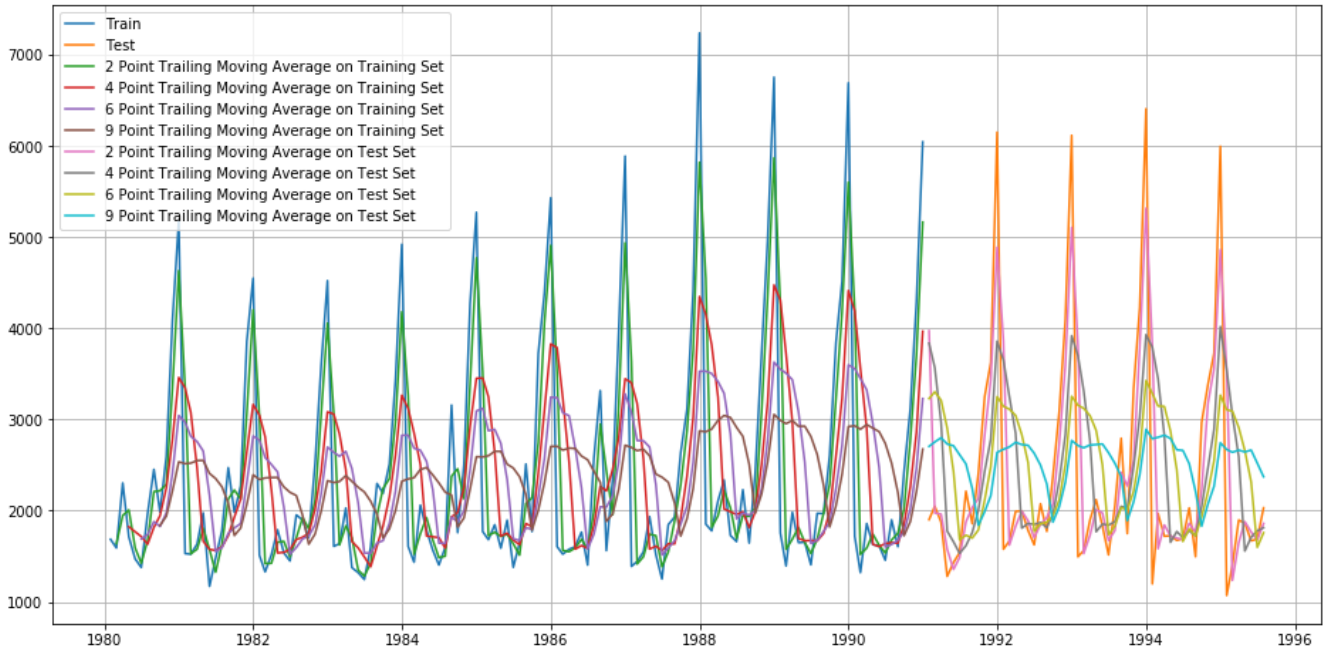
Time_Stamp	Sparkling	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-31	1686	NaN	NaN	NaN	NaN
1980-02-29	1591	1638.5	NaN	NaN	NaN
1980-03-31	2304	1947.5	NaN	NaN	NaN
1980-04-30	1712	2008.0	1823.25	NaN	NaN
1980-05-31	1471	1591.5	1769.50	NaN	NaN

Plotting on the whole data(Sparkling)



Creating train and test set(Sparkling)

Plotting on both the Training and Test data)



Model Evaluation of various Moving averages(Test Data - RMSE and MAPE)

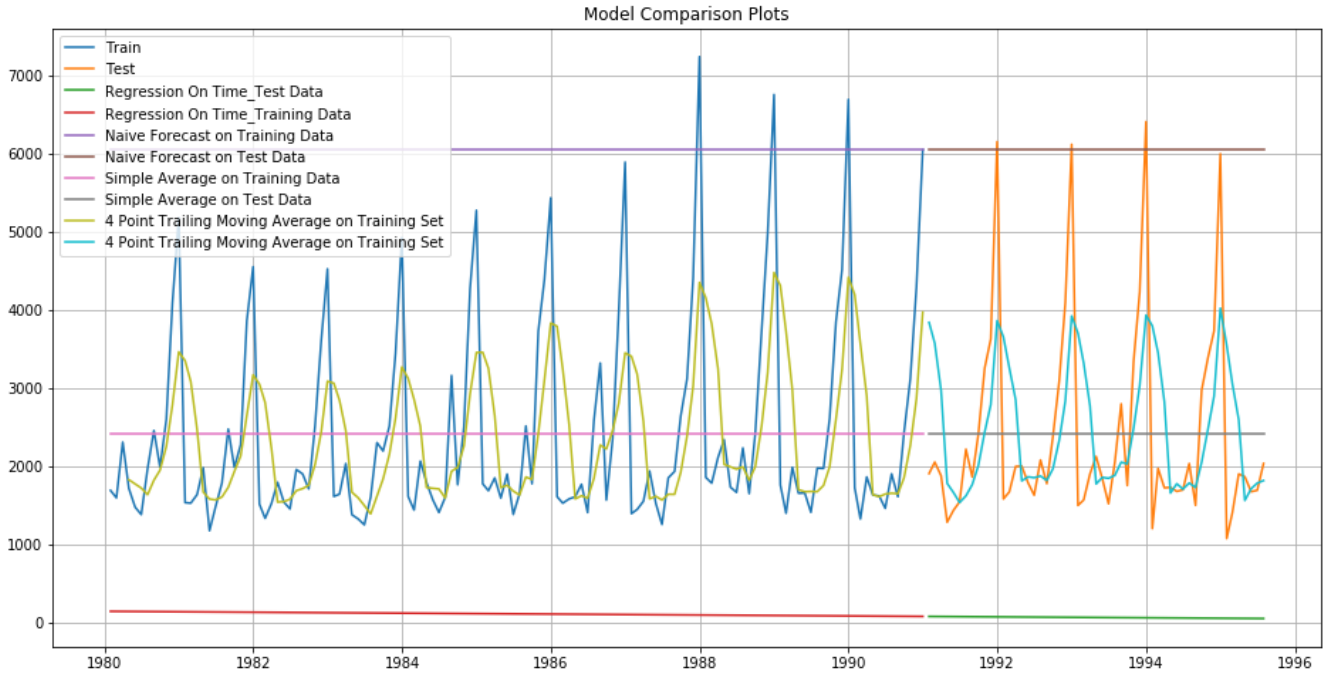
```
For 2 point Moving Average Model forecast on the Test Data,  RMSE is 813.401 MAPE is 1
9.70
For 4 point Moving Average Model forecast on the Test Data,  RMSE is 1156.590 MAPE is
35.96
For 6 point Moving Average Model forecast on the Test Data,  RMSE is 1283.927 MAPE is
43.86
For 9 point Moving Average Model forecast on the Test Data,  RMSE is 1346.278 MAPE is
46.86
```

Storing the Moving average Result In Dataframe

Out[197]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86

Sparkling(plot all the models and compare the Time Series plots)



Method 5(Sparkling): Simple Exponential Smoothing

(Sparkling): SES autofit parameters

Out[203]:
```
{'smoothing_level': 0.0,
 'smoothing_slope': nan,
 'smoothing_seasonal': nan,
 'damping_slope': nan,
 'initial_level': 2403.785621547663,
 'initial_slope': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

(Sparkling): SES Predicting Training top 5 autofit values

Out[204]:

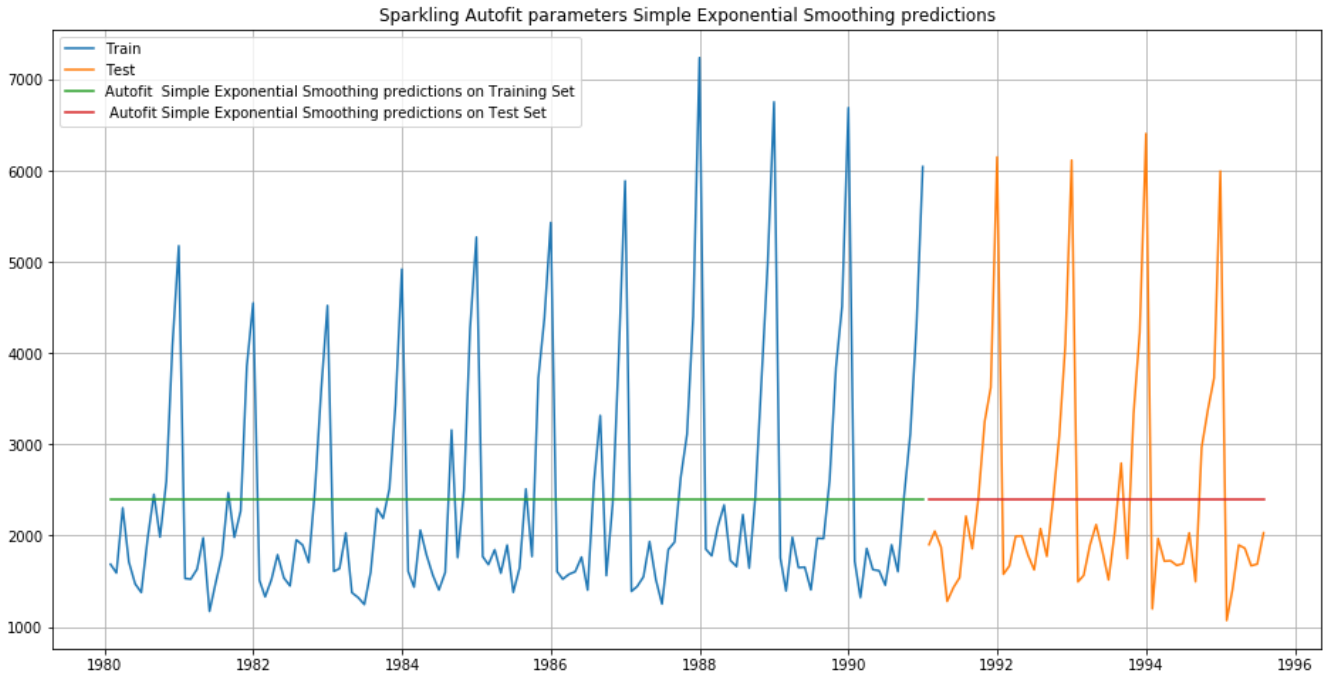
Time_Stamp	Sparkling	predict
1980-01-31	1686	2403.785622
1980-02-29	1591	2403.785622
1980-03-31	2304	2403.785622
1980-04-30	1712	2403.785622
1980-05-31	1471	2403.785622

(Sparkling): SES Predicting Test top 5 autofit values

Out[206]:

Time_Stamp	Sparkling	predict
1991-01-31	1902	2403.785622
1991-02-28	2049	2403.785622
1991-03-31	1874	2403.785622
1991-04-30	1279	2403.785622
1991-05-31	1432	2403.785622

Plotting on both the Training and Test data(Sparkling)



Sparkling Model Evaluation Simple Exponential Smoothing(Training Data)

```
For Sparkling Simple Exponential Smoothing Model forecast on the Training Data,  RMSE
is 1298.484 MAPE is 40.36
```

Model Evaluation Simple Exponential Smoothing(Test Data)

```
For Sparkling Simple Exponential Smoothing Model forecast on the Training Data,  RMSE
is 1275.082 MAPE is 38.90
```

Storing the SimpleExponentinalSmoothing Result In Dataframe

Out[210]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86
Sparkling_SimpleExponentialSmoothing	1275.081823	38.90

Define an empty dataframe

Out[211]:

Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

Model Evaluation at various Alpha Values(Sparkling)

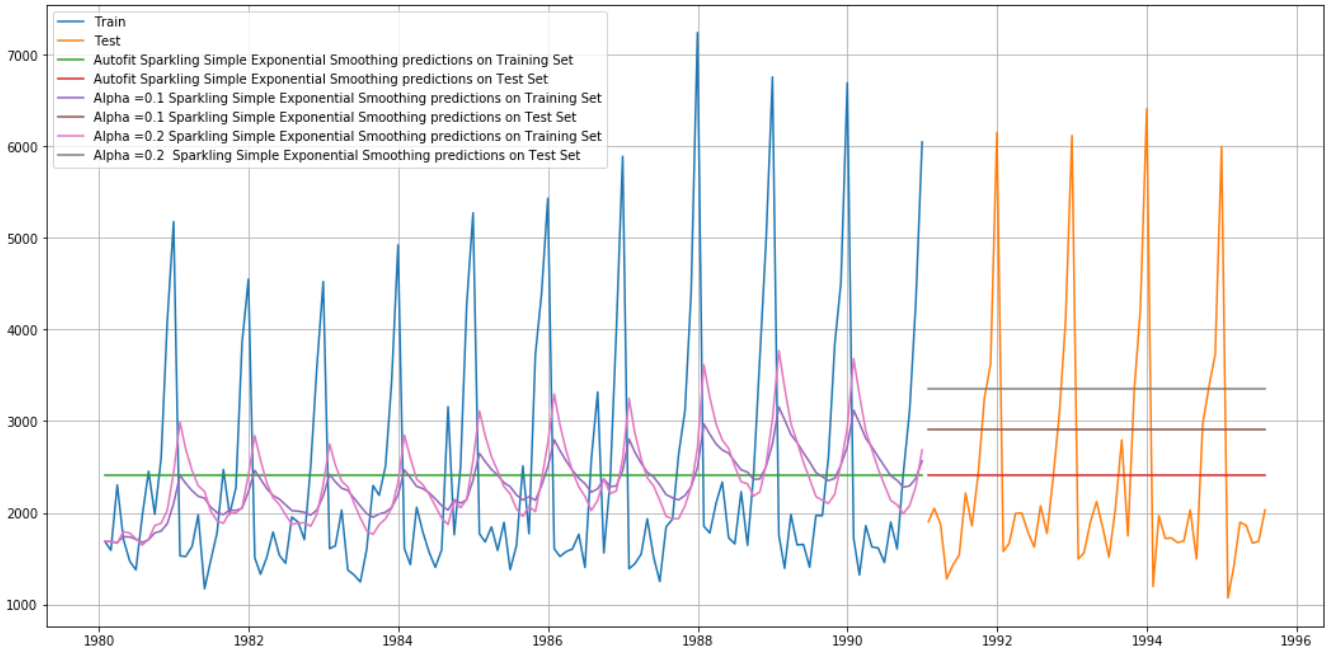
Out[213]:

	Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	1333.873836	42.03	1375.393398	49.53
1	0.2	1356.042987	43.79	1595.206839	60.46
2	0.3	1359.511747	43.73	1935.507132	75.66
3	0.4	1352.588879	42.75	2311.919615	91.55
4	0.5	1344.004369	41.16	2666.351413	106.27
5	0.6	1338.805381	39.80	2979.204388	118.77
6	0.7	1338.844308	38.55	3249.944092	129.34
7	0.8	1344.462091	37.60	3483.801006	138.34
8	0.9	1355.723518	36.79	3686.794285	146.08

Out[214]:

	Alpha Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	1333.873836	42.03	1375.393398	49.53
1	0.2	1356.042987	43.79	1595.206839	60.46
2	0.3	1359.511747	43.73	1935.507132	75.66
3	0.4	1352.588879	42.75	2311.919615	91.55
4	0.5	1344.004369	41.16	2666.351413	106.27

Plotting on both the Training and Test data(Sparkling)



Storing the Simple Exponential Smoothing Result In Dataframe

Out[216]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86
Sparkling_SimpleExponentialSmoothing	1275.081823	38.90
Alpha =0.1 Sparkling Simple Exponential Smoothing predictions	1375.393398	49.53
Alpha =0.2 Sparkling Simple Exponential Smoothing predictions	1595.206839	60.46

(Sparkling) Double Exponential Smoothing (Holt's Model)

define an empty dataframe to store our values from the loop

Out[220]:

Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

Tryng different values of Alpha and Beta(Sparkling)

Storing the Double Exponential Smoothing Result In Dataframe(Sparkling)

Out[222]:

	Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	1382.520870	44.37	1778.564670	67.20
1	0.1	0.2	1413.598835	46.14	2599.439986	95.43
2	0.1	0.3	1445.762015	47.43	4293.084674	155.43
3	0.1	0.4	1480.897776	48.79	6039.537339	218.98
4	0.1	0.5	1521.108657	50.34	7390.522201	267.94
...	...	...	...	...	...	...
95	1.0	0.6	1753.402326	45.75	49327.087977	1824.42
96	1.0	0.7	1825.187155	47.97	52655.765663	1945.23
97	1.0	0.8	1902.013709	50.11	55442.273880	2046.36
98	1.0	0.9	1985.368445	52.03	57823.177011	2132.75
99	1.0	1.0	2077.672157	53.81	59877.076519	2207.28

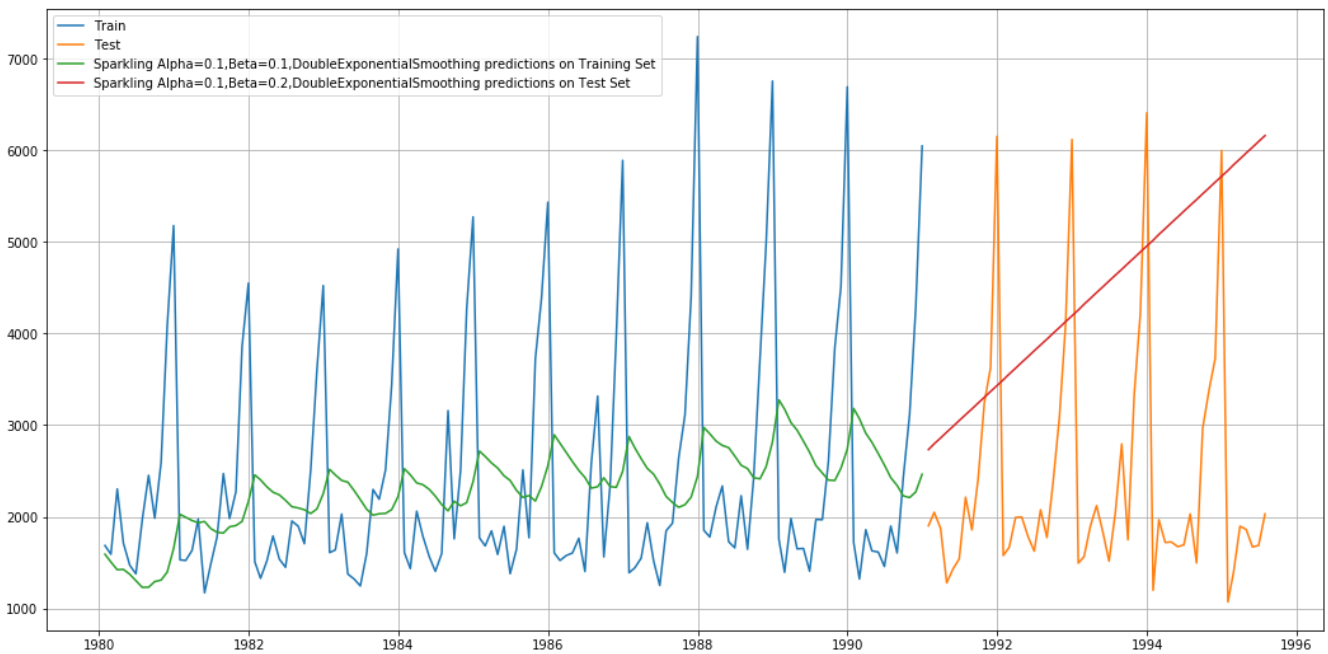
100 rows × 6 columns

Finding the Test RMSE in increasing order(Sparkling)

Out[223]:

	Alpha Values	Beta Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	1382.520870	44.37	1778.564670	67.20
1	0.1	0.2	1413.598835	46.14	2599.439986	95.43
10	0.2	0.1	1418.041591	46.65	3611.763322	135.41
2	0.1	0.3	1445.762015	47.43	4293.084674	155.43
20	0.3	0.1	1431.169601	46.99	5908.185554	223.50

Plotting on both the Training and Test data(Sparkling)



Storing the Double Exponential Smoothing Result In Dataframe

Out[225]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86
Sparkling_SimpleExponentialSmoothing	1275.081823	38.90
Sparkling Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670	67.20

Triple Exponential Smoothing (Holt - Winter's Model)

TES Autofit parameter (Sparkling)

Out[230]:

