Name: Vikram Radhakrishnan

Business Report : Time Series Forecasting

Date : 23rd August, 2020

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

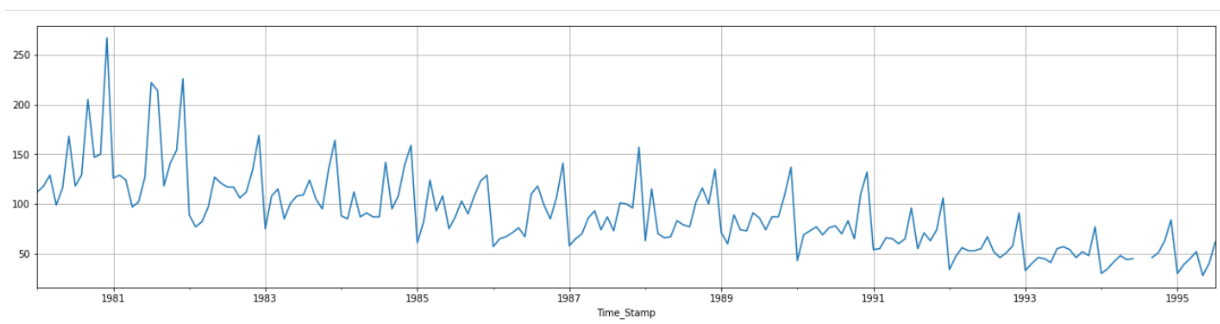
Two csv files have been provided, for which two separate jupyter notebooks have been created to work with each. In order to read them, the following procedure has been followed :

a) The files are read into a Pandas dataframe
b) A separate time series corresponding to the timeline in the data is created, which is called 'Timestamp' and added to the dataframe
c) The timestamp column is made into the index and the original 'YearMonth' column is dropped.

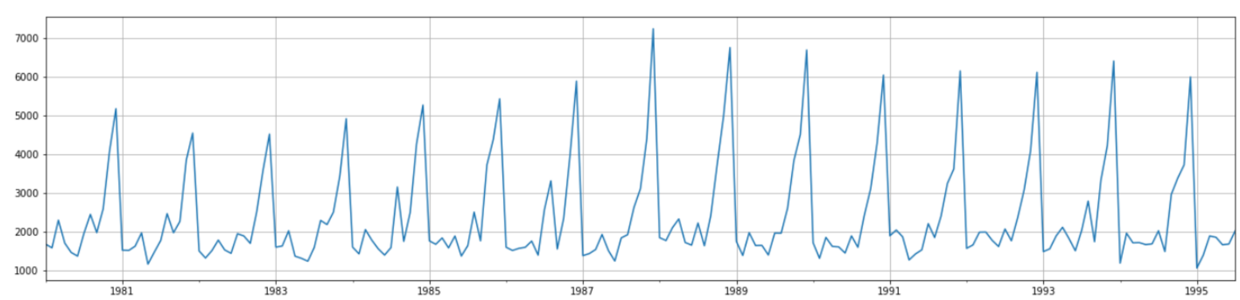
Once this procedure is followed, the format of the data will appear as follows:

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 SALES OVER THE YEARS

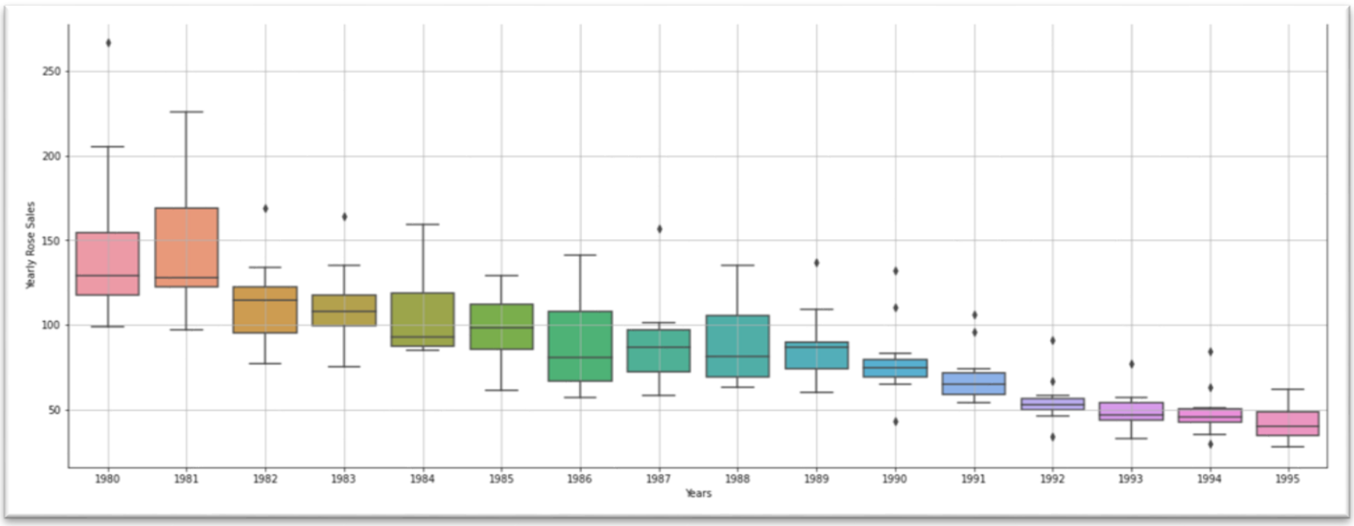


SPARKLING WINE SALES OVER THE YEARS

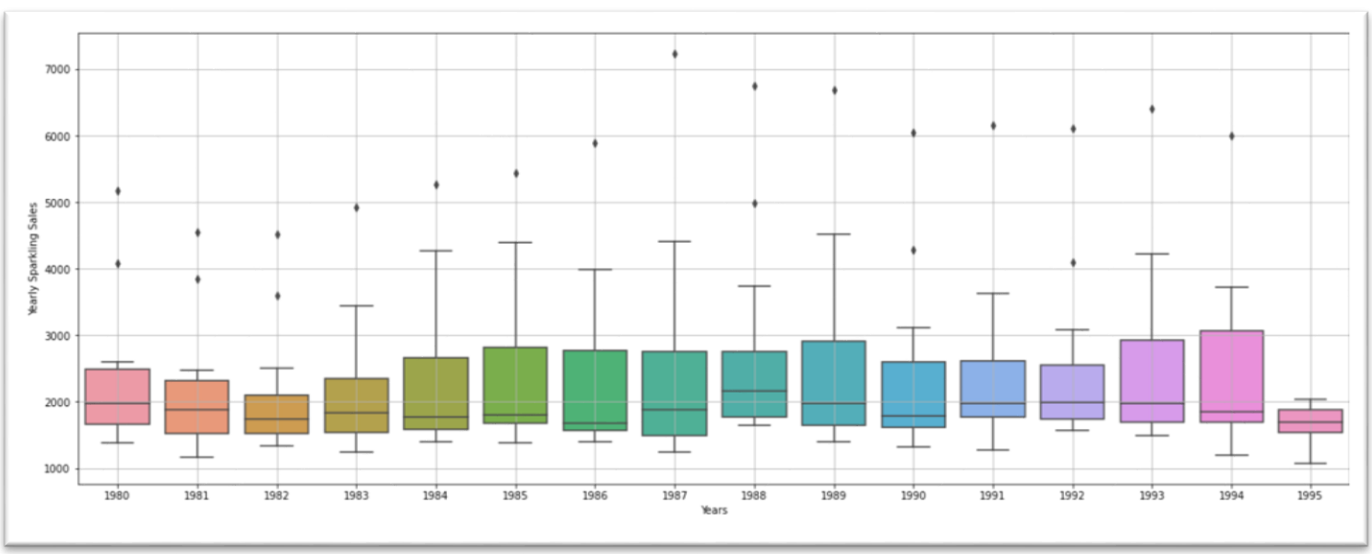


2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

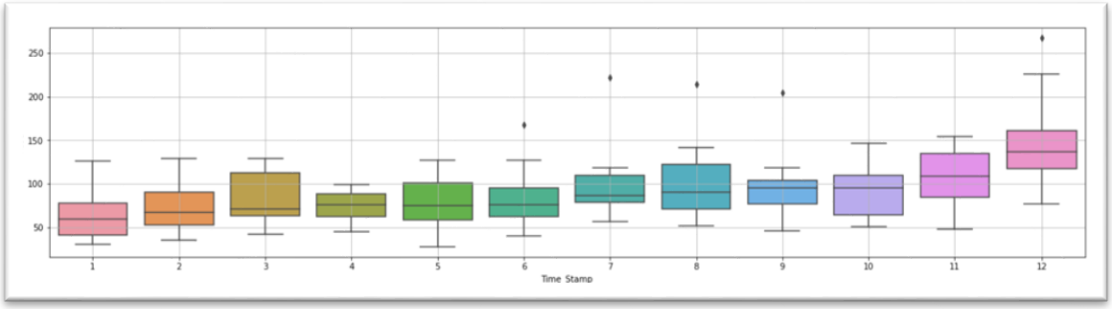
YEARLY SALES DATA : ROSE (Box Plot)