```
{'smoothing_level': 0.08622052864340005,
 'smoothing_slope': 1.7097923242790787e-09,
 'smoothing_seasonal': 0.476348411283582,
 'damping_slope': nan,
 'initial_level': 1685.0330927463103,
 'initial_slope': 3.268126463343508e-09,
 'initial_seasons': array([  38.53100271,  -37.57689912,  464.75420339,  205.89223136,
        -140.66189258, -156.89672956,  338.19195606,  856.38589201,
         402.98286327,  971.15758553, 2401.41729185, 3426.30639716]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

TES Prediction on the training data(Sparkling)

Out[231]:

Time_Stamp	Sparkling	auto_predict
1980-01-31	1686	1723.564095
1980-02-29	1591	1644.217397
1980-03-31	2304	2141.960068
1980-04-30	1712	1897.069264
1980-05-31	1471	1534.558371

Out[232]:

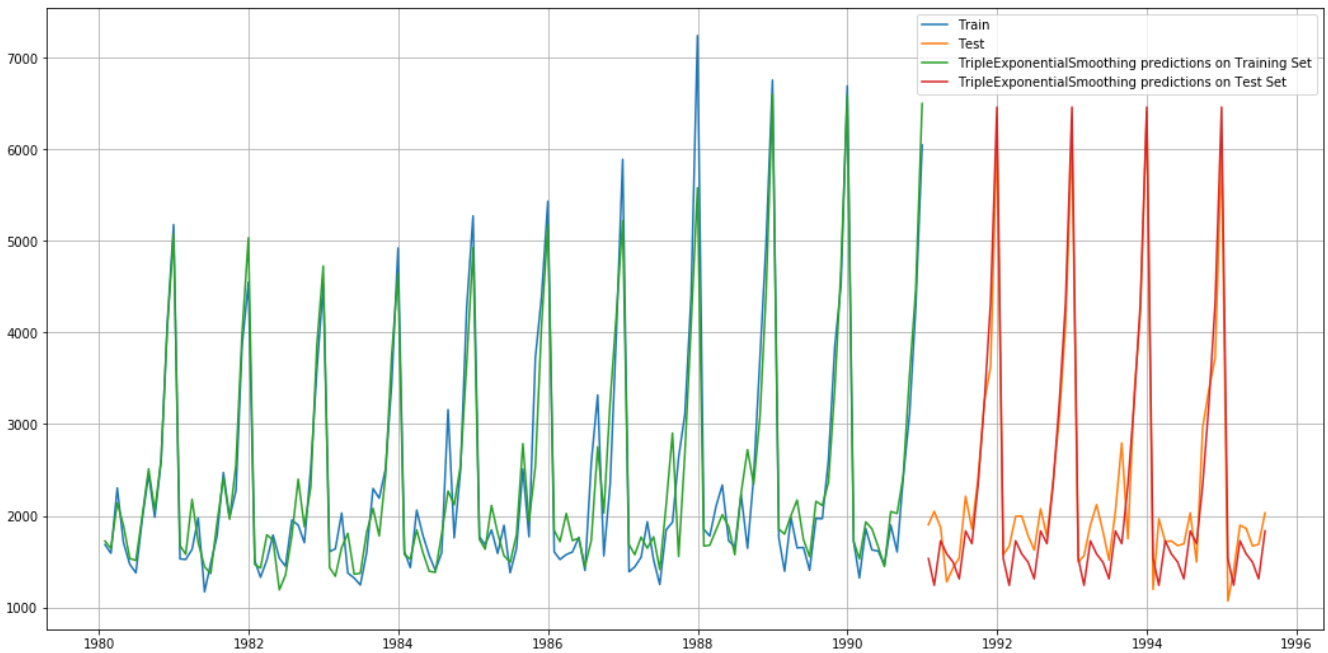
Time_Stamp	Sparkling	auto_predict
1980-01-31	1686	1723.564095
1980-02-29	1591	1644.217397
1980-03-31	2304	2141.960068
1980-04-30	1712	1897.069264
1980-05-31	1471	1534.558371

TES Prediction on the test data(Sparkling)

Out[233]:

Time_Stamp	Sparkling	auto_predict
1991-01-31	1902	1532.368807
1991-02-28	2049	1241.329822
1991-03-31	1874	1726.719960
1991-04-30	1279	1584.260962
1991-05-31	1432	1493.955208

Triple Exponential Smoothing(Plotting on both the Training and Test using autofit)



TES Model Evaluation on training Data(Using Auto Fit parameter)

```
Sparkling Autofit parameters Triple Exponential Smoothing Model forecast on the Traini
ng Data,  RMSE is 371.264 MAPE is 10.90
```

TES Model Evaluation on test Data(Using Auto Fit parameter)

```
 Sparkling Triple Exponential Smoothing Model forecast on the Test Data,  RMSE is 362.
792 MAPE is 12.08
```

Storing the TES Result In Dataframe

Out[237]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86
Sparkling_SimpleExponentialSmoothing	1275.081823	38.90
SparklingAutofitTripleExponentialSmoothing	362.791906	12.08

Define an empty dataframe to store our values(Sparkling)

Out[238]:

Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE

Finding the Test RMSE at different values of alpha, Beta and gamma(Sparkling)

Storing the Triple Exponential Smoothing Result In Dataframe(Sparkling)

Out[240]:

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
0	0.1	0.1	0.1	4.250948e+02	12.94	5.200117e+02	18.27
1	0.1	0.1	0.2	4.015222e+02	12.10	6.398682e+02	22.90
2	0.1	0.1	0.3	3.873034e+02	11.70	7.427758e+02	26.92
3	0.1	0.1	0.4	3.802781e+02	11.41	8.077458e+02	29.33
4	0.1	0.1	0.5	3.787107e+02	11.21	8.409400e+02	30.51
...	...	...	...	...	...	...	...
995	1.0	1.0	0.6	1.120481e+04	273.11	8.969735e+05	32683.32
996	1.0	1.0	0.7	4.737690e+04	942.03	2.479826e+06	90131.59
997	1.0	1.0	0.8	1.848687e+05	3284.35	6.299937e+06	228497.85
998	1.0	1.0	0.9	6.268726e+05	10732.44	2.202785e+07	799469.61
999	1.0	1.0	1.0	1.866231e+06	30835.34	8.971517e+07	3262039.00

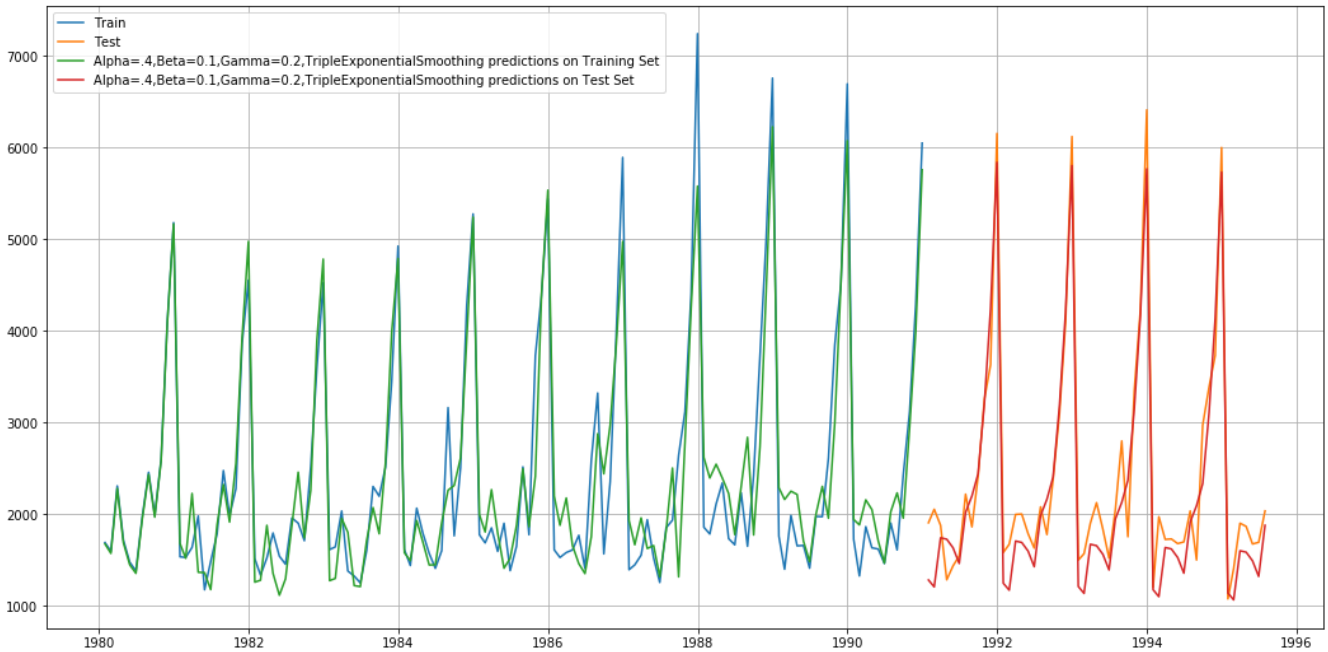
1000 rows × 7 columns

Checking Test RMSE in Ascending order(Sparkling)

Out[241]:

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Train MAPE	Test RMSE	Test MAPE
85	0.1	0.9	0.6	438.525019	13.46	338.458417	9.93
139	0.2	0.4	1.0	483.673805	14.39	338.840640	10.59
110	0.2	0.2	0.1	452.507225	13.62	343.715412	11.06
30	0.1	0.4	0.1	447.900255	13.12	346.302722	11.50
301	0.4	0.1	0.2	441.242988	13.11	365.192054	12.45

Plotting on both the Training and Test data((Sparkling)



Storing the Triple Exponential Smoothing Result In Dataframe(Sparkling)

Out[243]:

	Test RMSE	Test MAPE
RegressionOnTime	2665.351076	97.55
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86
Sparkling_SimpleExponentialSmoothing	1275.081823	38.90
SparklingAutofitTripleExponentialSmoothing	362.791906	12.08
Alpha=.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	338.458417	9.93

Storing the Test RMSE in Ascending Order

```
Sorted by RMSE values on the Test Data:
                                                      Test RMSE  Test MAPE
Alpha=.4,Beta=0.1,Gamma=0.2,TripleExponentialSm...   338.458417       9.93
SparklingAutofitTripleExponentialSmoothing           362.791906      12.08
2pointTrailingMovingAverage                          813.400684      19.70
4pointTrailingMovingAverage                         1156.589694      35.96
SimpleAverageModel                                  1275.081804      38.90
Sparkling_SimpleExponentialSmoothing                1275.081823      38.90
6pointTrailingMovingAverage                         1283.927428      43.86
9pointTrailingMovingAverage                         1346.278315      46.86
RegressionOnTime                                    2665.351076      97.55
NaiveModel                                          3864.279352     152.87
```

Final Model Building (Sparkling Times Series) Comments

- have built several models and went through a model building exercise. This particular exercise has given us an idea as to which particular model gives us the least error on our test set for this data
- Alpha=.4,Beta=0.1,Gamma=0.2 TripleExponential Model is bests Model up to now which gaves us the best accuracy (least RMSE)

Question 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 s
tationary. Check the new data for stationarity and comment.Note: Stationarity s
hould be checked at alpha = 0.05
```

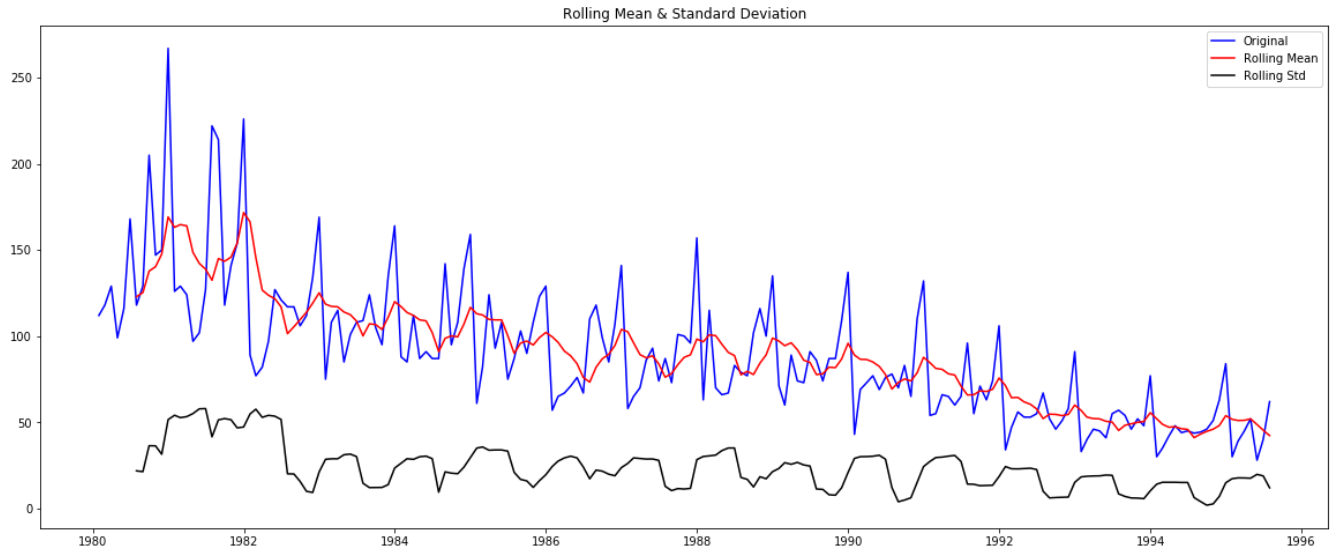
Checking for Stationarity(Rose)

Define the Stationarity Function

Hypothesis Test (Rose)

- Null Hypothesis Test (H0):- Series is not stationary
- Alternative Hypothesis Test (H1):- Series is stationary

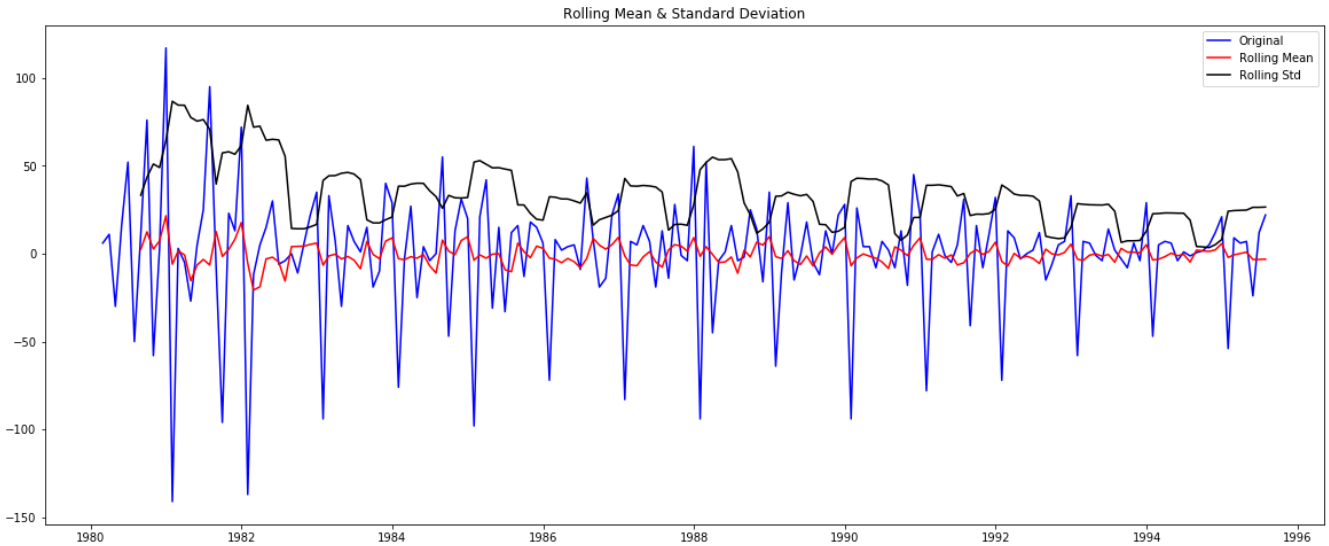
Checking the Stationarity (Rose)




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.873273
p-value                          0.344737
#Lags Used                      13.000000
Number of Observations Used    173.000000
Critical Value (1%)             -3.468726
Critical Value (5%)             -2.878396
Critical Value (10%)            -2.575756
dtype: float64
```

Considering the p value , At Present Series is not stationarity (Rose)