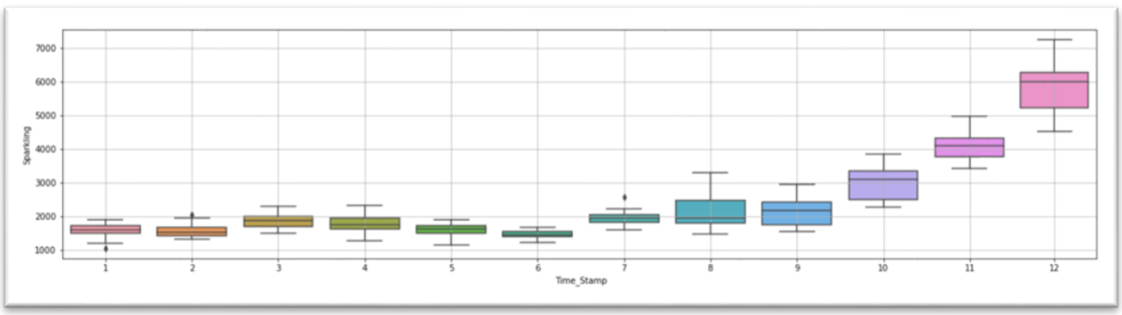
YEARLY SALES DATA : SPARKLING (Box Plot)



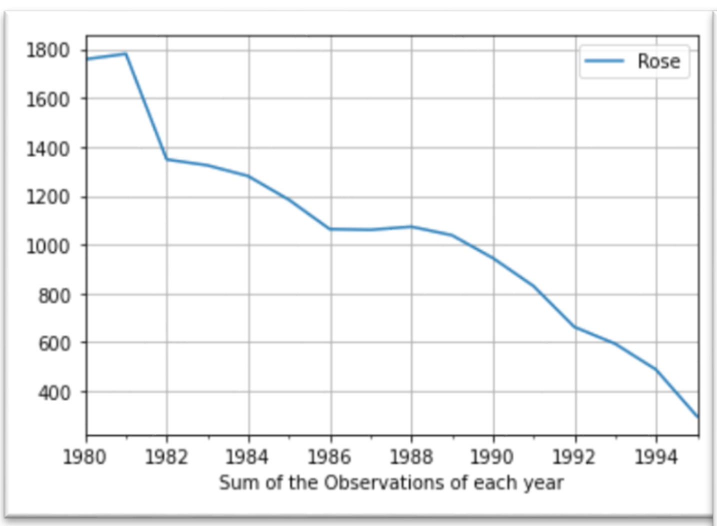
MONTHLY SALES DATA (MEAN ACROSS YEARS) : ROSE (Box Plot)

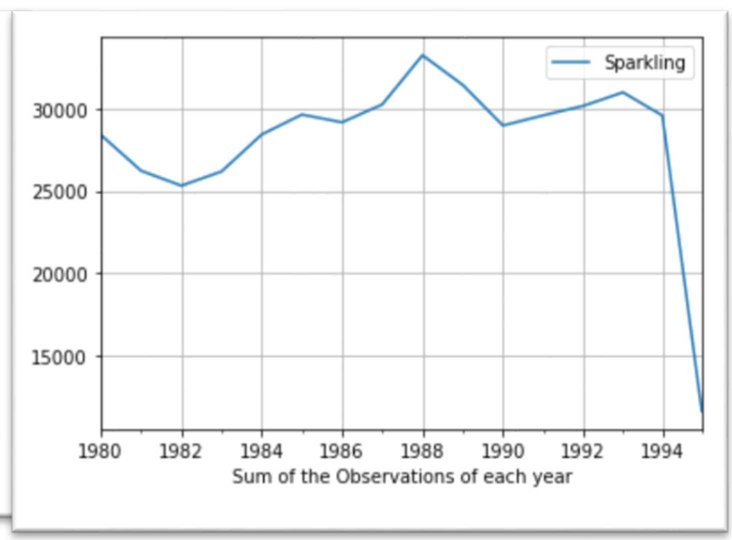


MONTHLY SALES DATA (MEAN ACROSS YEARS) : SPARKLING (Box Plot)



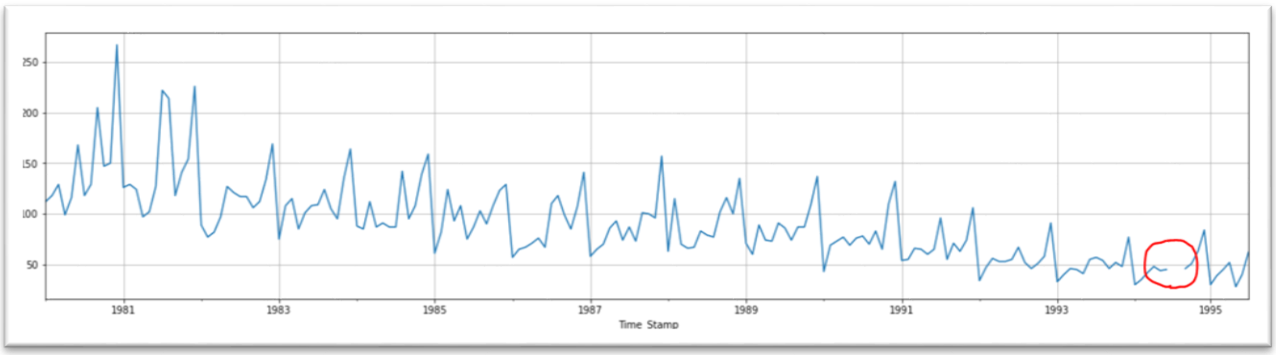
COMPARISON OF TOTAL SALES PER YEAR – ROSE vs SPARKLING





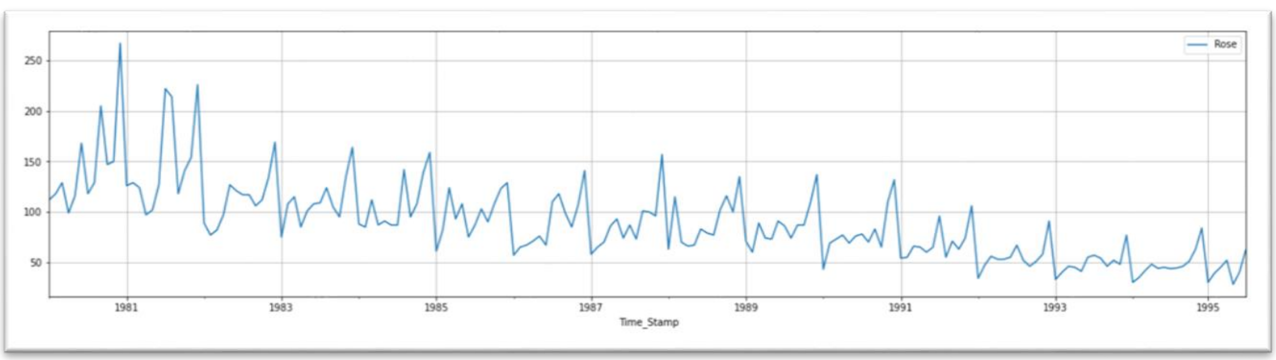
Similarly, many such plots have been checked to observe the general behavior of the data. The interested reader may go through the Jupyter Notebooks for more/different plots using the same data.

HANDLING MISSING DATA



It is observed that in the year 1994, two entries are missing for Rose Wine. These are interpolated using a 3rd order spline to estimate the two missing values which are used subsequently.

```
df1['1994']=df1.interpolate(method='spline',order=3)['1994']
```



After populating the missing data, the plot looks continuous as above.