Make the Series Stationarity



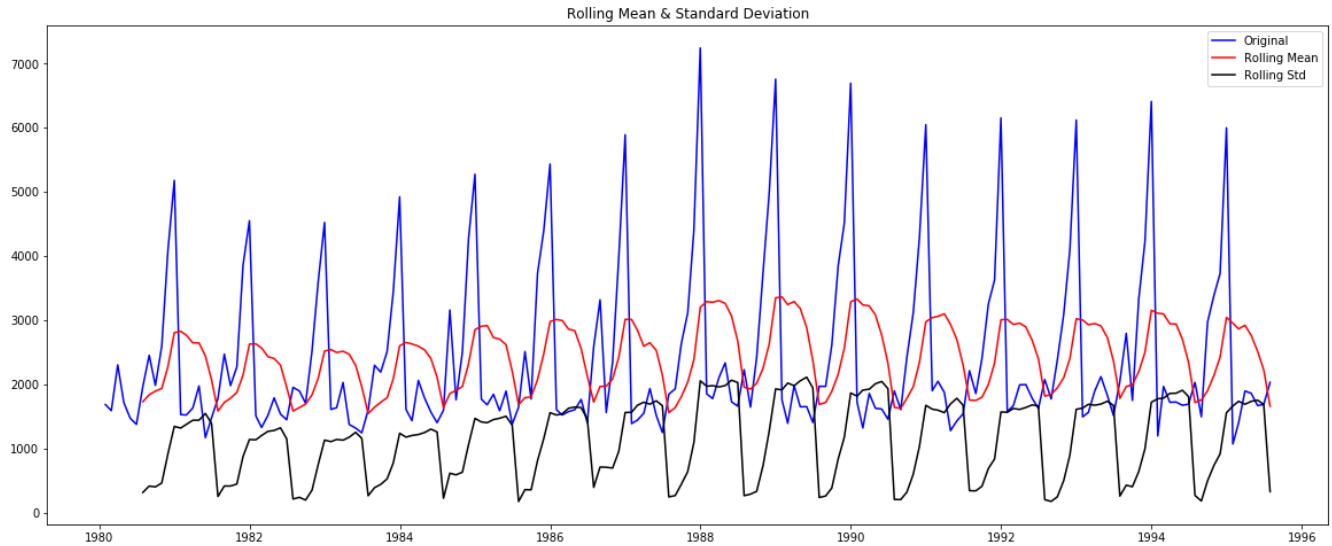

```
Results of Dickey-Fuller Test:
Test Statistic                  -8.043385e+00
p-value                          1.821604e-12
#Lags Used                       1.200000e+01
Number of Observations Used      1.730000e+02
Critical Value (1%)             -3.468726e+00
Critical Value (5%)             -2.878396e+00
Critical Value (10%)            -2.575756e+00
dtype: float64
```

Considering the p value , Now Series is stationary (Rose) Since p value is less that .05

Checking for Stationarity(Sparkling)

Hypothesis Test (Sparkling)

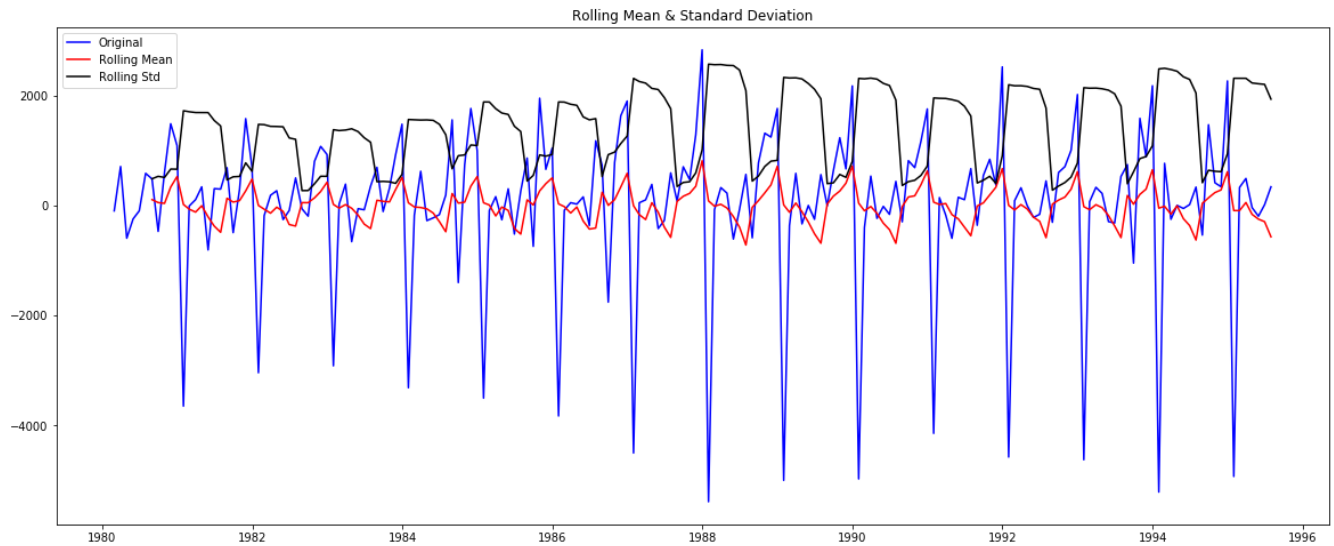
- Null Hypothesis Test (H0):- Series is not stationary
- Alternative Hypothesis Test (H1):- Series is stationary




```
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
```

Considering the p value , At Present Series is not stationarity (Sparkling)

Make the Series Stationarity(Sparkling)




```
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
```

Considering the p value , Now Series is stationarity (Sparkling) Since p value is less that .05

Question6:- Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest

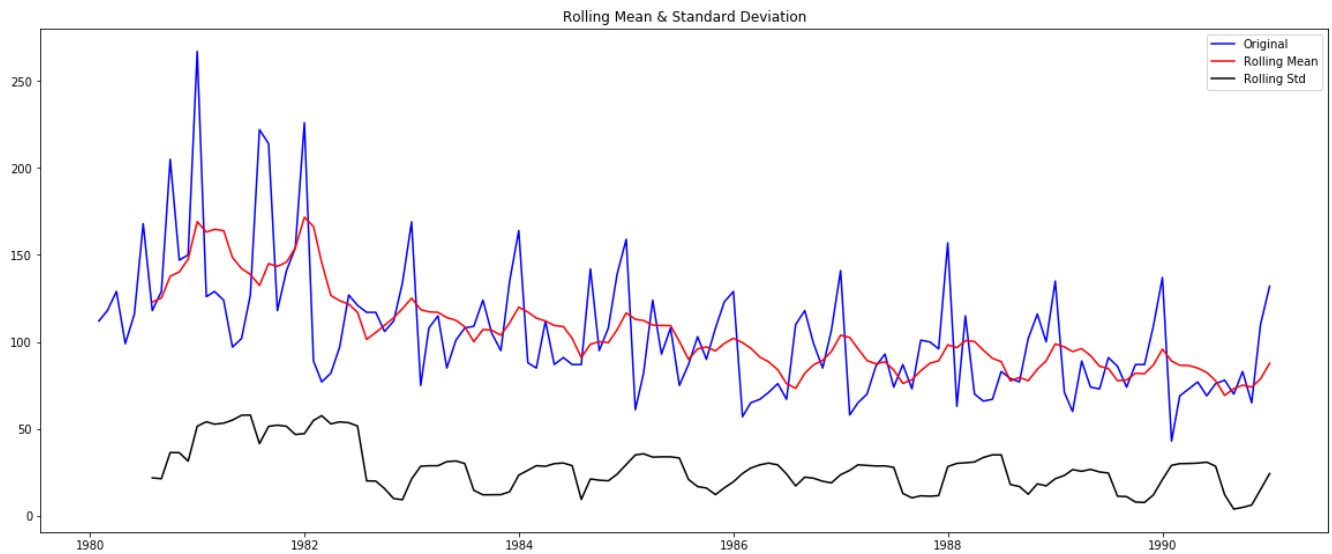
- Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE.

Need to check the stationarity of Training Set (Rose) before Building the Arima and Sarima Models

Hypothesis Test (Rose Training Set)

- Null Hypothesis Test (H0):- Series is not stationary
- Alternative Hypothesis Test (H1):- Series is stationary

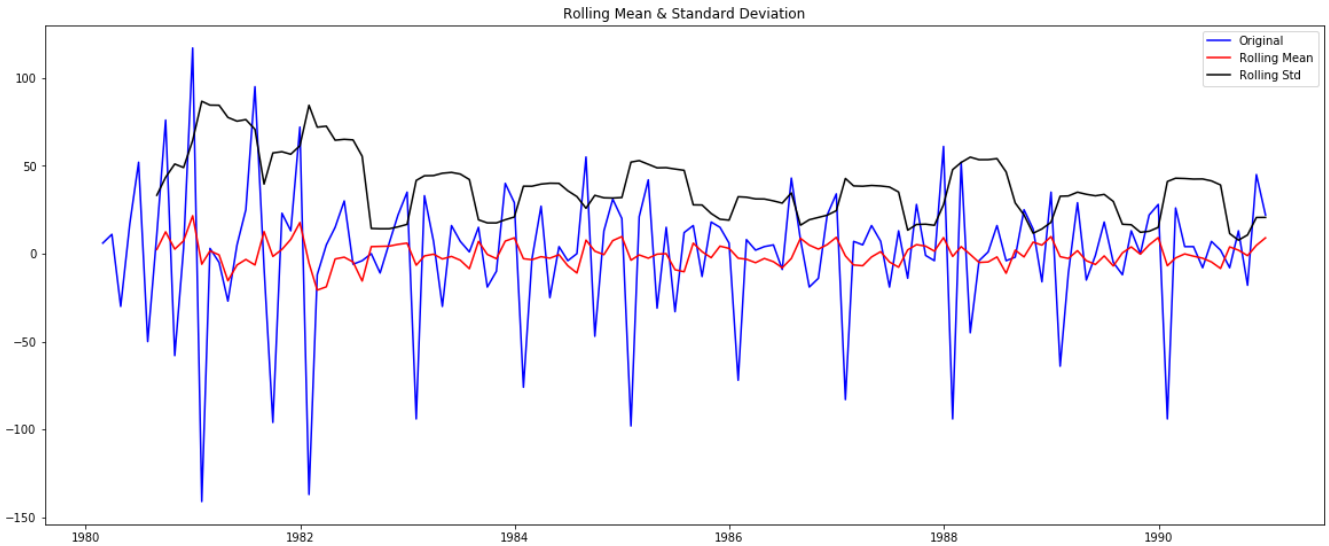
Checking the Stationarity (Rose Training Set)




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

Considering the p value , At Present Series is not stationary (Rose Training Set)

Make the Series Stationarity(Rose Training Set)




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

Considering the p value , Now Series is stationarity (Rose) Since p value is less that .05

Getting a combination of different parameters of p and q(Rose)

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

Creating an empty Dataframe(Rose)

Out[253]:

param	AIC

Checking the AIC against various p,q parameters(Rose)

```
ARIMA(0, 1, 0) - AIC:1335.1526583086775
ARIMA(0, 1, 1) - AIC:1280.7261830464035
ARIMA(0, 1, 2) - AIC:1276.8353735869105
ARIMA(1, 1, 0) - AIC:1319.3483105803125
ARIMA(1, 1, 1) - AIC:1277.7757488779507
ARIMA(1, 1, 2) - AIC:1277.3592330926072
ARIMA(2, 1, 0) - AIC:1300.609261174427
ARIMA(2, 1, 1) - AIC:1279.045689409309
ARIMA(2, 1, 2) - AIC:1279.2986939365255
```

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

Out[255]:

	param	AIC
2	(0, 1, 2)	1276.835374
5	(1, 1, 2)	1277.359233
4	(1, 1, 1)	1277.775749
7	(2, 1, 1)	1279.045689
8	(2, 1, 2)	1279.298694
1	(0, 1, 1)	1280.726183
6	(2, 1, 0)	1300.609261
3	(1, 1, 0)	1319.348311
0	(0, 1, 0)	1335.152658

Storing the Result In Dataframe(Rose)

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.Rose   No. Observations:                  131
Model:                 ARIMA(0, 1, 2)   Log Likelihood                -634.418
Method:                       css-mle   S.D. of innovations             30.167
Date:                Sat, 12 Sep 2020   AIC                           1276.835
Time:                        19:34:39   BIC                           1288.336
Sample:                    02-29-1980   HQIC                          1281.509
                         - 12-31-1990
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -0.4885      0.085     -5.742      0.000      -0.655      -0.322
ma.L1.D.Rose    -0.7601      0.101     -7.499      0.000      -0.959      -0.561
ma.L2.D.Rose    -0.2398      0.095     -2.518      0.013      -0.427      -0.053
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1            1.0001           +0.0000j            1.0001            0.0000
MA.2           -4.1695           +0.0000j            4.1695            0.5000
-----------------------------------------------------------------------------
```

Predict on the Test Set using this model and evaluate the model(Rose)

Model Evaluation(Rose Auto Arima)

```
15.640799000948249
```

Storing th Result In Dataframe(Rose Auto Arima)

Out[259]:

	RMSE
ARIMA(0,1,2)	15.640799

Build an automated version of the SARIMA mode(Rose)

Checking the AIC against various required parameters(Rose)

```
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)
```

Define an empty dataframe

Out[261]:

param	seasonal	AIC

```
SARIMA(0, 1, 0)x(0, 0, 0, 12)7 - AIC:1323.9657875279158
SARIMA(0, 1, 0)x(0, 0, 1, 12)7 - AIC:1145.4230827207484
SARIMA(0, 1, 0)x(0, 0, 2, 12)7 - AIC:976.4375296380877
SARIMA(0, 1, 0)x(1, 0, 0, 12)7 - AIC:1139.921738995602
SARIMA(0, 1, 0)x(1, 0, 1, 12)7 - AIC:1116.0207869386836
SARIMA(0, 1, 0)x(1, 0, 2, 12)7 - AIC:969.6913635753085
SARIMA(0, 1, 0)x(2, 0, 0, 12)7 - AIC:960.8812220353041
SARIMA(0, 1, 0)x(2, 0, 1, 12)7 - AIC:962.8794540697554
SARIMA(0, 1, 0)x(2, 0, 2, 12)7 - AIC:955.5735408945827
SARIMA(0, 1, 1)x(0, 0, 0, 12)7 - AIC:1263.5369097383966
SARIMA(0, 1, 1)x(0, 0, 1, 12)7 - AIC:1098.5554825918332
SARIMA(0, 1, 1)x(0, 0, 2, 12)7 - AIC:923.6314049383755
SARIMA(0, 1, 1)x(1, 0, 0, 12)7 - AIC:1095.7936324918244
SARIMA(0, 1, 1)x(1, 0, 1, 12)7 - AIC:1054.7434330946082
SARIMA(0, 1, 1)x(1, 0, 2, 12)7 - AIC:918.8573483303601
SARIMA(0, 1, 1)x(2, 0, 0, 12)7 - AIC:914.5982866535918
SARIMA(0, 1, 1)x(2, 0, 1, 12)7 - AIC:915.3332430461658
SARIMA(0, 1, 1)x(2, 0, 2, 12)7 - AIC:901.1988290990013
SARIMA(0, 1, 2)x(0, 0, 0, 12)7 - AIC:1251.6675430541038
SARIMA(0, 1, 2)x(0, 0, 1, 12)7 - AIC:1083.486697526493
SARIMA(0, 1, 2)x(0, 0, 2, 12)7 - AIC:913.4938486617701
SARIMA(0, 1, 2)x(1, 0, 0, 12)7 - AIC:1088.8332843414155
SARIMA(0, 1, 2)x(1, 0, 1, 12)7 - AIC:1045.5400933552442
SARIMA(0, 1, 2)x(1, 0, 2, 12)7 - AIC:904.83109136398
SARIMA(0, 1, 2)x(2, 0, 0, 12)7 - AIC:913.0105912257985
SARIMA(0, 1, 2)x(2, 0, 1, 12)7 - AIC:914.1707545029554
SARIMA(0, 1, 2)x(2, 0, 2, 12)7 - AIC:887.9375085680414
SARIMA(1, 1, 0)x(0, 0, 0, 12)7 - AIC:1308.161871082466
SARIMA(1, 1, 0)x(0, 0, 1, 12)7 - AIC:1135.2955447585719
SARIMA(1, 1, 0)x(0, 0, 2, 12)7 - AIC:963.940539125769
SARIMA(1, 1, 0)x(1, 0, 0, 12)7 - AIC:1124.8860786804557
SARIMA(1, 1, 0)x(1, 0, 1, 12)7 - AIC:1105.4080055024747
SARIMA(1, 1, 0)x(1, 0, 2, 12)7 - AIC:958.5001972948003
SARIMA(1, 1, 0)x(2, 0, 0, 12)7 - AIC:939.098477866384
SARIMA(1, 1, 0)x(2, 0, 1, 12)7 - AIC:940.908713366107
SARIMA(1, 1, 0)x(2, 0, 2, 12)7 - AIC:942.297310307086
SARIMA(1, 1, 1)x(0, 0, 0, 12)7 - AIC:1262.1840064255507
SARIMA(1, 1, 1)x(0, 0, 1, 12)7 - AIC:1094.3172708640798
SARIMA(1, 1, 1)x(0, 0, 2, 12)7 - AIC:923.0862224063887
SARIMA(1, 1, 1)x(1, 0, 0, 12)7 - AIC:1083.3937965031942
SARIMA(1, 1, 1)x(1, 0, 1, 12)7 - AIC:1054.7180547134699
SARIMA(1, 1, 1)x(1, 0, 2, 12)7 - AIC:916.354942850357
SARIMA(1, 1, 1)x(2, 0, 0, 12)7 - AIC:905.92490608419
SARIMA(1, 1, 1)x(2, 0, 1, 12)7 - AIC:907.297286747082
SARIMA(1, 1, 1)x(2, 0, 2, 12)7 - AIC:900.6725795936935
SARIMA(1, 1, 2)x(0, 0, 0, 12)7 - AIC:1251.94950407063
SARIMA(1, 1, 2)x(0, 0, 1, 12)7 - AIC:1085.4861928101088
SARIMA(1, 1, 2)x(0, 0, 2, 12)7 - AIC:915.4938402421758
SARIMA(1, 1, 2)x(1, 0, 0, 12)7 - AIC:1090.7760924450708
SARIMA(1, 1, 2)x(1, 0, 1, 12)7 - AIC:1042.6183213683041
SARIMA(1, 1, 2)x(1, 0, 2, 12)7 - AIC:906.7318500287658
SARIMA(1, 1, 2)x(2, 0, 0, 12)7 - AIC:906.1690196793622
SARIMA(1, 1, 2)x(2, 0, 1, 12)7 - AIC:907.4597827872758
SARIMA(1, 1, 2)x(2, 0, 2, 12)7 - AIC:889.9021457559716
SARIMA(2, 1, 0)x(0, 0, 0, 12)7 - AIC:1280.253756153577
SARIMA(2, 1, 0)x(0, 0, 1, 12)7 - AIC:1128.7773704710978
SARIMA(2, 1, 0)x(0, 0, 2, 12)7 - AIC:958.079320882995
SARIMA(2, 1, 0)x(1, 0, 0, 12)7 - AIC:1099.508602157594
SARIMA(2, 1, 0)x(1, 0, 1, 12)7 - AIC:1076.7863198641217
SARIMA(2, 1, 0)x(1, 0, 2, 12)7 - AIC:951.1988165559325
SARIMA(2, 1, 0)x(2, 0, 0, 12)7 - AIC:924.6004792645429
SARIMA(2, 1, 0)x(2, 0, 1, 12)7 - AIC:925.9757801384793
```