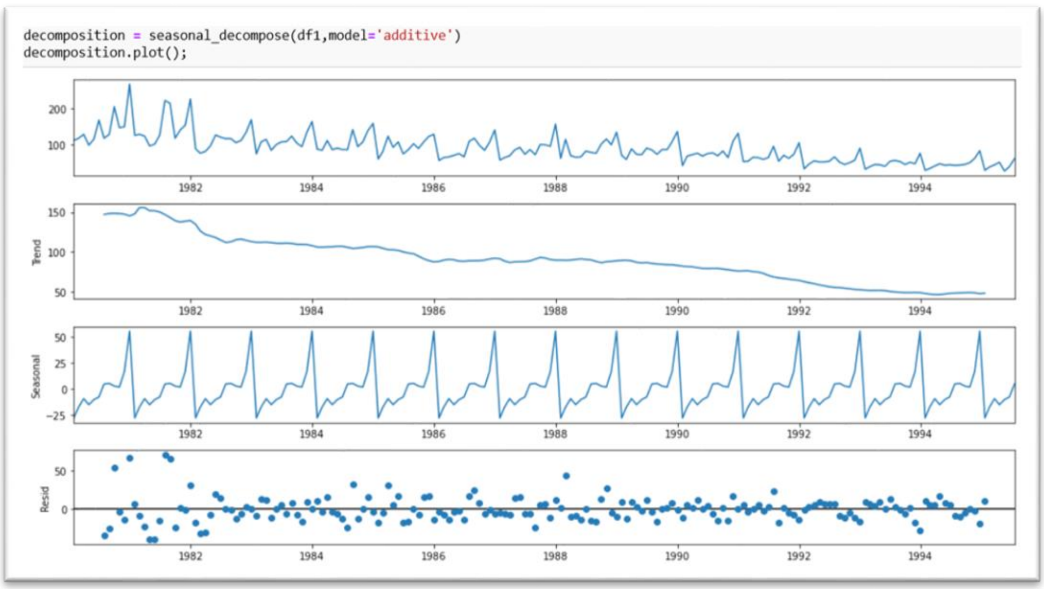
The sparkling wine data does not seem to have any such issues with missing data.

DECOMPOSITION OF DATA:

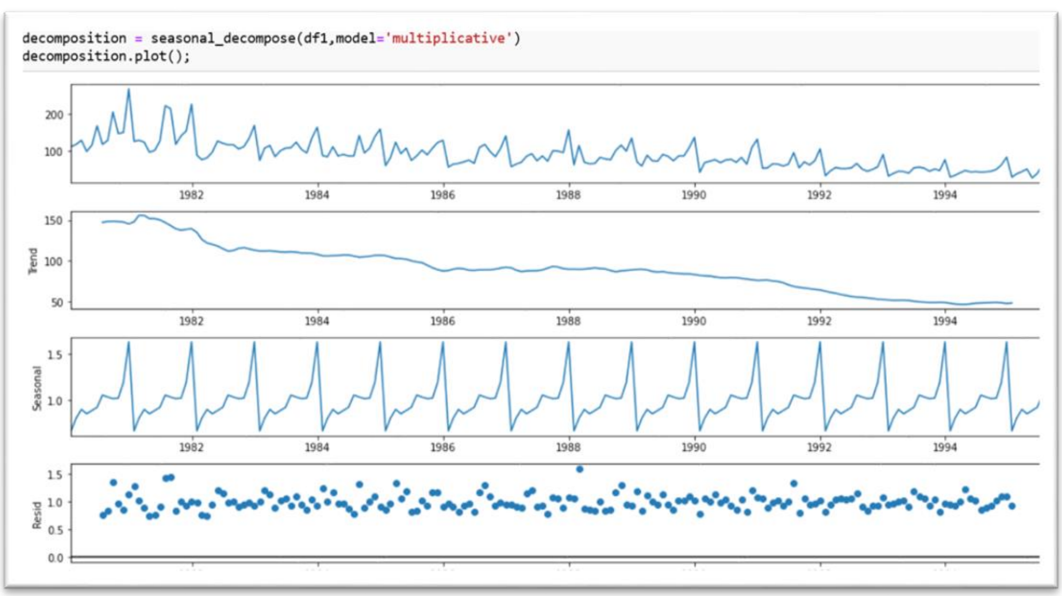
In order to understand the magnitude and nature of TREND and SEASONALITY of the data, decomposition of the data is performed for both Rose as well as Sparkling Wine.

a) **ROSE WINE :** Multiplicative decomposition is used generally when there is a strong increasing or decreasing component of seasonality that changes year-on-year (or as per corresponding observation period). In our case, we do not observe such a behavior in the magnitude of seasonality varying multiplicatively, so ADDITIVE decomposition would be the better choice.

Additive Decomposition (Rose)



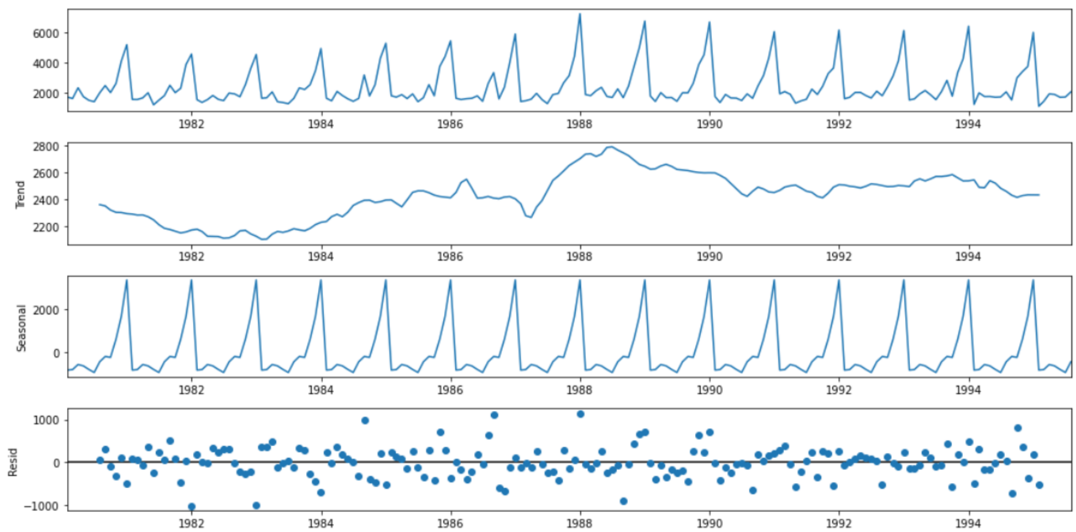
Multiplicative Decomposition (Rose)



b) **SPARKLING WINE :** Multiplicative decomposition is used generally when there is a strong increasing or decreasing component of seasonality that changes year-on-year (or as per corresponding observation period). In this case, we observe a small trend in the seasonality magnitude and also observe that the residual are more or less constant at a value of approximately 1. So MULTIPLICATIVE decomposition would be the better choice.

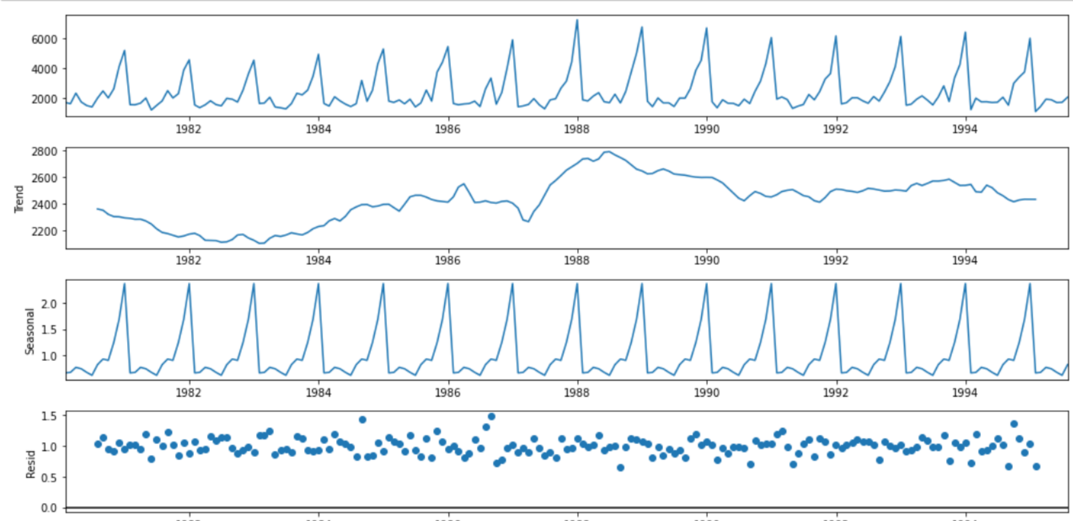
Additive Decomposition (Sparkling)

```
decomposition = seasonal_decompose(df1,model='additive')
decomposition.plot();
```