```
SARIMA(2, 1, 0)x(2, 0, 2, 12)7 - AIC:927.8380693280818
SARIMA(2, 1, 1)x(0, 0, 0, 12)7 - AIC:1263.2315231800167
SARIMA(2, 1, 1)x(0, 0, 1, 12)7 - AIC:1094.209349194942
SARIMA(2, 1, 1)x(0, 0, 2, 12)7 - AIC:922.9408472077411
SARIMA(2, 1, 1)x(1, 0, 0, 12)7 - AIC:1071.424960110127
SARIMA(2, 1, 1)x(1, 0, 1, 12)7 - AIC:1052.9244471207892
SARIMA(2, 1, 1)x(1, 0, 2, 12)7 - AIC:916.2424912822738
SARIMA(2, 1, 1)x(2, 0, 0, 12)7 - AIC:896.5181608039705
SARIMA(2, 1, 1)x(2, 0, 1, 12)7 - AIC:897.6399565369395
SARIMA(2, 1, 1)x(2, 0, 2, 12)7 - AIC:899.4835866284487
SARIMA(2, 1, 2)x(0, 0, 0, 12)7 - AIC:1253.9102116146687
SARIMA(2, 1, 2)x(0, 0, 1, 12)7 - AIC:1085.964355259821
SARIMA(2, 1, 2)x(0, 0, 2, 12)7 - AIC:916.3258311105169
SARIMA(2, 1, 2)x(1, 0, 0, 12)7 - AIC:1073.2912713702283
SARIMA(2, 1, 2)x(1, 0, 1, 12)7 - AIC:1044.1909353375177
SARIMA(2, 1, 2)x(1, 0, 2, 12)7 - AIC:907.6661488812231
SARIMA(2, 1, 2)x(2, 0, 0, 12)7 - AIC:897.3464439911958
SARIMA(2, 1, 2)x(2, 0, 1, 12)7 - AIC:898.3781891889461
SARIMA(2, 1, 2)x(2, 0, 2, 12)7 - AIC:890.6687980923202
```

Checking the AIC against required parameters(Rose)

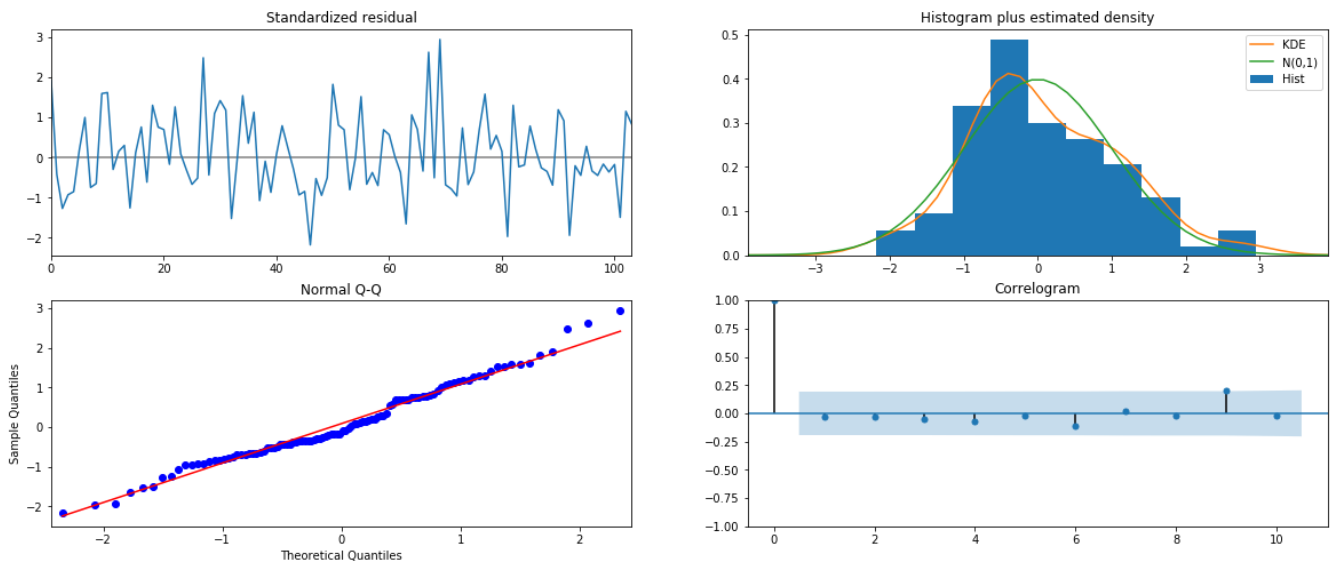
Out[263]:

	param	seasonal	AIC
26	(0, 1, 2)	(2, 0, 2, 12)	887.937509
53	(1, 1, 2)	(2, 0, 2, 12)	889.902146
80	(2, 1, 2)	(2, 0, 2, 12)	890.668798
69	(2, 1, 1)	(2, 0, 0, 12)	896.518161
78	(2, 1, 2)	(2, 0, 0, 12)	897.346444

Model Result(Rose)

```
                                     Statespace Model Results
==========================================================================================
====
Dep. Variable:                                   y   No. Observations:
132
Model:             SARIMAX(0, 1, 2)x(2, 0, 2, 12)   Log Likelihood                -43
6.969
Date:                            Sat, 12 Sep 2020   AIC                            88
7.938
Time:                                    19:36:56   BIC                            90
6.448
Sample:                                         0   HQIC                           89
5.437
                                            - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8427    190.003     -0.004      0.996    -373.243     371.557
ma.L2         -0.1573     29.851     -0.005      0.996     -58.663      58.349
ar.S.L12       0.3467      0.079      4.375      0.000       0.191       0.502
ar.S.L24       0.3023      0.076      3.996      0.000       0.154       0.451
ma.S.L12       0.0767      0.133      0.577      0.564      -0.184       0.337
ma.S.L24      -0.0726      0.146     -0.498      0.618      -0.358       0.213
sigma2       251.3137   4.78e+04      0.005      0.996   -9.33e+04    9.39e+04
===================================================================================
Ljung-Box (Q):                       24.56   Jarque-Bera (JB):                 2.33
Prob(Q):                              0.97   Prob(JB):                         0.31
Heteroskedasticity (H):               0.88   Skew:                             0.37
Prob(H) (two-sided):                  0.70   Kurtosis:                         3.03
===================================================================================

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

```
<Figure size 1800x1440 with 0 Axes>
```



Predict on the Test Set using this model and evaluate the model(Rose)

Out[269]:

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	62.867265	15.928501	31.647976	94.086553
1	70.541190	16.147659	38.892360	102.190021
2	77.356411	16.147657	45.707585	109.005237
3	76.208814	16.147657	44.559988	107.857639
4	72.747398	16.147657	41.098573	104.396224

Model Evaluation(Rose Auto Sarima)

```
26.99203807352471
```

Storing the Result In Dataframe

Out[271]:

	RMSE
ARIMA(0,1,2)	15.640799
SARIMA(1,1,2)(2,0,2,12)	26.992038

Build an automated version of the ARIMA/SARIMA (AIC) (Rose Time series)

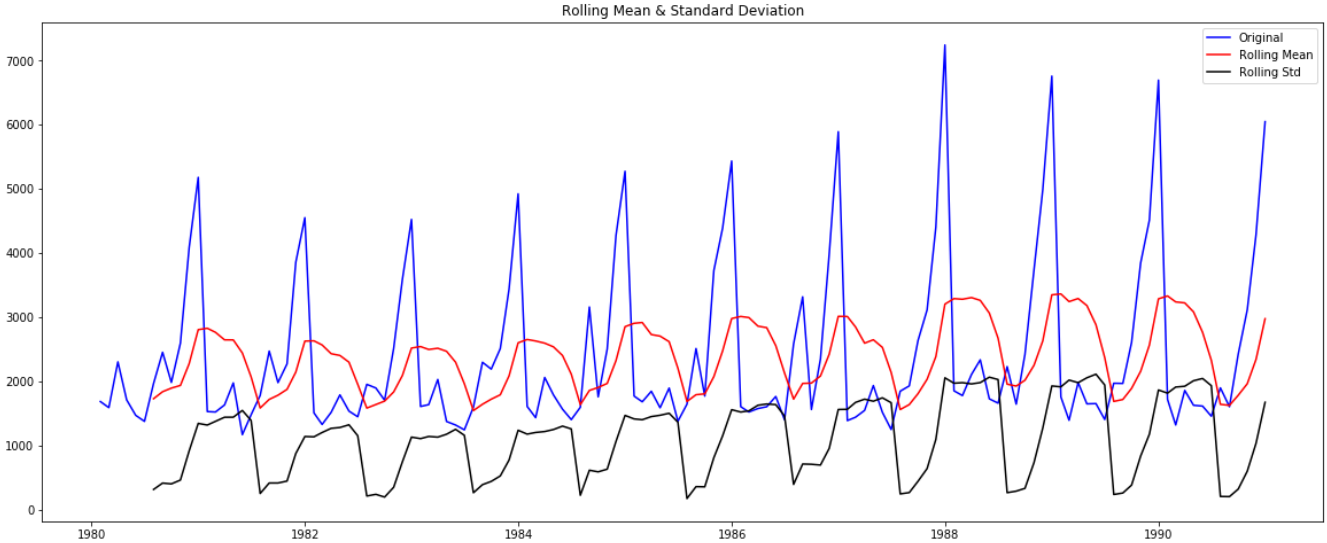
- Automated Arima Model Arima(0.1,2) for Rose Time Series having RMSE 15.64.
- Automated Srima Model SARIMA(1,1,2)(2,0,2,12) for Rose Time Series having RMSE 26.92.

Need to check the stationarity of Training Set (Sparkling) before Building the Arima and Sarima Models

Hypothesis Test (Sparkling Training Set)

- Null Hypothesis Test (H0):- Series is not stationary
- Alternative Hypothesis Test (H1):- Series is stationary

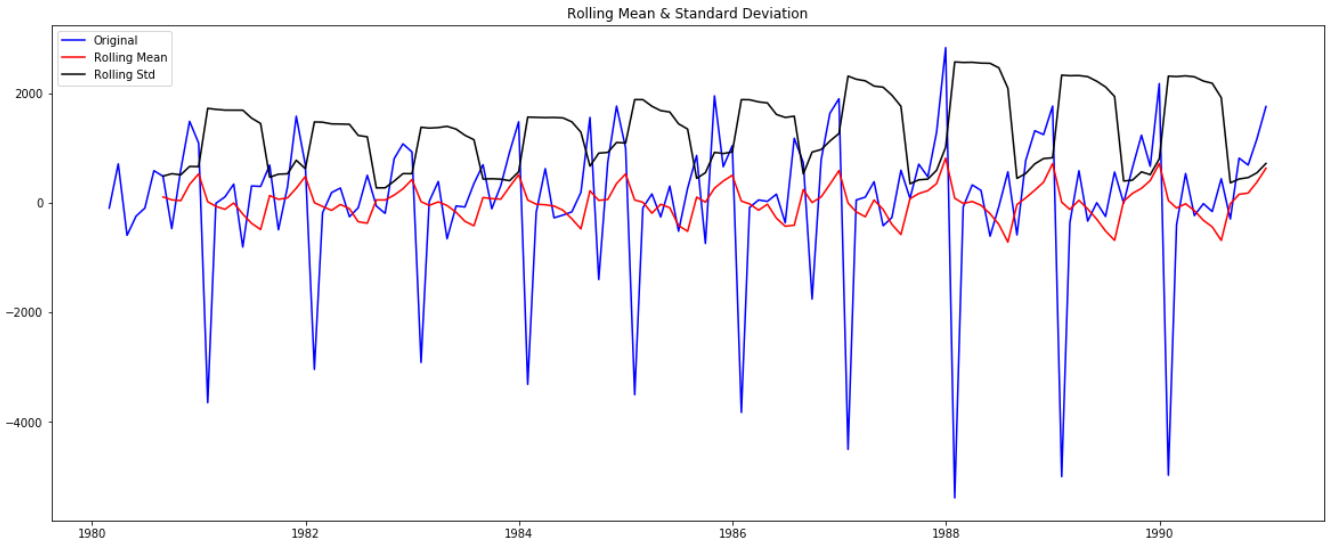
Checking the Stationarity (Sparkling Training Set)




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

Considering the p value , At Present Series is not stationary (Sparkling Training Set)

Making the Series Stationary




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

Considering the p value , Now Series is stationarity (Sparkling) Since p value is less that .05

Getting a combination of different parameters of p and q(Sparkling)

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

Creating an empty Dataframe(Sparkling)

Out[275]:

param	AIC

Checking the AIC against various p,q parameters(Sparkling)

```
ARIMA(0, 1, 0) - AIC:2269.582796371201
ARIMA(0, 1, 1) - AIC:2264.906437115225
ARIMA(0, 1, 2) - AIC:2232.783097684332
ARIMA(1, 1, 0) - AIC:2268.5280605942976
ARIMA(1, 1, 1) - AIC:2235.013945351675
ARIMA(1, 1, 2) - AIC:2233.5976471190743
ARIMA(2, 1, 0) - AIC:2262.0356001096925
ARIMA(2, 1, 1) - AIC:2232.3604898897147
ARIMA(2, 1, 2) - AIC:2210.6176473763135
```

Sort the above AIC values in the ascending order to get the parameters for the minimum AIC value(Sparkling)

Out[277]:

	param	AIC
8	(2, 1, 2)	2210.617647
7	(2, 1, 1)	2232.360490
2	(0, 1, 2)	2232.783098
5	(1, 1, 2)	2233.597647
4	(1, 1, 1)	2235.013945
6	(2, 1, 0)	2262.035600
1	(0, 1, 1)	2264.906437
3	(1, 1, 0)	2268.528061
0	(0, 1, 0)	2269.582796

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:            D.Sparkling   No. Observations:                  131
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1099.309
Method:                       css-mle   S.D. of innovations           1012.410
Date:                Sat, 12 Sep 2020   AIC                           2210.618
Time:                        19:37:11   BIC                           2227.869
Sample:                    02-29-1980   HQIC                          2217.628
                         - 12-31-1990
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 5.5858      0.516     10.819      0.000       4.574       6.598
ar.L1.D.Sparkling     1.2700      0.074     17.047      0.000       1.124       1.416
ar.L2.D.Sparkling    -0.5602      0.074     -7.619      0.000      -0.704      -0.416
ma.L1.D.Sparkling    -1.9984      0.042    -47.174      0.000      -2.081      -1.915
ma.L2.D.Sparkling     0.9984      0.042     23.566      0.000       0.915       1.081
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.1334           -0.7074j            1.3360           -0.0888
AR.2            1.1334           +0.7074j            1.3360            0.0888
MA.1            1.0003           +0.0000j            1.0003            0.0000
MA.2            1.0013           +0.0000j            1.0013            0.0000
-----------------------------------------------------------------------------
```

Predict on the Test Set using this model and evaluate the model(Sparkling)

Model Evaluation(Sparkling Auto Arima)

```
1374.786712941422
```

Creating an empty Dataframe(Sparkling)

Out[281]:

	RMSE
Sparkling_Auto_Arima_RMSE_ARIMA(2,1,2)	1374.786713

Build an Automated version of the SARIMA mode(Sparkling)

```
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)
```

Checking the AIC against various required parameters(Sparkling)

```
SARIMA(0, 1, 0)x(0, 0, 0, 12)6 - AIC:2251.3597196862966
SARIMA(0, 1, 0)x(0, 0, 1, 12)6 - AIC:1956.2614616845638
SARIMA(0, 1, 0)x(0, 0, 2, 12)6 - AIC:1723.1533640238588
SARIMA(0, 1, 0)x(1, 0, 0, 12)6 - AIC:1837.4366022456677
SARIMA(0, 1, 0)x(1, 0, 1, 12)6 - AIC:1806.9905301389376
SARIMA(0, 1, 0)x(1, 0, 2, 12)6 - AIC:1633.2108735791742
SARIMA(0, 1, 0)x(2, 0, 0, 12)6 - AIC:1648.3776153470858
SARIMA(0, 1, 0)x(2, 0, 1, 12)6 - AIC:1647.205415861309
SARIMA(0, 1, 0)x(2, 0, 2, 12)6 - AIC:1630.9898053920822
SARIMA(0, 1, 1)x(0, 0, 0, 12)6 - AIC:2230.1629078505857
SARIMA(0, 1, 1)x(0, 0, 1, 12)6 - AIC:1923.7688649566387
SARIMA(0, 1, 1)x(0, 0, 2, 12)6 - AIC:1692.708957286202
SARIMA(0, 1, 1)x(1, 0, 0, 12)6 - AIC:1797.1795881838439
SARIMA(0, 1, 1)x(1, 0, 1, 12)6 - AIC:1738.0903193773431
SARIMA(0, 1, 1)x(1, 0, 2, 12)6 - AIC:1570.1509144419695
SARIMA(0, 1, 1)x(2, 0, 0, 12)6 - AIC:1605.6751954177505
SARIMA(0, 1, 1)x(2, 0, 1, 12)6 - AIC:1599.2245095300011
SARIMA(0, 1, 1)x(2, 0, 2, 12)6 - AIC:1570.401882349495
SARIMA(0, 1, 2)x(0, 0, 0, 12)6 - AIC:2187.441010168715
SARIMA(0, 1, 2)x(0, 0, 1, 12)6 - AIC:1887.9128007201286
SARIMA(0, 1, 2)x(0, 0, 2, 12)6 - AIC:1659.878989100836
SARIMA(0, 1, 2)x(1, 0, 0, 12)6 - AIC:1790.0326332308127
SARIMA(0, 1, 2)x(1, 0, 1, 12)6 - AIC:1724.167507078052
SARIMA(0, 1, 2)x(1, 0, 2, 12)6 - AIC:1557.1605068607357
SARIMA(0, 1, 2)x(2, 0, 0, 12)6 - AIC:1603.9654774379253
SARIMA(0, 1, 2)x(2, 0, 1, 12)6 - AIC:1600.5438799870506
SARIMA(0, 1, 2)x(2, 0, 2, 12)6 - AIC:1557.1215626849962
SARIMA(1, 1, 0)x(0, 0, 0, 12)6 - AIC:2250.3181267386713
SARIMA(1, 1, 0)x(0, 0, 1, 12)6 - AIC:1954.393833990475
SARIMA(1, 1, 0)x(0, 0, 2, 12)6 - AIC:1721.2688476355372
SARIMA(1, 1, 0)x(1, 0, 0, 12)6 - AIC:1811.2440279330344
SARIMA(1, 1, 0)x(1, 0, 1, 12)6 - AIC:1788.5343592678848
SARIMA(1, 1, 0)x(1, 0, 2, 12)6 - AIC:1616.4894404973431
SARIMA(1, 1, 0)x(2, 0, 0, 12)6 - AIC:1621.6355080129129
SARIMA(1, 1, 0)x(2, 0, 1, 12)6 - AIC:1617.1356132107712
SARIMA(1, 1, 0)x(2, 0, 2, 12)6 - AIC:1616.5412067514503
SARIMA(1, 1, 1)x(0, 0, 0, 12)6 - AIC:2204.9340491545663
SARIMA(1, 1, 1)x(0, 0, 1, 12)6 - AIC:1907.3558974126247
SARIMA(1, 1, 1)x(0, 0, 2, 12)6 - AIC:1678.0981352620317
SARIMA(1, 1, 1)x(1, 0, 0, 12)6 - AIC:1775.1424478024594
SARIMA(1, 1, 1)x(1, 0, 1, 12)6 - AIC:1739.544925062429
SARIMA(1, 1, 1)x(1, 0, 2, 12)6 - AIC:1571.3248863528565
SARIMA(1, 1, 1)x(2, 0, 0, 12)6 - AIC:1590.6161606873197
SARIMA(1, 1, 1)x(2, 0, 1, 12)6 - AIC:1586.3142234789561
SARIMA(1, 1, 1)x(2, 0, 2, 12)6 - AIC:1571.8069970537151
SARIMA(1, 1, 2)x(0, 0, 0, 12)6 - AIC:2188.4633450503966
SARIMA(1, 1, 2)x(0, 0, 1, 12)6 - AIC:1889.7708307496468
SARIMA(1, 1, 2)x(0, 0, 2, 12)6 - AIC:1659.6291421759486
SARIMA(1, 1, 2)x(1, 0, 0, 12)6 - AIC:1771.8259799640057
SARIMA(1, 1, 2)x(1, 0, 1, 12)6 - AIC:1723.987179893407
SARIMA(1, 1, 2)x(1, 0, 2, 12)6 - AIC:1555.5842470091523
SARIMA(1, 1, 2)x(2, 0, 0, 12)6 - AIC:1588.4216933305888
SARIMA(1, 1, 2)x(2, 0, 1, 12)6 - AIC:1585.510160435047
SARIMA(1, 1, 2)x(2, 0, 2, 12)6 - AIC:1556.0767705363062
SARIMA(2, 1, 0)x(0, 0, 0, 12)6 - AIC:2227.302761872421
SARIMA(2, 1, 0)x(0, 0, 1, 12)6 - AIC:1946.4383435411748
SARIMA(2, 1, 0)x(0, 0, 2, 12)6 - AIC:1711.412303983535
SARIMA(2, 1, 0)x(1, 0, 0, 12)6 - AIC:1780.7646066053217
SARIMA(2, 1, 0)x(1, 0, 1, 12)6 - AIC:1756.9357347830432
SARIMA(2, 1, 0)x(1, 0, 2, 12)6 - AIC:1600.970220448109
SARIMA(2, 1, 0)x(2, 0, 0, 12)6 - AIC:1592.2403464858617
SARIMA(2, 1, 0)x(2, 0, 1, 12)6 - AIC:1587.6344987484952
```

```
SARIMA(2, 1, 0)x(2, 0, 2, 12)6 - AIC:1585.9191731701335
SARIMA(2, 1, 1)x(0, 0, 0, 12)6 - AIC:2199.858613145632
SARIMA(2, 1, 1)x(0, 0, 1, 12)6 - AIC:1905.0209495439822
SARIMA(2, 1, 1)x(0, 0, 2, 12)6 - AIC:1675.4234080334497
SARIMA(2, 1, 1)x(1, 0, 0, 12)6 - AIC:1792.8234289905215
SARIMA(2, 1, 1)x(1, 0, 1, 12)6 - AIC:1740.0911247057754
SARIMA(2, 1, 1)x(1, 0, 2, 12)6 - AIC:1571.988827921354
SARIMA(2, 1, 1)x(2, 0, 0, 12)6 - AIC:1577.123506089681
SARIMA(2, 1, 1)x(2, 0, 1, 12)6 - AIC:1573.1595848658199
SARIMA(2, 1, 1)x(2, 0, 2, 12)6 - AIC:1572.3428681888954
SARIMA(2, 1, 2)x(0, 0, 0, 12)6 - AIC:2176.868114915305
SARIMA(2, 1, 2)x(0, 0, 1, 12)6 - AIC:1892.2372640176598
SARIMA(2, 1, 2)x(0, 0, 2, 12)6 - AIC:1661.5523432392195
SARIMA(2, 1, 2)x(1, 0, 0, 12)6 - AIC:1757.2140932497
SARIMA(2, 1, 2)x(1, 0, 1, 12)6 - AIC:1725.6005728109985
SARIMA(2, 1, 2)x(1, 0, 2, 12)6 - AIC:1557.3404041799117
SARIMA(2, 1, 2)x(2, 0, 0, 12)6 - AIC:1599.3524157640218
SARIMA(2, 1, 2)x(2, 0, 1, 12)6 - AIC:1573.547600511475
SARIMA(2, 1, 2)x(2, 0, 2, 12)6 - AIC:1557.6893217689922
```

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

Out[284]:

	param	seasonal	AIC
26	(0, 1, 2)	(2, 0, 2, 12)	887.937509
53	(1, 1, 2)	(2, 0, 2, 12)	889.902146
80	(2, 1, 2)	(2, 0, 2, 12)	890.668798
69	(2, 1, 1)	(2, 0, 0, 12)	896.518161
78	(2, 1, 2)	(2, 0, 0, 12)	897.346444

Model Result(Sparkling)

```
                                     Statespace Model Results
==========================================================================================
====
Dep. Variable:                                   y   No. Observations:
132
Model:             SARIMAX(0, 1, 2)x(2, 0, 2, 12)   Log Likelihood                 -77
1.561
Date:                            Sat, 12 Sep 2020   AIC                            155
7.122
Time:                                    19:40:40   BIC                            157
5.632
Sample:                                         0   HQIC                           156
4.621
                                            - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.7771      0.101     -7.680      0.000      -0.975      -0.579
ma.L2         -0.1239      0.121     -1.022      0.307      -0.361       0.114
ar.S.L12       0.6982      0.751      0.930      0.353      -0.774       2.170
ar.S.L24       0.3598      0.783      0.460      0.646      -1.174       1.894
ma.S.L12       1.6416      3.716      0.442      0.659      -5.641       8.925
ma.S.L24      -1.5935      2.664     -0.598      0.550      -6.814       3.627
sigma2      2.914e+04   9.51e+04      0.307      0.759   -1.57e+05    2.15e+05
===================================================================================
Ljung-Box (Q):                       26.83   Jarque-Bera (JB):                22.06
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               1.45   Skew:                             0.49
Prob(H) (two-sided):                  0.28   Kurtosis:                         5.04
===================================================================================

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

Predict on the Test Set using this model and evaluate the model(Sparkling)

Out[287]:

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	1411.398364	397.249410	632.803827	2189.992901
1	1233.112372	407.000817	435.405429	2030.819316
2	1700.243136	408.886155	898.840998	2501.645275
3	1586.610616	410.772334	781.511637	2391.709596
4	1432.428731	412.650259	623.649085	2241.208377

```
526.4322500911778
```

Storing the Result In Dataframe

Out[291]:

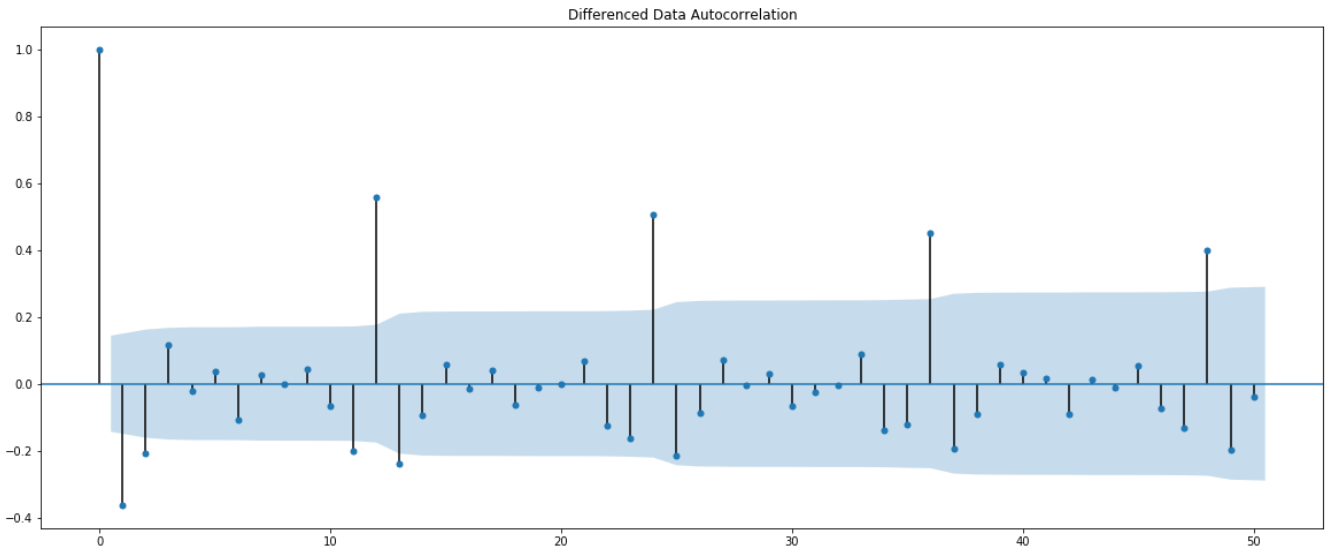
	RMSE
Sparkling_Auto_Arima_RMSE_ARIMA(2,1,2)	1374.786713
auto_SARIMA(0,1,2)(2,0,2,12)	526.432250

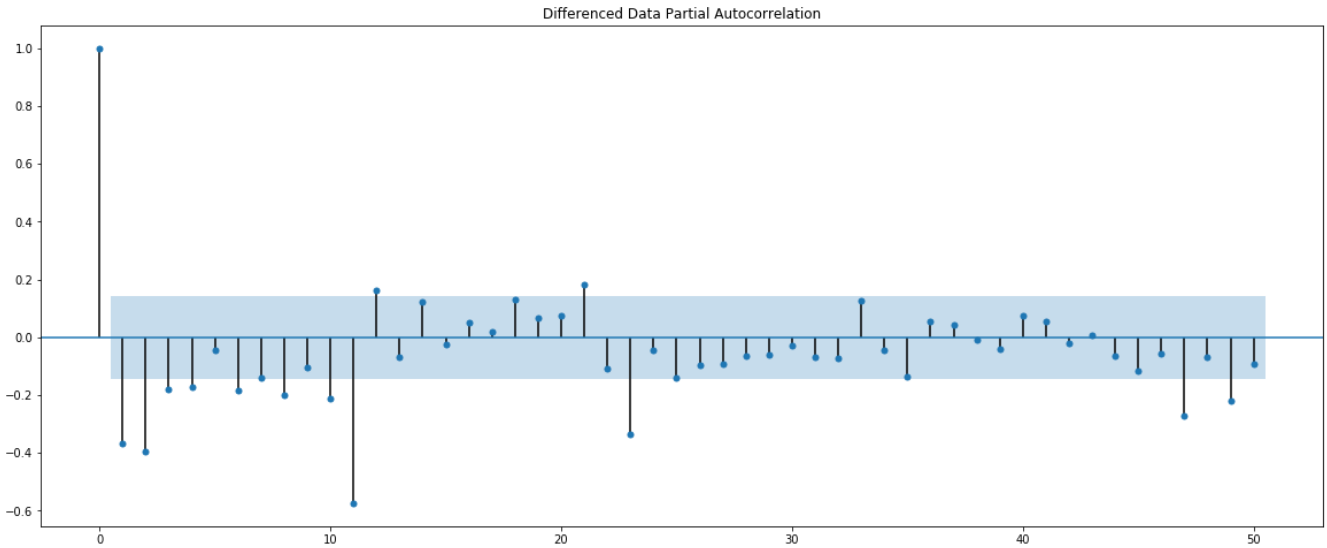
Build an automated version of the ARIMA/SARIMA (AIC) (Sparkling Time series)

- Automated Arima Model Sparkling_Auto_Arima_RMSE_ARIMA(2,1,2) for Sparkling Time Series having RMSE 1374.78
- Automated Srima Model auto_SARIMA(0,1,2)(2,0,2,12 for Sparkling Time Series having RMSE 526.432250

Question 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.

For Rose Time Series





Manual Arima Model(Rose)