Multiplicative Decomposition (Sparkling)

```
decomposition = seasonal_decompose(df1,model='multiplicative')
decomposition.plot();
```

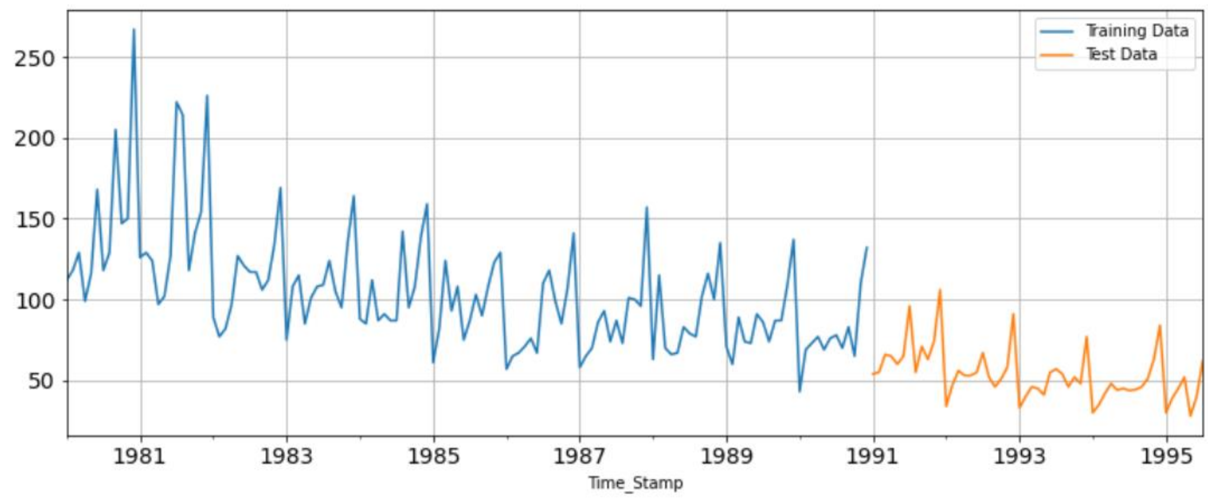


3. Split the data into training and test. The test data should start in 1991.

The data is split into test and training data based on the year. Years before 1991 are in TRAIN and the year 1991 and hence are in TEST.

```
train=df1[df1.index.year < 1991]
test=df1[df1.index.year >= 1991]
```

EXAMPLE OF TEST AND TRAINING DATA PLOTTED TOGETHER FOR ROSE WINE



4. Build various exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other models such as regression, naïve forecast models, simple average models etc. should also be built on the training data and check the performance on the test data using RMSE. - Please do try to build as many models as possible and as many iterations of models as possible with different parameters.

The following are the models tried :

a) Naïve Model
b) Simple Average Model
c) Linear Regression on Time
d) Simple Exponential Smoothing (with Alpha=1)
e) Simple Exponential Smoothing (Iterated for least RMSE)
f) Double Exponential Smoothing (Optimized Fit)
g) Double Exponential Smoothing (Looping through parameters for least RMSE)
h) Triple Exponential Smoothing (Optimal Fit)
i) Triple Exponential Smoothing (Looping through parameters for least RMSE)
j) Moving Averages with 2 point, 4 point, 6 point and 9 point trailing

ROSE WINE

	Test RMSE	Test MAPE
NaiveModel	79.778066	145.35
SimpleAverageModel	53.521557	95.13
RegressionOnTime	15.291460	22.94
Alpha=1:SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.07,SimpleExponentialSmoothing	43.768690	76.74
Alpha=0.158,Beta=0.158:DoubleExponentialSmoothing	70.642717	120.47
Alpha=0.11,Beta=0.05,Gamma=0.000: TripleExponentialSmoothing	17.445169	29.01
Alpha=0.04,Beta=0.47,DoubleExponentialSmoothing	450.311022	831.28
Alpha=0.1,Beta=0.2,Gamma=0.2:Triple ExponentialSmoothing	9.665739	14.08
2pointTrailingMovingAverage	11.530180	13.60
4pointTrailingMovingAverage	14.462330	19.59
6pointTrailingMovingAverage	14.586916	20.83
9pointTrailingMovingAverage	14.740112	21.13

SPARKLING WINE

	Test RMSE	Test MAPE
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
RegressionOnTime	1389.135175	50.15
Alpha=1:SimpleExponentialSmoothing	1275.081823	38.90
Alpha=0.02,SimpleExponentialSmoothing	1279.495201	40.97
Alpha=0.158,Beta=0.158:DoubleExponentialSmoothing	3850.989796	152.06
Alpha=0.65,Beta=0:DoubleExponentialSmoothing	3850.989796	152.06
Alpha=0.15,Beta=5.31,Gamma=0.37: TripleExponentialSmoothing	383.157627	11.91
Alpha=0.02,Beta=0.50,DoubleExponentialSmoothing	6336.376572	257.11
Alpha=0.4,Beta=0.1,Gamma=0.2:Triple ExponentialSmoothing	336.715300	110.56
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86

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

The **Augmented Dickey Fuller Test (ADF)** determines whether a time series is non-stationary.

NULL HYPOTHESIS Ho : The time series is non-stationary

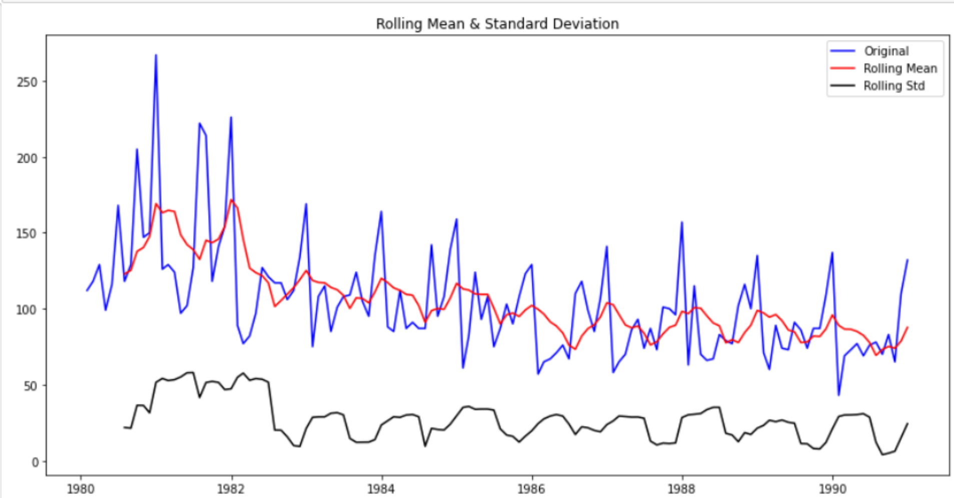
ALT HYPOTHESIS H1 : Time series is stationary

(Rejection of Null Hypothesis means time series is stationary)

A) ROSE WINE :