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.Rose   No. Observations:                  131
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -633.649
Method:                       css-mle   S.D. of innovations             29.975
Date:                Sat, 12 Sep 2020   AIC                           1279.299
Time:                        19:40:43   BIC                           1296.550
Sample:                    02-29-1980   HQIC                          1286.309
                         - 12-31-1990
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -0.4911      0.081     -6.076      0.000      -0.649      -0.333
ar.L1.D.Rose    -0.4383      0.218     -2.015      0.046      -0.865      -0.012
ar.L2.D.Rose     0.0269      0.109      0.246      0.806      -0.188       0.241
ma.L1.D.Rose    -0.3316      0.203     -1.633      0.105      -0.729       0.066
ma.L2.D.Rose    -0.6684      0.201     -3.332      0.001      -1.062      -0.275
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -2.0290           +0.0000j            2.0290            0.5000
AR.2           18.3389           +0.0000j           18.3389            0.0000
MA.1            1.0000           +0.0000j            1.0000            0.0000
MA.2           -1.4961           +0.0000j            1.4961            0.5000
-----------------------------------------------------------------------------
```

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

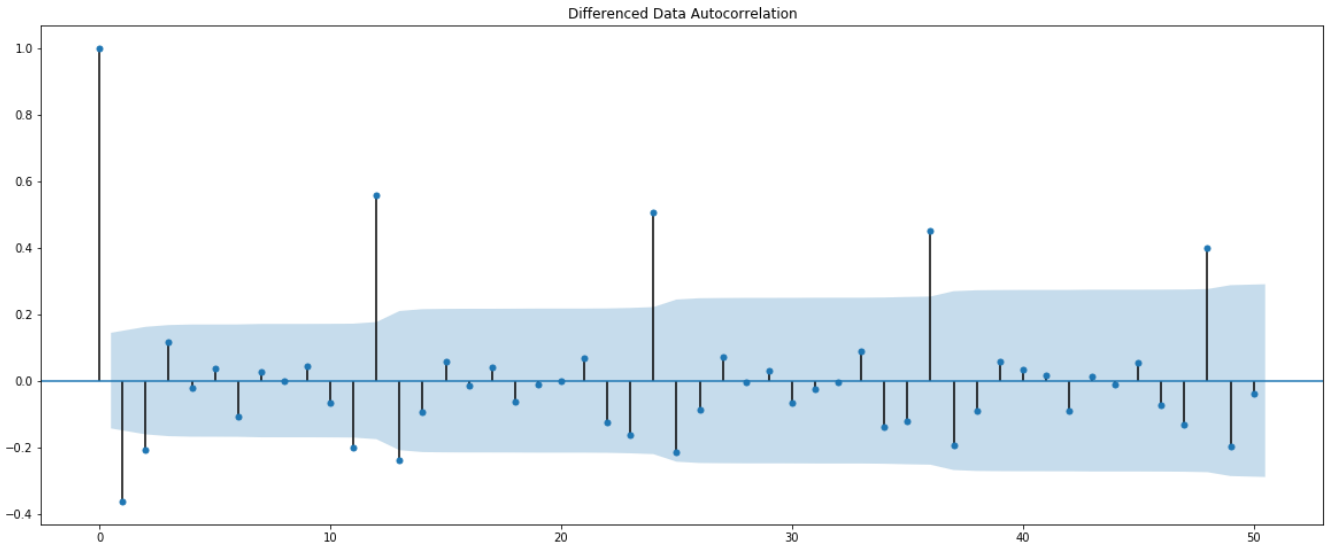
```
15.376562722433532
```

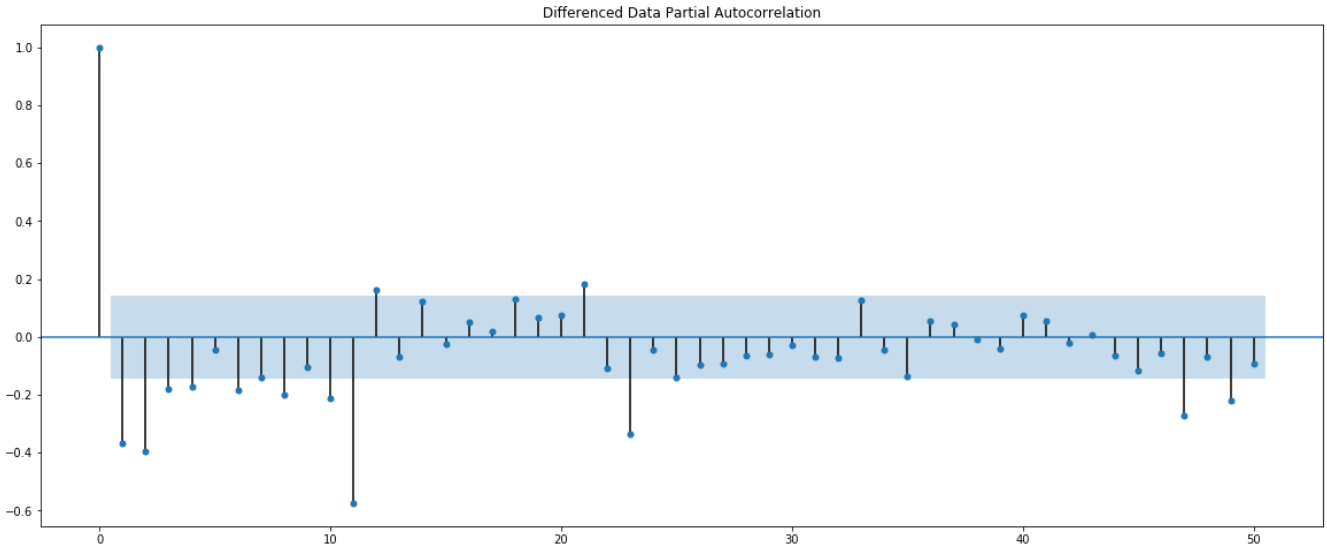
Storing the Result in dataframe

Out[362]:

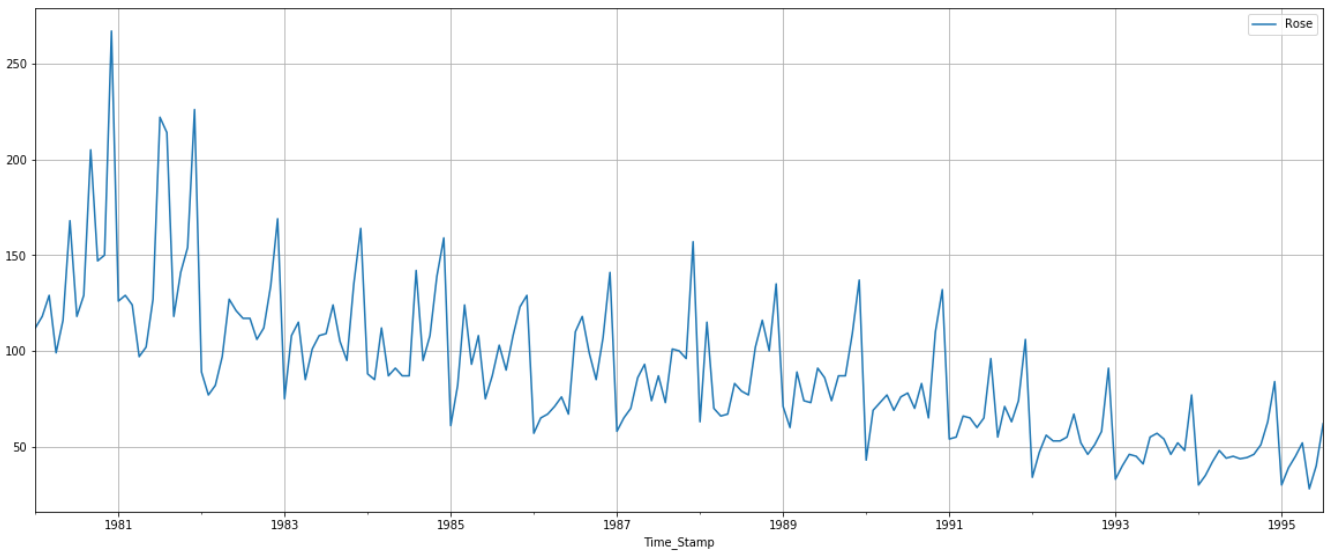
	RMSE
ARIMA(2,1,2)	15.376563

Build a Manual SARIMA Model (Rose)



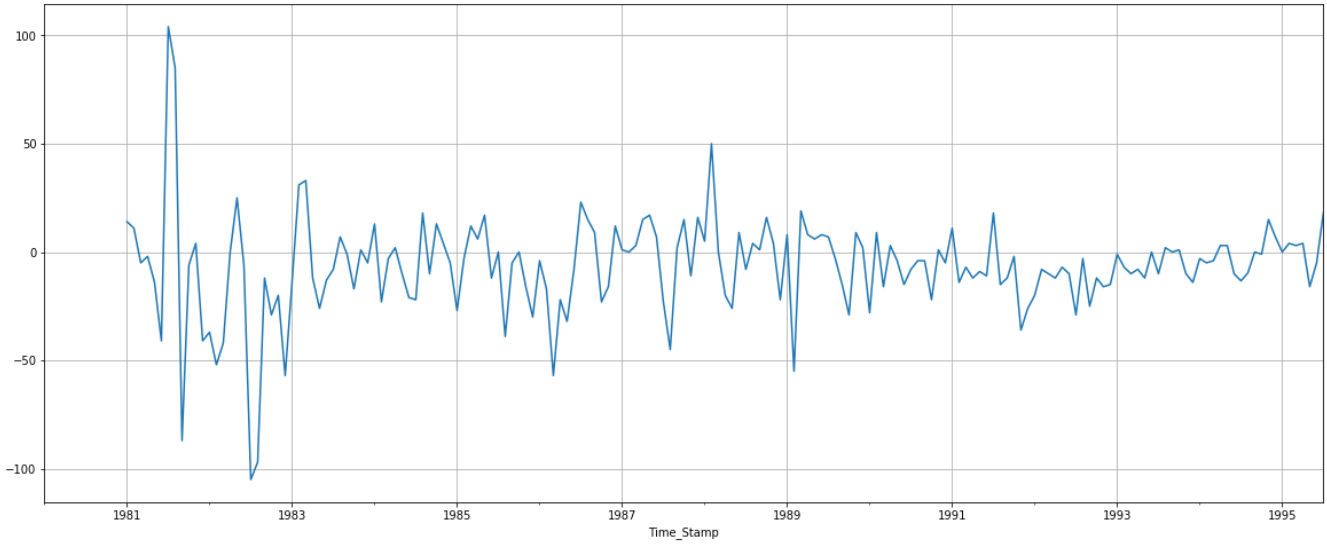


Original Rose Series



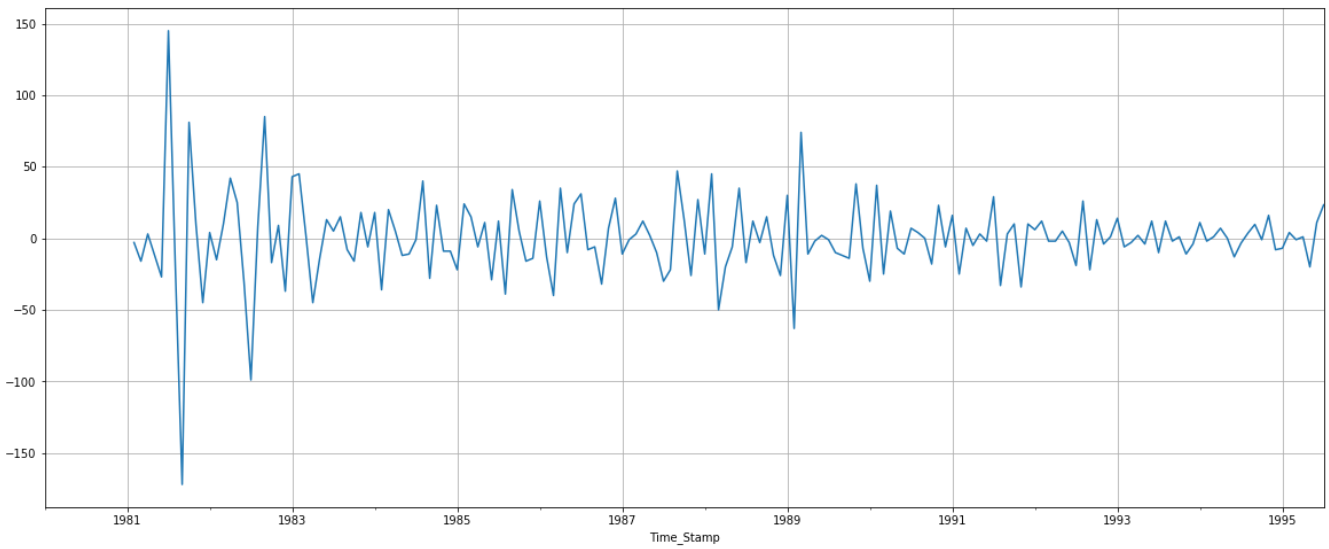
We see that our ACF plot at the seasonal interval (12) does not taper off. So, we go ahead and take a seasonal differencing of the original series. Before that let us look at the original series.

Plotting the Rose Series At lag 12

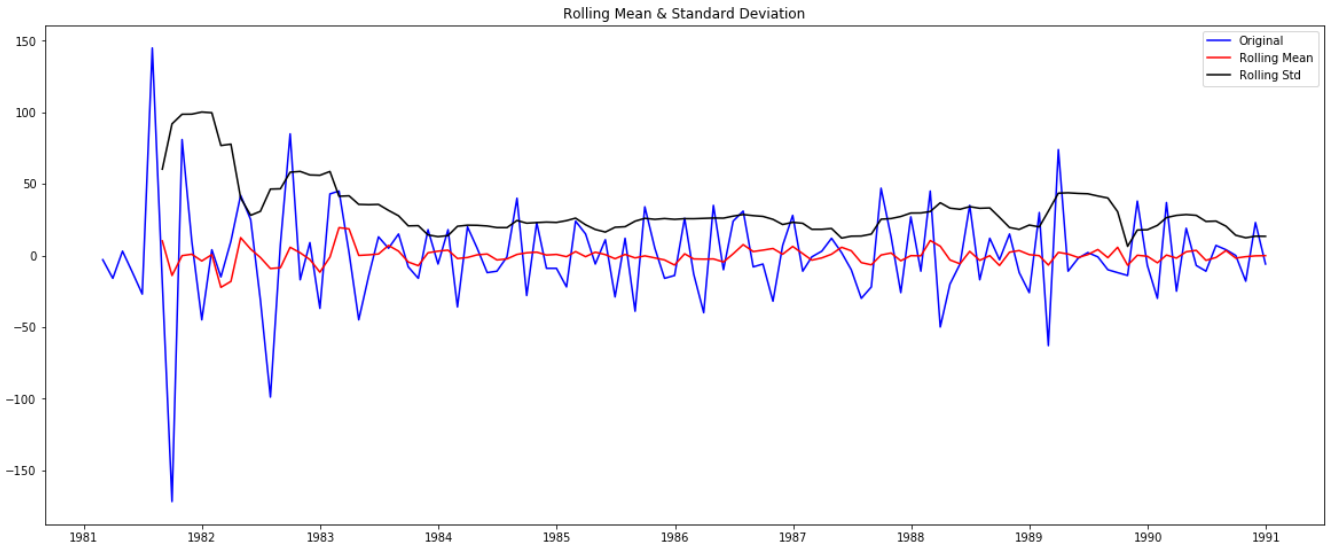


We see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series.

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 (Rose)

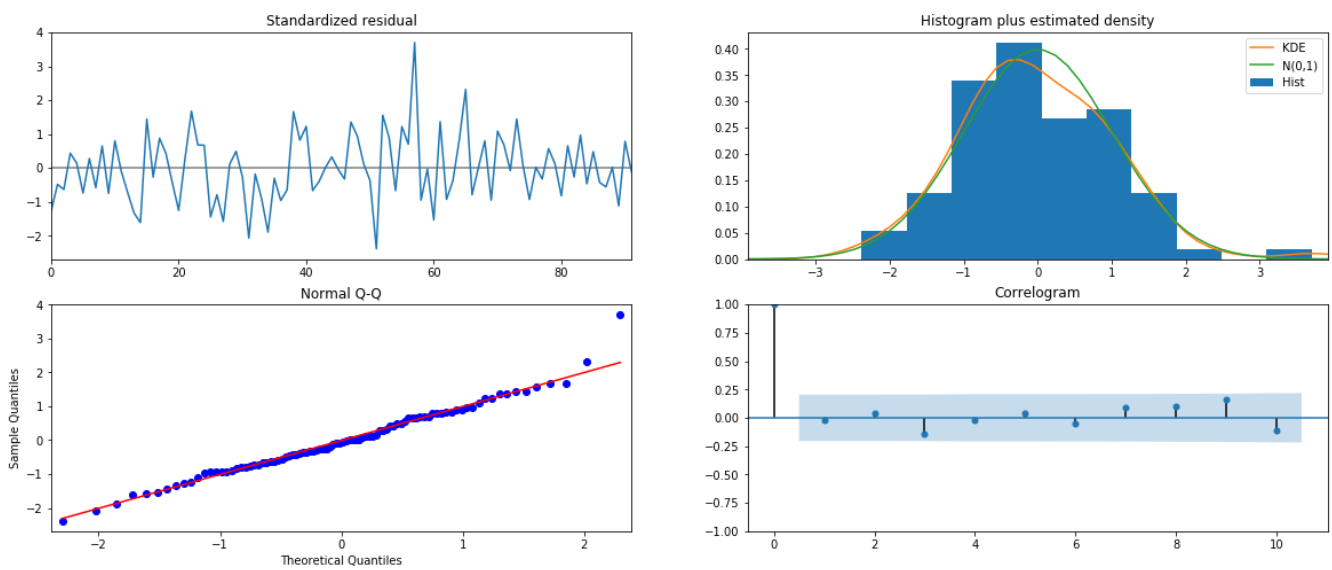



```
Results of Dickey-Fuller Test:
Test Statistic                  -3.692348
p-value                          0.004222
#Lags Used                      11.000000
Number of Observations Used    107.000000
Critical Value (1%)             -3.492996
Critical Value (5%)             -2.888955
Critical Value (10%)            -2.581393
dtype: float64
```

Build a Manual SARIMA Model (Rose)

```
                                     Statespace Model Results
==========================================================================================
====
Dep. Variable:                                   y   No. Observations:
132
Model:             SARIMAX(2, 1, 2)x(2, 1, 2, 12)   Log Likelihood                -37
9.498
Date:                            Sat, 12 Sep 2020   AIC                            77
6.996
Time:                                    19:41:10   BIC                            79
9.692
Sample:                                         0   HQIC                           78
6.156
                                            - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8551      0.146     -5.838      0.000      -1.142      -0.568
ar.L2         -0.0022      0.125     -0.017      0.986      -0.247       0.242
ma.L1          0.0121      0.184      0.066      0.948      -0.348       0.372
ma.L2         -0.9435      0.150     -6.285      0.000      -1.238      -0.649
ar.S.L12       0.0347      0.185      0.188      0.851      -0.328       0.397
ar.S.L24      -0.0459      0.029     -1.599      0.110      -0.102       0.010
ma.S.L12      -0.7224      0.333     -2.172      0.030      -1.374      -0.071
ma.S.L24      -0.0772      0.212     -0.364      0.716      -0.493       0.339
sigma2       192.1845     39.496      4.866      0.000     114.775     269.594
===================================================================================
Ljung-Box (Q):                       29.82   Jarque-Bera (JB):                 7.06
Prob(Q):                              0.88   Prob(JB):                         0.03
Heteroskedasticity (H):               0.87   Skew:                             0.45
Prob(H) (two-sided):                  0.71   Kurtosis:                         4.01
===================================================================================

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

Dignostic Results of Manual Sarima Model(Rose)



Predict on the Test Set using this model and evaluate the model(Rose)

```
16.606637572967568
```

Out[365]:

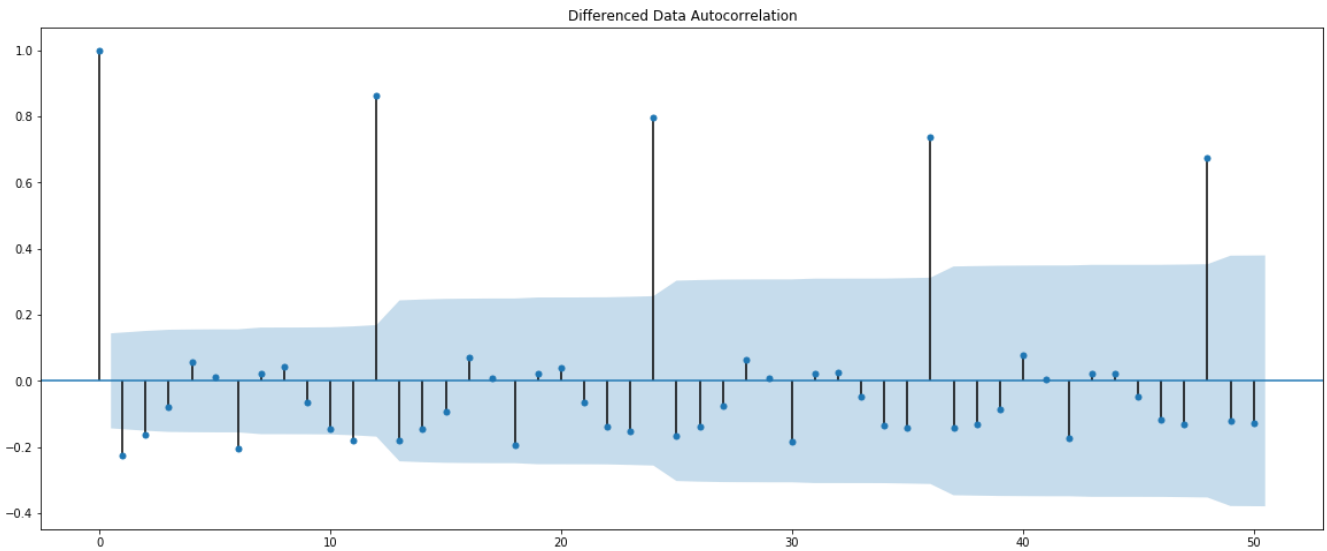
	RMSE
ARIMA(2,1,2)	15.376563
SARIMA(2,1,2)(2,1,2,12)	16.606638

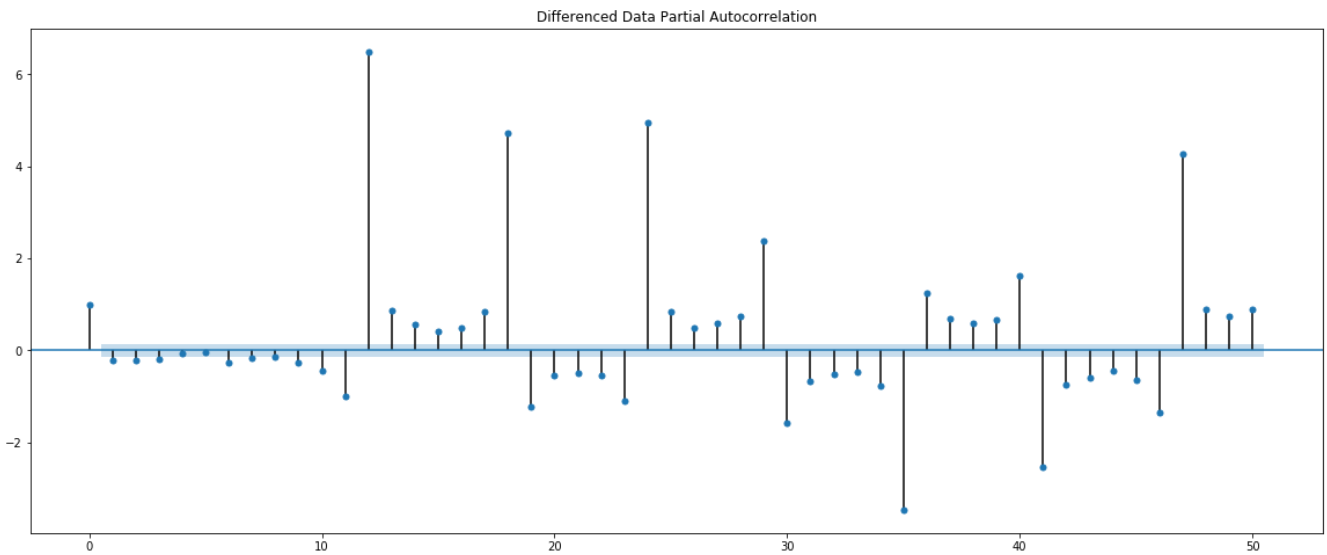
Build an Manual version of the ARIMA/SARIMA (AIC) (Rose Time series)

- Manual Arima Model ARIMA(2,1,2) for Rose Time Series having RMSE 15.37
- Manual Srima Model SARIMA(2,1,2)(2,1,2,12) for Rose Time Series having RMSE 16.60

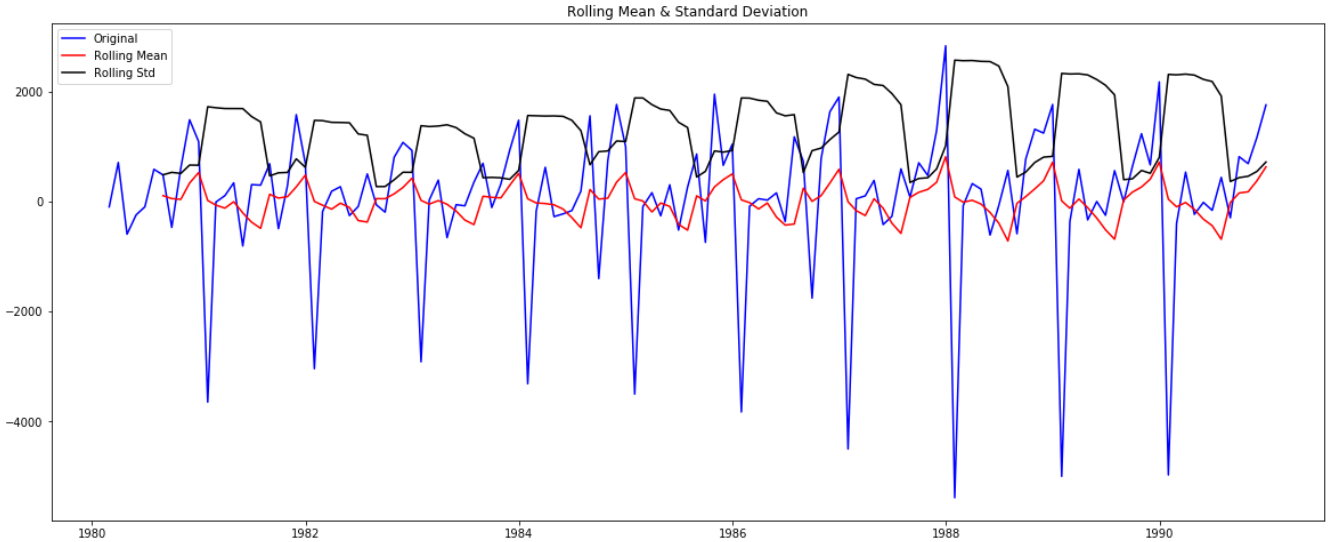
Manual Arima Model(Sparkling)

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





Check the stationarity Before Preparing the Model(Sparkling)




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

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:            D.Sparkling   No. Observations:                  131
Model:                 ARIMA(0, 1, 2)   Log Likelihood               -1112.392
Method:                       css-mle   S.D. of innovations           1159.696
Date:                Sat, 12 Sep 2020   AIC                           2232.783
Time:                        19:41:18   BIC                           2244.284
Sample:                    02-29-1980   HQIC                          2237.456
                         - 12-31-1990
=======================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------
const                 6.2471      3.800      1.644      0.103      -1.201      13.695
ma.L1.D.Sparkling    -0.5555      0.073     -7.583      0.000      -0.699      -0.412
ma.L2.D.Sparkling    -0.4445      0.071     -6.247      0.000      -0.584      -0.305
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1            1.0000           +0.0000j            1.0000            0.0000
MA.2           -2.2495           +0.0000j            2.2495            0.5000
-----------------------------------------------------------------------------
```

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

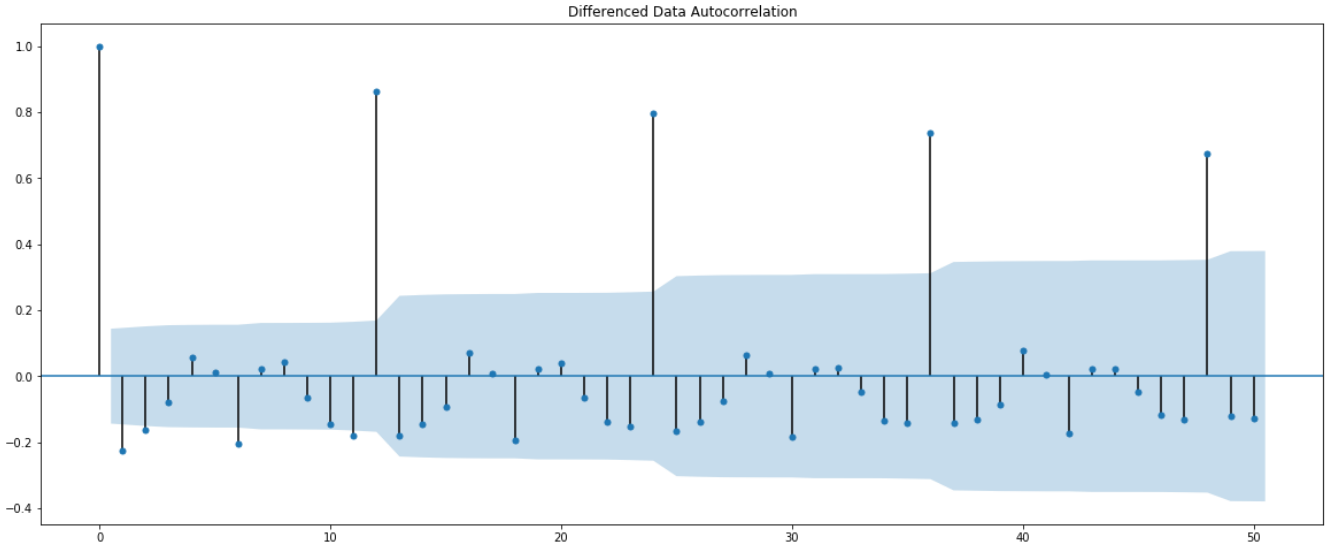
```
1417.4985384817937
```

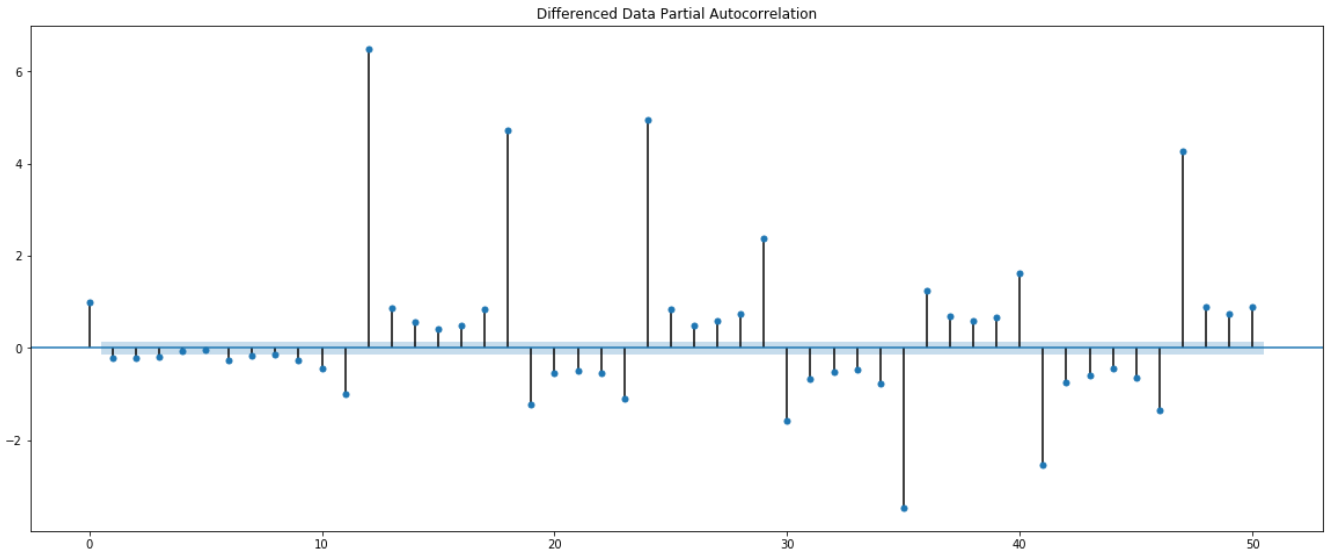
Storing the Result In Dataframe

Out[366]:

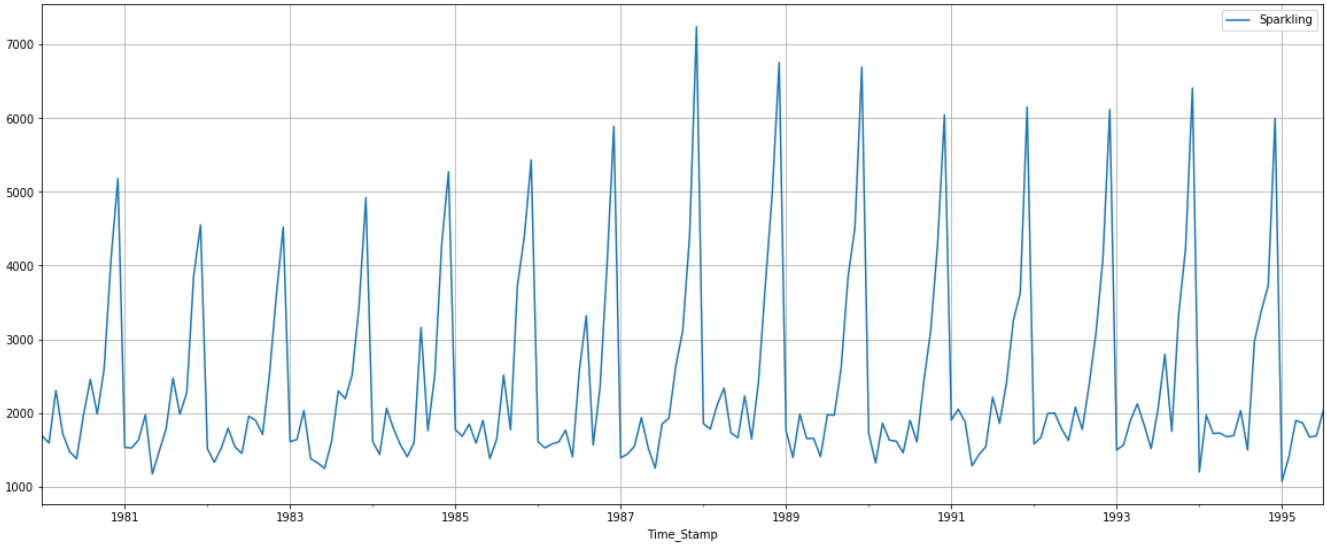
	RMSE
ManualSparkling_ARIMA(0,1,2)	1417.498538

Build a version of the SARIMA model for which the best parameters are selected by looking at the ACF and the PACF plots(Sparkling)

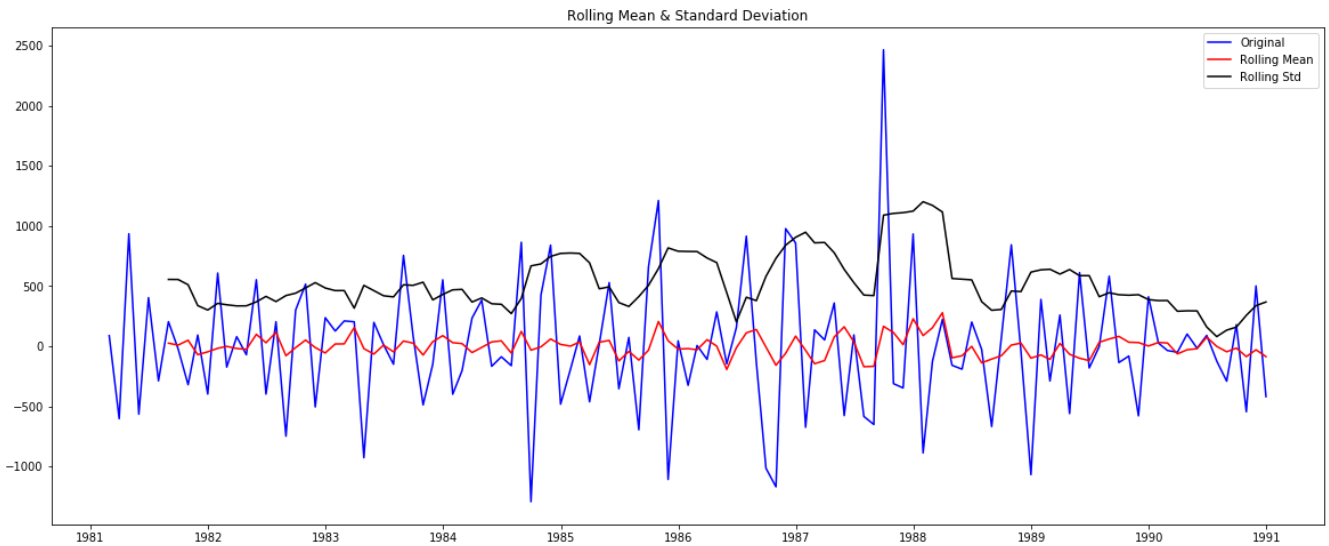




Original Sparkling Series



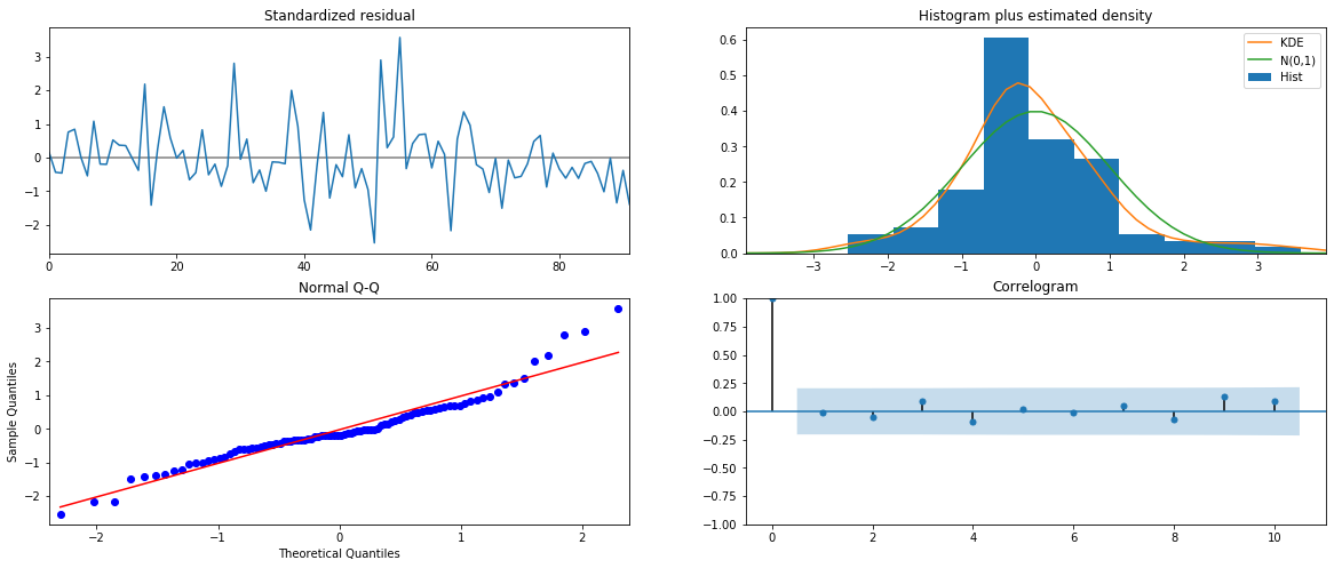
Checking the stationarity (Sparkling Series)




```
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
```

Building the Manual Srima Model(Sparkling)