TRAINING DATA :

```
test_stationarity(train['Rose'])
```

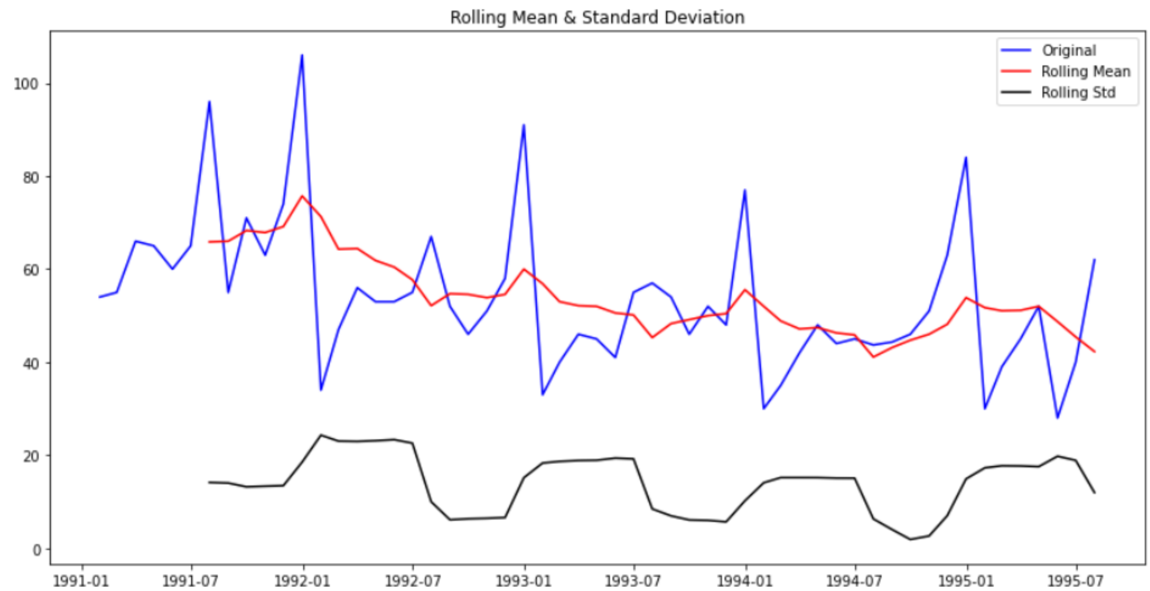



```
Results of Dickey-Fuller Test:
Test Statistic                 -2.164250
p-value                         0.219476
#Lags Used                     13.000000
Number of Observations Used   118.000000
Critical Value (1%)            -3.487022
Critical Value (5%)            -2.886363
Critical Value (10%)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

We observe a P value greater than alpha(0.05). So the training data appears to be stationary as per the Alternate Hypothesis.

TEST DATA :

```
: test_stationarity(test['Rose'])
```



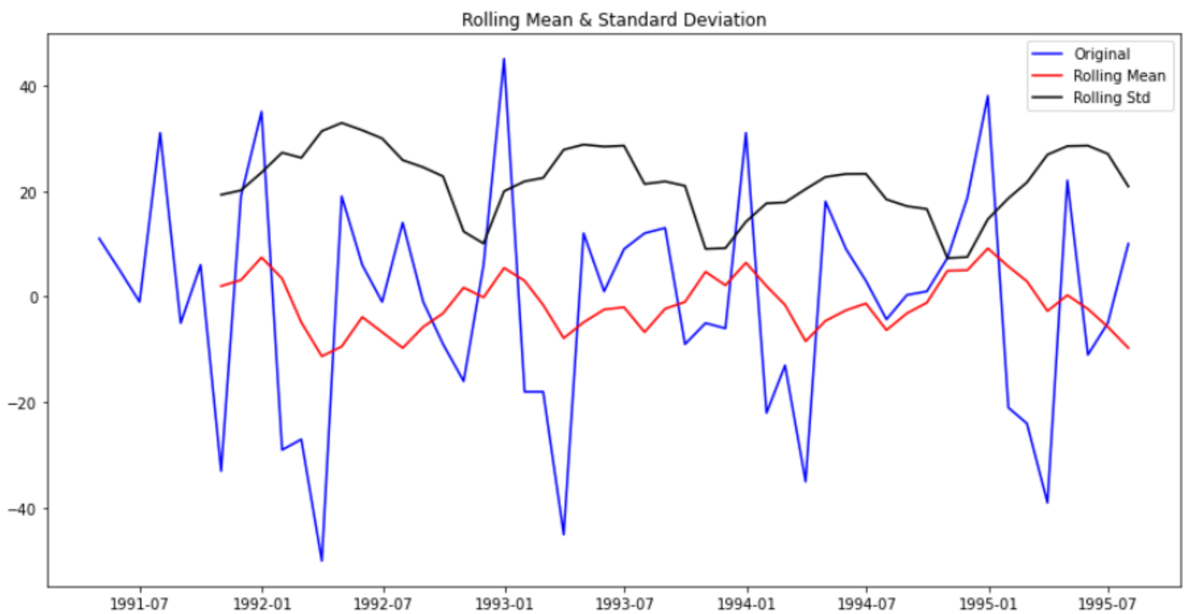

```
Results of Dickey-Fuller Test:
Test Statistic                 -4.333930
p-value                         0.000388
#Lags Used                     11.000000
Number of Observations Used    43.000000
Critical Value (1%)            -3.592504
Critical Value (5%)            -2.931550
Critical Value (10%)           -2.604066
dtype: float64
```

The test data appears to be non-stationary since P value is less than 0.05.

DIFFERENCING is one way to make a non-stationary time series stationary — compute the differences between consecutive observations.

In this case, we apply differencing until the P value becomes less than 0.05, which in our case takes three times.

```
test_stationarity(test.diff(3).dropna())
```

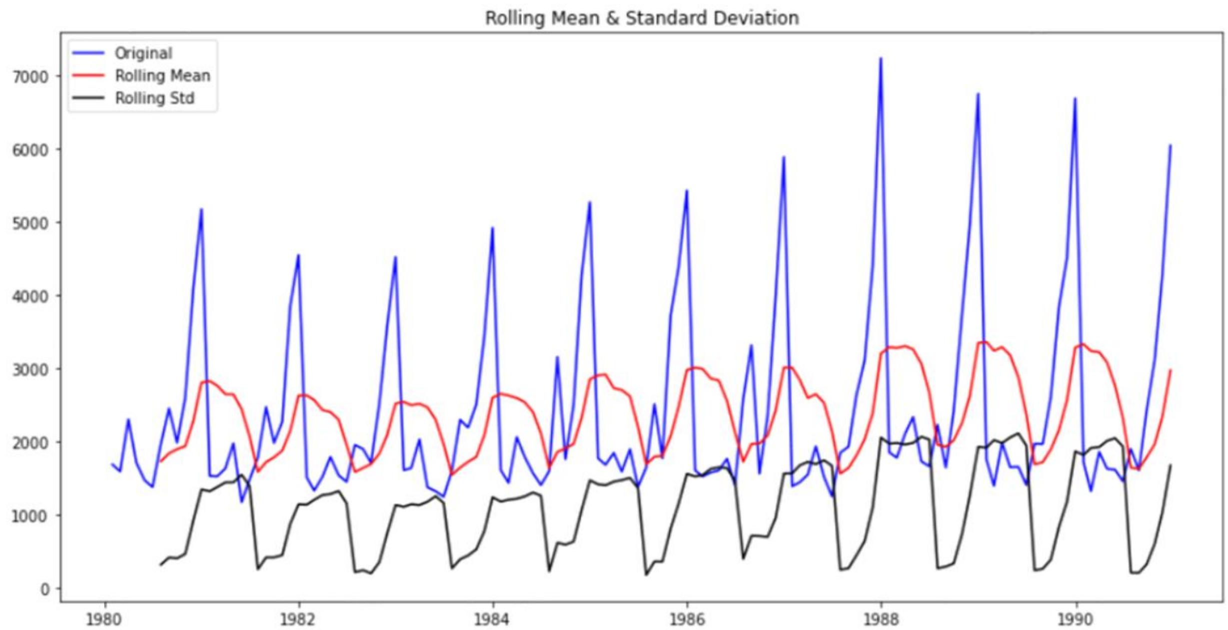



```
Results of Dickey-Fuller Test:
Test Statistic                 -3.036974
p-value                         0.031583
#Lags Used                      9.000000
Number of Observations Used    42.000000
Critical Value (1%)            -3.596636
Critical Value (5%)            -2.933297
Critical Value (10%)           -2.604991
```

Here we apply differencing to make the series stationary.

B) SPARKLING WINE :

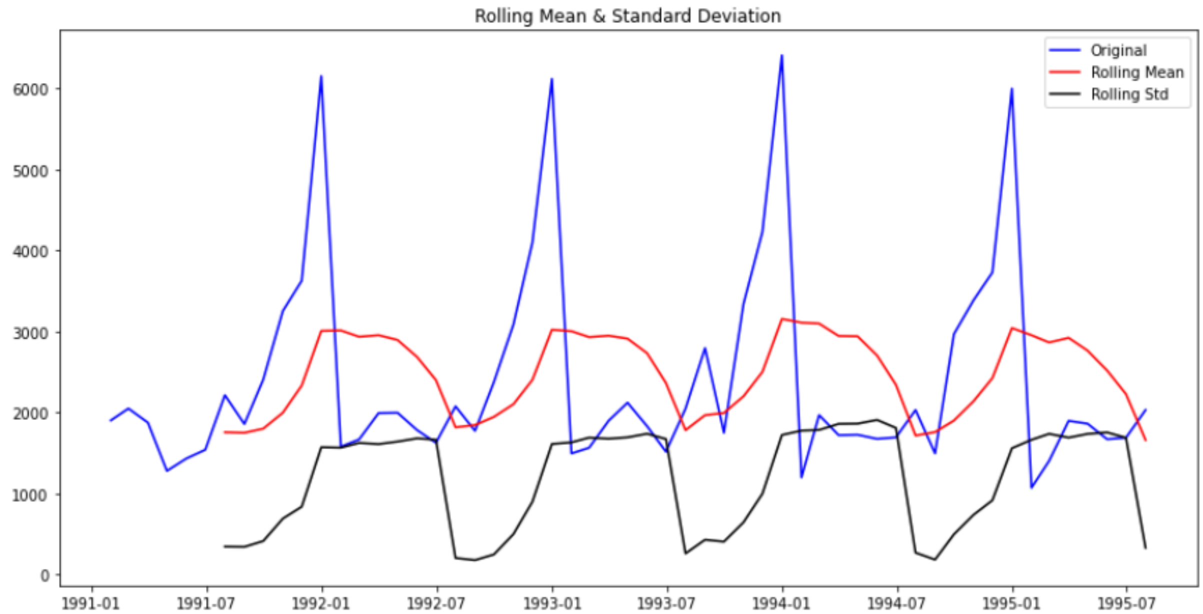
```
test_stationarity(train['Sparkling'])
```




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

We observe a P value greater than alpha(0.05). So the training data appears to be stationary as per the Alternate Hypothesis.

```
: test_stationarity(test['Sparkling'])
```




```
Results of Dickey-Fuller Test:
Test Statistic                 -1.790189
p-value                         0.385343
#Lags Used                     11.000000
Number of Observations Used    43.000000
Critical Value (1%)            -3.592504
Critical Value (5%)            -2.931550
Critical Value (10%)           -2.604066
dtype: float64
```

Test Data also shows a P value greater than 0.05. Therefore the test data exhibits stationarity.

6. *Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE.*

The p,d,q values of the ARIMA are varied between 0 and 2 to perform a design of experiments (DOE) through itertools. The values are then sorted to reveal the combination that provides the lowest AIC value. This value is used to construct the best ARIMA model in this case.

A) ROSE :

```
ARIMA_AIC.sort_values(by='AIC',ascending=True)
```

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

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   Rose   No. Observations:                  132
Model:                 ARIMA(0, 1, 2)   Log Likelihood                -636.836
Date:                Sun, 23 Aug 2020   AIC                           1279.672
Time:                        21:27:40   BIC                           1288.297
Sample:                    01-31-1980   HQIC                          1283.176
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.6970      0.072     -9.689      0.000      -0.838      -0.556
ma.L2         -0.2042      0.073     -2.794      0.005      -0.347      -0.061
sigma2       965.8407     88.305     10.938      0.000     792.766    1138.915
===================================================================================
Ljung-Box (Q):                      112.54   Jarque-Bera (JB):                39.24
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.36   Skew:                             0.82
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.13
===================================================================================
```

Similarly the SARIMA model is also tried out by varying both the seasonal and non-seasonal variables in an iterative fashion as above and checked for least AIC.

	param	seasonal	AIC
26	(0, 1, 2)	(2, 0, 2, 12)	887.937509
53	(1, 1, 2)	(2, 0, 2, 12)	889.871768
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

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                Rose   No. Observations:                  132
Model:             SARIMAX(0, 1, 2)x(2, 0, 2, 12)   Log Likelihood                -436.969
Date:                             Sun, 23 Aug 2020   AIC                            887.938
Time:                                     21:33:45   BIC                            906.448
Sample:                                 01-31-1980   HQIC                           895.437
                                      - 12-31-1990
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8427    189.452     -0.004      0.996    -372.161     370.476
ma.L2         -0.1573     29.764     -0.005      0.996     -58.493      58.179
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       251.3136   4.76e+04      0.005      0.996   -9.31e+04    9.36e+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
===================================================================================
```

	RMSE	MAPE
ARIMA(0,1,2)	37.368538	64.98
SARIMA(0, 1, 2)(2, 0, 2, 12)	26.992037	46.75

Finally, its found that the SARIMA model performs better in our case with a lower RMSE value, and hence can be chosen.

B) SPARKLING :

	param	AIC
8	(2, 1, 2)	2213.509212
7	(2, 1, 1)	2233.777626
2	(0, 1, 2)	2234.408323
5	(1, 1, 2)	2234.527200
4	(1, 1, 1)	2235.755095
6	(2, 1, 0)	2260.365744
1	(0, 1, 1)	2263.060016
3	(1, 1, 0)	2266.608539
0	(0, 1, 0)	2267.663036

```
                                      SARIMAX Results
==========================================================================================
Dep. Variable:                             Sparkling   No. Observations:              132
Model:             SARIMAX(1, 1, 2)x(1, 0, 2, 12)   Log Likelihood               -770.792
Date:                            Sun, 23 Aug 2020   AIC                          1555.585
Time:                                    22:23:53   BIC                          1574.096
Sample:                                01-31-1980   HQIC                         1563.084
                                     - 12-31-1990
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6283      0.253     -2.483      0.013      -1.124      -0.132
ma.L1         -0.1030      0.223     -0.463      0.643      -0.539       0.333
ma.L2         -0.7291      0.151     -4.813      0.000      -1.026      -0.432
ar.S.L12       1.0438      0.014     72.785      0.000       1.016       1.072
ma.S.L12      -0.5552      0.098     -5.661      0.000      -0.747      -0.363
ma.S.L24      -0.1352      0.120     -1.131      0.258      -0.369       0.099
sigma2      1.505e+05   2.03e+04      7.410      0.000    1.11e+05     1.9e+05
===================================================================================
Ljung-Box (Q):                       23.01   Jarque-Bera (JB):                11.63
Prob(Q):                              0.99   Prob(JB):                         0.00
Heteroskedasticity (H):               1.47   Skew:                             0.36
Prob(H) (two-sided):                  0.26   Kurtosis:                         4.47
===================================================================================
```

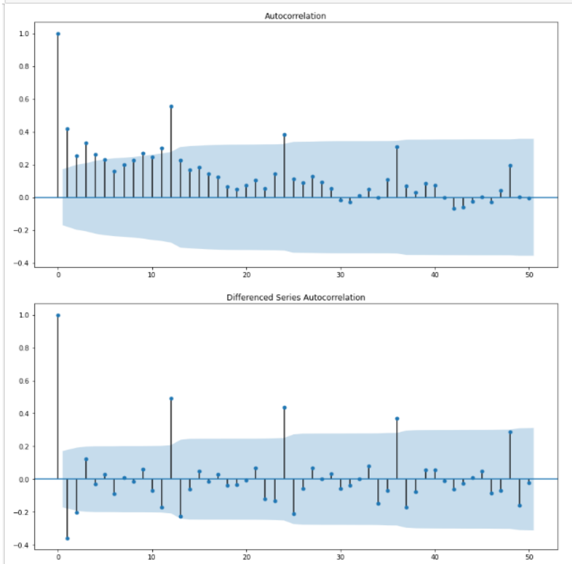
The ARIMA/SARIMA values are plotted and it is found that the SARIMA model performs much better in terms of having a lower RMSE value, compared to ARIMA.

	RMSE	MAPE
ARIMA(2,1,2)	1299.980204	43.20
SARIMA(0, 1, 2)(2, 0, 2, 12)	527.571342	18.85

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.

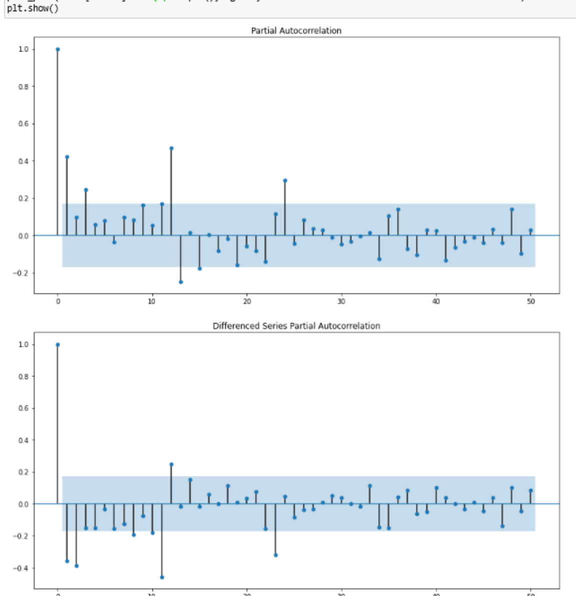
ROSE: ACF and Differenced ACF