```
                                     Statespace Model Results
==========================================================================================
======
Dep. Variable:                                   y   No. Observations:
132
Model:             SARIMAX(0, 1, 2)x(0, 1, 2, 12)   Log Likelihood                  -6
86.242
Date:                            Sat, 12 Sep 2020   AIC                             13
82.484
Time:                                    19:41:31   BIC                             13
95.093
Sample:                                         0   HQIC                            13
87.573
                                            - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.7223      0.107     -6.752      0.000      -0.932      -0.513
ma.L2         -0.2292      0.137     -1.671      0.095      -0.498       0.040
ma.S.L12      -0.4113      0.087     -4.743      0.000      -0.581      -0.241
ma.S.L24      -0.0419      0.138     -0.304      0.761      -0.312       0.228
sigma2      1.736e+05   2.06e+04      8.425      0.000    1.33e+05    2.14e+05
===================================================================================
Ljung-Box (Q):                       22.84   Jarque-Bera (JB):                27.42
Prob(Q):                              0.99   Prob(JB):                         0.00
Heteroskedasticity (H):               0.84   Skew:                             0.80
Prob(H) (two-sided):                  0.62   Kurtosis:                         5.15
===================================================================================

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



Model Evaluation (Manual SRIMA Sparkling)

```
321.47625535055874
```

Out[370]:

	RMSE
ManualSparkling_ARIMA(0,1,2)	1417.498538
ManualSARIMA(2,1,2)(2,1,2,12)	321.476255

Build an Manual version of the ARIMA/SARIMA (AIC) (Sparkling Time series)

- Manual Arima Model ManualSparkling_ARIMA(0,1,2 for Sparkling Time Series having RMSE 1417.49
- Manual Srima Model ManualSARIMA(2,1,2)(2,1,2,12) for Sparkling Time Series having RMSE 321.47

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

Rose TIme Series Models in Ascending order on the basis of Test RMSE

Out[337]:

	Test RMSE
Rose_Alpha=.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.665739
Rose_2pointTrailingMovingAverage	11.530180
Rose_4pointTrailingMovingAverage	14.462330
Rose_6pointTrailingMovingAverage	14.586916
Rose_9pointTrailingMovingAverage	14.740112
Rose_RegressionOnTime	15.291460
RoseManualArima_ARIMA(2,1,2)	15.376563
RoseAutoArima_ARIMA(0,1,2)	15.640799
RoseManualSrima_SARIMA(2,1,2)(2,1,2,12	16.606638
alpha=.1,beta=.4,gamma=0.0,Rose_TripleExponentialSmoothing	17.445169
RoseAutoSrima_(1,1,2)(2,0,2,12)	26.992038
Rose_SimpleExponentialSmoothing	36.858586
Alpha=0.1,Rose_SimpleExponentialSmoothing	36.890375
Alpha=0.1,Beta=0.1,Rose DoubleExponentialSmoothing	36.987695
Alpha=0.2,Rose_SimpleExponentialSmoothing	41.423877
Rose_SimpleAverageModel	53.521557
Rose_NaiveModel	79.778066

Best Model (Rose Time series) in Totality

- Rose_Alpha=.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing having RMSE 9.665739

Sparkling TIme Series Models in Ascending order on the basis of Test RMSE

Out[336]:

	Test RMSE
ManualSARIMA(2,1,2)(2,1,2,12)	321.476255
Alpha=.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	338.458417
SparklingAutofitTripleExponentialSmoothing	362.791906
auto_SARIMA(1,1,2)(2,0,2,12)	526.432250
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
SimpleAverageModel	1275.081804
Sparkling_SimpleExponentialSmoothing	1275.081823
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Sparkling_Auto_Arima_RMSE_ARIMA(2,1,2)	1374.786713
Alpha =0.1 Sparkling Simple Exponential Smoothing predictions	1375.393398
ManualSparkling_ARIMA(0,1,2)	1417.498538
Sparkling Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1595.206839
Alpha =0.2 Sparkling Simple Exponential Smoothing predictions	1778.564670
RegressionOnTime	2665.351076
NaiveModel	3864.279352

Best Model (Sparkling Time series) in Totality

- ManualSARIMA(2,1,2)(2,1,2,12 having RMSE 321.476255

Question 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

Building the most optimum model on the Full Data.(Rose)

We see that the best model is the Triple Exponential Smoothing with the parameters α = .1, β = 0.2 and γ = 0.2.

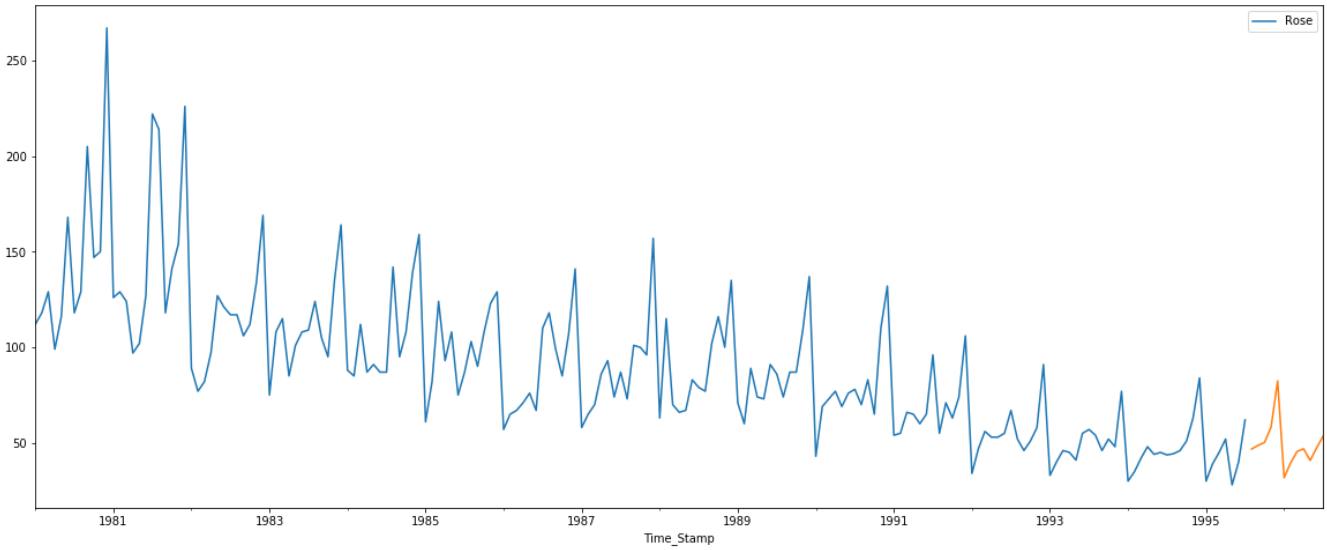
RMSE of Full Model(Rose)

```
RMSE: 20.88649228267161
MAPE: 14.5
```

Prediction of another 12 Months(Rose)

Out[355]:
```
1995-08-31    46.776563
1995-09-30    48.694691
1995-10-31    50.279311
1995-11-30    58.326617
1995-12-31    82.417296
1996-01-31    31.756620
1996-02-29    39.531421
1996-03-31    45.519735
1996-04-30    46.892167
1996-05-31    40.785302
1996-06-30    47.408245
1996-07-31    53.527683
Freq: M, dtype: float64
```

Prediction of original and Future Data(Rose)

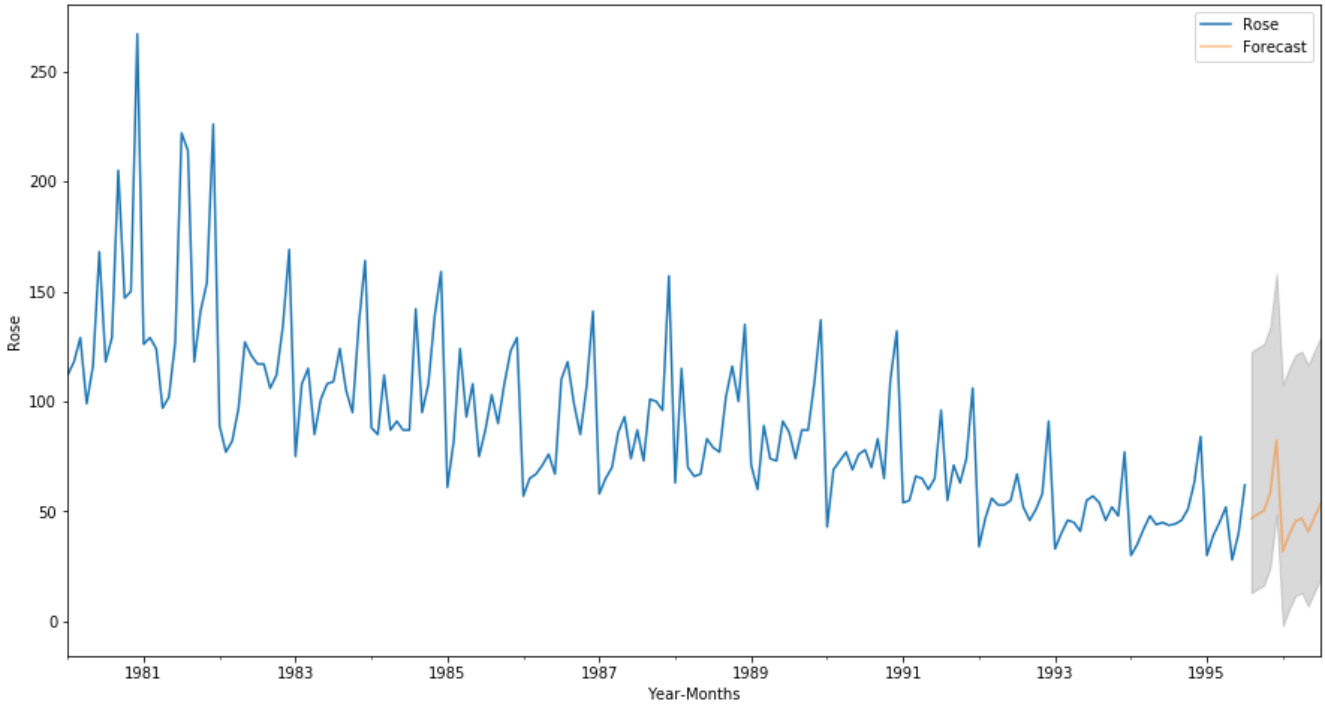


Calculated the upper and lower confidence bands at 95% confidence level

Out[357]:

	lower_CI	prediction	upper_ci
1995-08-31	12.881873	46.776563	122.568922
1995-09-30	14.800000	48.694691	124.487050
1995-10-31	16.384620	50.279311	126.071670
1995-11-30	24.431927	58.326617	134.118976
1995-12-31	48.522605	82.417296	158.209655

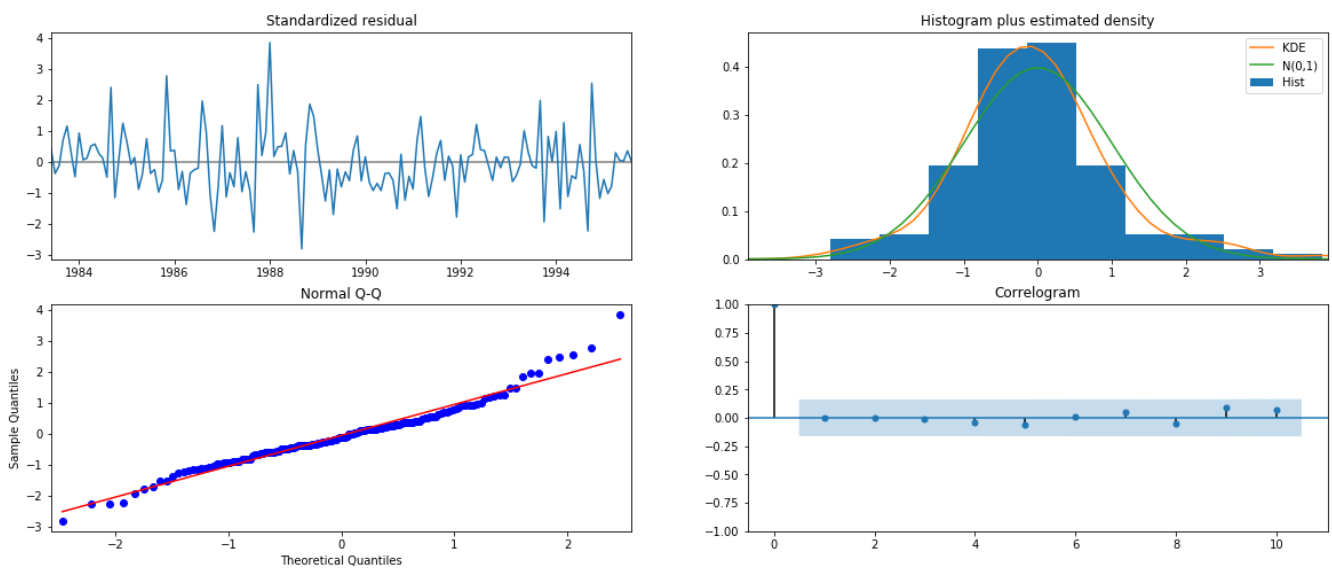
Plot the forecast along with the confidence band(Rose)



Building the most optimum model on the Full Data.(Sparkling)

```
                                     Statespace Model Results
==========================================================================================
====
Dep. Variable:                                Sparkling   No. Observations:
187
Model:                 SARIMAX(2, 1, 2)x(2, 1, 2, 12)   Log Likelihood                -108
5.726
Date:                                Sat, 12 Sep 2020   AIC                            218
9.453
Time:                                        20:22:11   BIC                            221
6.367
Sample:                                    01-31-1980   HQIC                           220
0.388
                                         - 07-31-1995
Covariance Type:                                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6655      0.275     -2.421      0.015      -1.204      -0.127
ar.L2         -0.0111      0.113     -0.099      0.921      -0.232       0.210
ma.L1         -0.1696      0.260     -0.652      0.515      -0.680       0.340
ma.L2         -0.7336      0.242     -3.032      0.002      -1.208      -0.259
ar.S.L12       0.0659      0.370      0.178      0.859      -0.659       0.791
ar.S.L24       0.2880      0.140      2.060      0.039       0.014       0.562
ma.S.L12      -0.7496      0.997     -0.752      0.452      -2.704       1.205
ma.S.L24      -0.2228      0.327     -0.682      0.495      -0.863       0.418
sigma2      1.332e+05   9.92e+04      1.343      0.179   -6.12e+04    3.28e+05
===================================================================================
Ljung-Box (Q):                       17.23   Jarque-Bera (JB):                26.11
Prob(Q):                              1.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.89   Skew:                             0.56
Prob(H) (two-sided):                  0.68   Kurtosis:                         4.74
===================================================================================

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

Full Data plot diagnostics(Sparkling)



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

future (Sparkling)

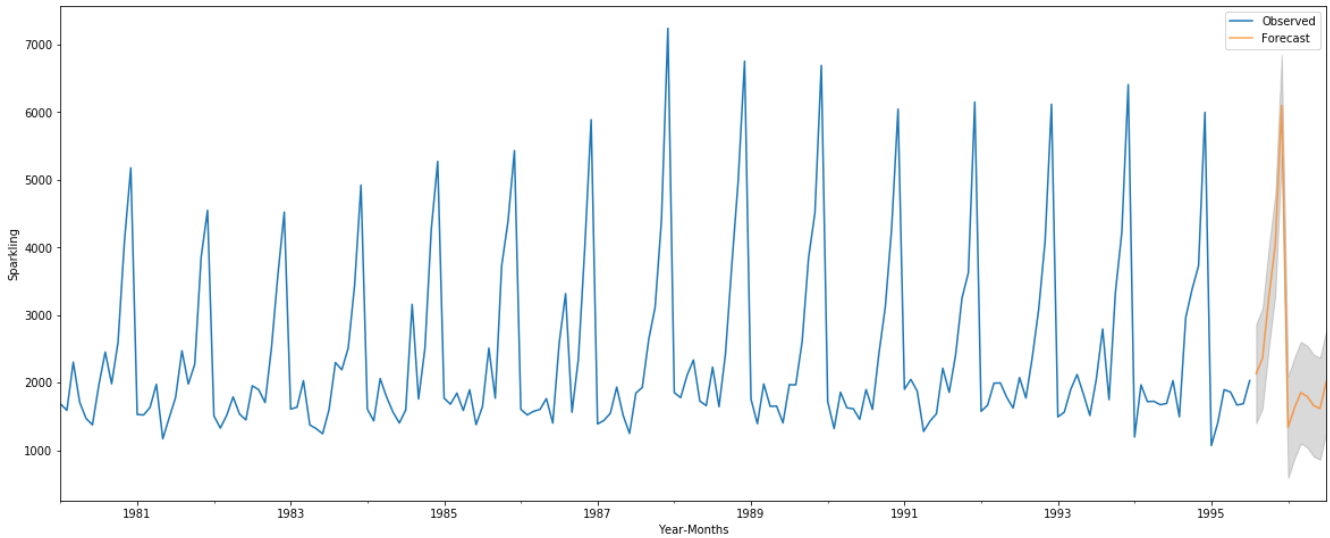
Out[343]:

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	2132.779263	373.934142	1399.881812	2865.676714
1995-09-30	2359.397464	379.075035	1616.424048	3102.370881
1995-10-31	3266.882699	379.150345	2523.761677	4010.003721
1995-11-30	4040.248869	381.486889	3292.548305	4787.949432
1995-12-31	6101.119275	381.594085	5353.208611	6849.029939

RMSE of the Full Model(Sparkling)

```
RMSE of the Full Model 568.7891702213292
```

Plot the forecast along with the confidence band(Sparkling)



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

Comment on the model and findings

- In this particular we have built several models and went through a model building exercise. This particular exercise has given us an idea as to which particular model gives us the least error on our test set for this data
- In order to forecast using any of the models built, we have built the models again (this time on the complete data) with the same parameters.
- we have built below 2 models which gaves us the best accuracy (least RMSE) in the case of Rose and Sparkling TimeSeries
- Best model is the Triple Exponential Smoothing with the parameters α = .1, β = 0.2 and γ = 0.2. in Rose Time Series
- Best Model is ManualSARIMA(2,1,2)(2,1,2) as far as Sparkling TIme Series is concerned

Rose Business Insights and Recommendations

- There is upward trend in every Decemember across the years

- Janurary and April having downword trend rest other months having zig zag trend across the years
- In the Month of janurary and April company can launch various Marketing and Advertisement campaign like give the discount to loyal customer ,Attractive Gift /crediti point to the customer who are recommending Rose wine to other people

Sparkling Business Insights and Recommendations

- In the Case of Sparkling September to Decemember trend is upward and This trend is accross the years . which is good sign .
- January having downward trend accross the years and other Months having zig zag trend across the various years.
- In the january Month Company can launch loyal customer plan underwhich company can offer some gifts / discount offers.
- If the customer take the wine in bulk during the downtrend month Attractive discount can be given