```
plot_acf(train['Rose'],lags=50)
plot_acf(train['Rose'].diff(1).dropna(),lags=50,title='Differenced Series Autocorrelation')
plt.show()
```



ROSE: PACF and Differenced PACF

```
plot_pacf(train['Rose'],lags=50)
plot_pacf(train['Rose'].diff(1).dropna(),lags=50,title='Differenced Series Partial Autocorrelation')
plt.show()
```



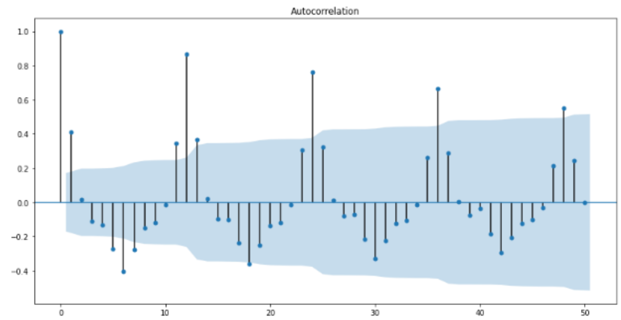
The PACF provides an estimate of a good 'p' value and the ACF for the 'q' value. Differencing to make the data stationary gives an estimate of the value of 'd'. Similarly, through differenced measures, we can get the values of P,D,Q.

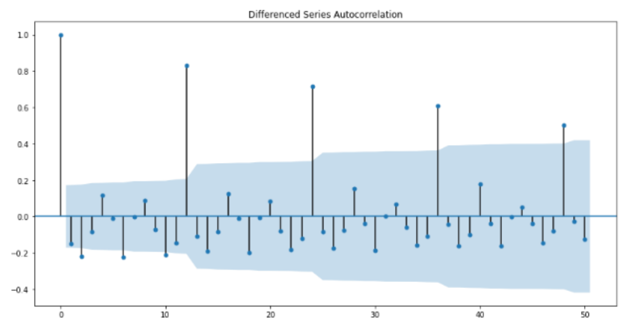
In our case, we go with p=4, d=1,q=1 for non-seasonal values, and P=1,D=1,Q=0 for seasonal values and estimate the value of RMSE for this condition. It is found to be lower than the earlier SARIMA and ARIMA as per comparative table below.

```
                                      SARIMAX Results
==========================================================================================
Dep. Variable:                                 Rose   No. Observations:                132
Model:             SARIMAX(4, 1, 1)x(1, 1, [], 12)   Log Likelihood             -445.457
Date:                             Sun, 23 Aug 2020   AIC                         904.913
Time:                                     21:45:34   BIC                         923.356
Sample:                                 01-31-1980   HQIC                        912.383
                                      - 12-31-1990
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1406      0.130      1.086      0.278      -0.113       0.395
ar.L2         -0.0685      0.111     -0.619      0.536      -0.285       0.148
ar.L3         -0.1384      0.112     -1.234      0.217      -0.358       0.081
ar.L4         -0.0081      0.108     -0.074      0.941      -0.221       0.204
ma.L1         -0.8767      0.085    -10.321      0.000      -1.043      -0.710
ar.S.L12      -0.3807      0.056     -6.845      0.000      -0.490      -0.272
sigma2       332.8673     44.663      7.453      0.000     245.329     420.406
===================================================================================
Ljung-Box (Q):                       27.75   Jarque-Bera (JB):                 0.43
Prob(Q):                              0.93   Prob(JB):                         0.81
click to hide dasticity (H):          0.51   Skew:                             0.07
         (two-sided):                 0.05   Kurtosis:                         3.28
===================================================================================
```

SPARKLING : ACF and Differenced ACF

```
plot_acf(train['Sparkling'],lags=50)
plot_acf(train['Sparkling'].diff(1).dropna(),lags=50,title='Differenced Series Autocorrelation')
plt.show()
```

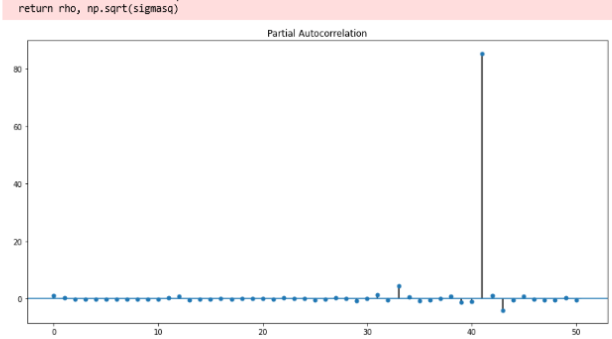


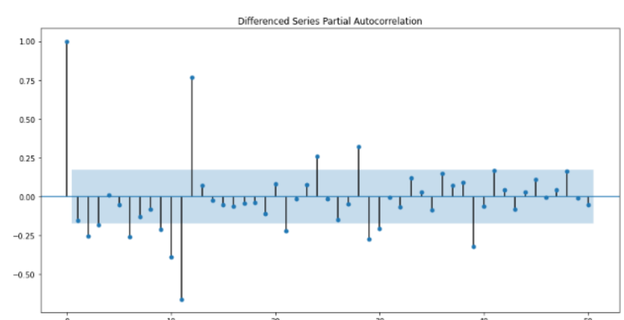


SPARKLING : PACF and Differenced PACF

```
plot_pacf(train['Sparkling'],lags=50)
plot_pacf(train['Sparkling'].diff().dropna(),lags=50,title='Differenced Series Partial Autocorrelation')
plt.show()
```

```
C:\Users\v2n\AppData\Local\Continuum\anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1406: RuntimeW
nvalid value encountered in sqrt
  return rho, np.sqrt(sigmasq)
```





```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                            Sparkling   No. Observations:                  132
Model:             SARIMAX(0, 1, 1)x(0, 1, [], 12)   Log Likelihood                -878.328
Date:                              Sun, 23 Aug 2020   AIC                           1760.657
Time:                                      22:40:08   BIC                           1766.181
Sample:                                  01-31-1980   HQIC                          1762.899
                                       - 12-31-1990
Covariance Type:                                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.9200      0.038    -24.315      0.000      -0.994      -0.846
sigma2      1.919e+05   1.86e+04     10.340      0.000    1.56e+05    2.28e+05
===================================================================================
Ljung-Box (Q):                       50.71   Jarque-Bera (JB):                16.55
Prob(Q):                              0.12   Prob(JB):                         0.00
Heteroskedasticity (H):               2.11   Skew:                             0.15
Prob(H) (two-sided):                  0.02   Kurtosis:                         4.82
===================================================================================
```

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

ROSE (ARIMA/SARIMA) :

	RMSE
ARIMA(0,1,2)	37.368538
SARIMA(0, 1, 2)(2, 0, 2, 12)	26.992037
SARIMA(4, 1, 1)(1, 1, 0, 12)	15.603040

ROSE (Other Models):

	Test RMSE	Test MAPE
NaiveModel	79.778066	145.35
SimpleAverageModel	53.521557	95.13
RegressionOnTime	15.291460	22.94
Alpha=1:SimpleExponentialSmoothing	36.858586	64.05
Alpha=0.07,SimpleExponentialSmoothing	43.768690	76.74
Alpha=0.158,Beta=0.158:DoubleExponentialSmoothing	70.642717	120.47
Alpha=0.11,Beta=0.05,Gamma=0.000: TripleExponentialSmoothing	17.445169	29.01
Alpha=0.04,Beta=0.47,DoubleExponentialSmoothing	450.311022	831.28
Alpha=0.1,Beta=0.2,Gamma=0.2:Triple ExponentialSmoothing	9.665739	14.08
2pointTrailingMovingAverage	11.530180	13.60
4pointTrailingMovingAverage	14.462330	19.59
6pointTrailingMovingAverage	14.586916	20.83
9pointTrailingMovingAverage	14.740112	21.13

The SARIMA and the moving average models provide a low RMSE value and can be used for future predictions on sales for the next 12 months.

SPARKLING (ARIMA/SARIMA) :

	RMSE
ARIMA(2,1,2)	1299.980204
SARIMA(0, 1, 2)(2, 0, 2, 12)	527.571342
SARIMA(0, 1, 1)(0, 1, 0, 12)	681.719781

	Test RMSE	Test MAPE
NaiveModel	3864.279352	152.87
SimpleAverageModel	1275.081804	38.90
RegressionOnTime	1389.135175	50.15
Alpha=1:SimpleExponentialSmoothing	1275.081823	38.90
Alpha=0.02,SimpleExponentialSmoothing	1279.495201	40.97
Alpha=0.65,Beta=0:DoubleExponentialSmoothing	3850.989796	152.06
Alpha=0.15,Beta=5.31,Gamma=0.37: TripleExponentialSmoothing	383.157627	11.91
Alpha=0.02,Beta=0.50,DoubleExponentialSmoothing	6336.376572	257.11
Alpha=0.4,Beta=0.1,Gamma=0.2:Triple ExponentialSmoothing	336.715300	110.56
2pointTrailingMovingAverage	813.400684	19.70
4pointTrailingMovingAverage	1156.589694	35.96
6pointTrailingMovingAverage	1283.927428	43.86
9pointTrailingMovingAverage	1346.278315	46.86

The Triple Exponential model with an RMSE of 337 units appears to be the best model, but provides unexpected results when a prediction is done. For ARIMA/SARIMA, the SARIMA model with an RMSE of 528 units appears suitable with a prediction on expected lines.

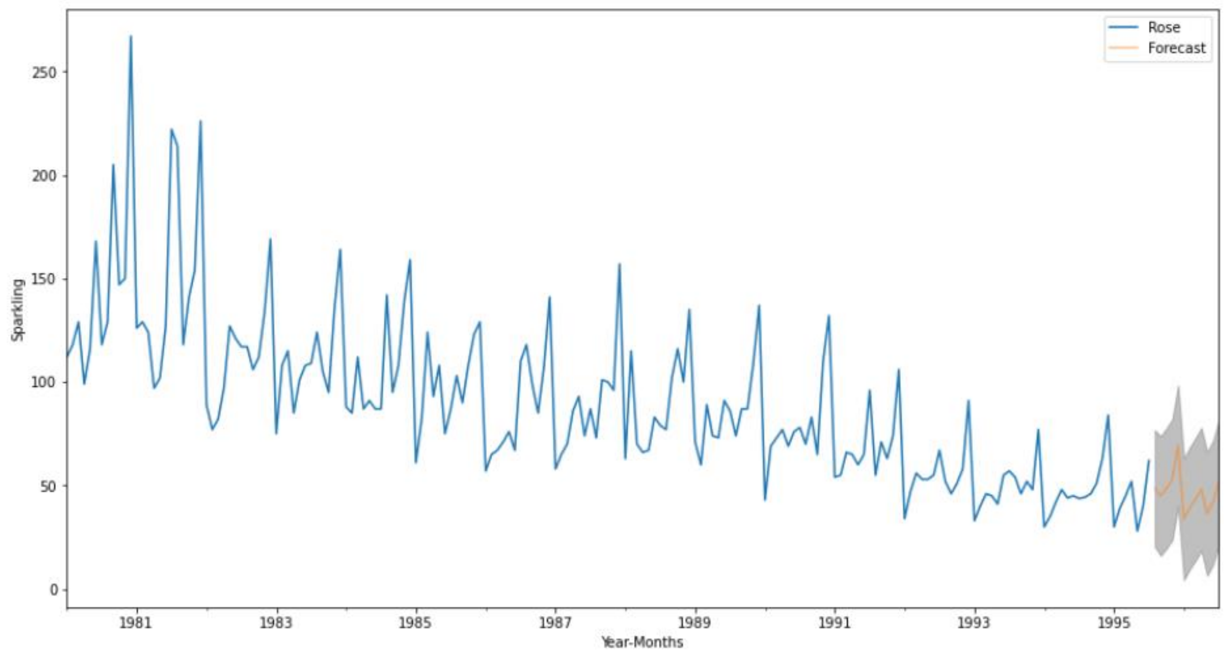
9. Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.

ROSE WINE (PREDICTION) :

Orange line shows prediction, gray line shows the confidence interval.

The SARIMA model shown below is utilized for prediction owing to its low RMSE score of ~15 units and owing to the ability of the model to capture both seasonal and non-seasonal effects.

SARIMA(4, 1, 1)(1, 1, 0, 12)	15.603040

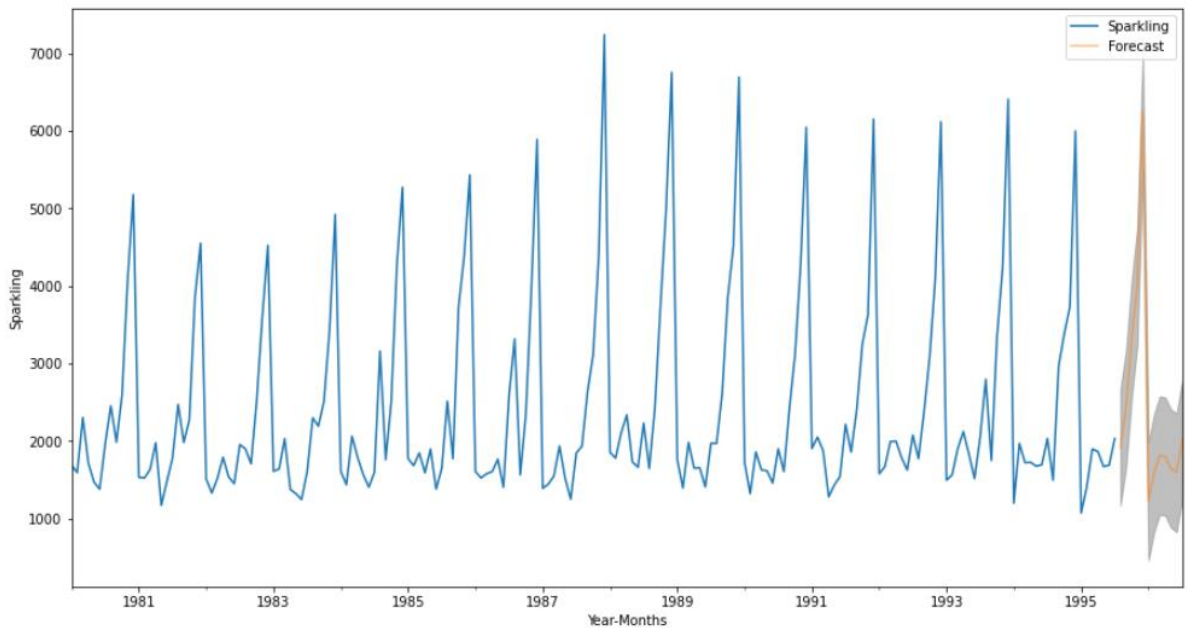


SPARLKLING WINE (PREDICTION) :

Orange line shows prediction, gray line shows the confidence interval of 95%.

The SARIMA model shown below is utilized for prediction owing to its low RMSE score of ~19 units and owing to the ability of the model to capture both seasonal and non-seasonal effects.

SARIMA(0, 1, 2)(2, 0, 2, 12)	527.571342	18.85



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

Based on the forecasting graphs shown in the previous pages for both varieties of Wine, it is clear that

a) The sales of Rose wine has been decreasing over the years and is predicted to decrease over the next 12 months.

b) The sales of Sparkling wine has been steady with promise of increasing and appears to follow a similar trend over the next twelve months.

c) The sales of Wine appears to be highest during the holiday season.

Keeping these observations in mind, its recommended that ABC company look into the reasons for decrease in the popularity of their Rose wine, and undertake any sales or marketing campaigns to improve its sales. Its possible that other drinks have been introduced into the market and are eating into total market that Rose wine is a part of.

With Sparkling wine on the other hand, its observed that the sales is steady or even improving over the years. Its therefore necessary to keep the good work and start campaigns that build the popularity of sparkling wine exponentially and improve its sales.