PROJECT ON TIME SERIES FORECASTING

BUSINESS REPORT

Sankesh Nagrare
sankeshnagrare92@gmail.com
PGP – Data Science & Business Analytics

TABLE OF CONTENTS

Sr. No.	Topic	Page No.
1	Case Study 1 – Time Series Forecasting on ShoeSales	5
	1.1 Read the data as an appropriate Time Series data and plot the data.	5
	1.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.	6
	1.3 Split the data into training and test. The test data should start in 1991.	9
	1.4 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.	11
	1.5 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.	19
	1.6 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.	21
	1.7 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.	23
	1.8 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.	25
	1.9 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.	26
	1.10 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.	27
2	Case Study 2 – Time Series Forecasting on SoftDrink Production	28
	2.1 Read the data as an appropriate Time Series data and plot the data.	28
	2.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.	29
	2.3 Split the data into training and test. The test data should start in 1991.	33
	2.4 Build various exponential smoothing models on the training data and evaluate the model using RMSE on the test data.	35

	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. 2.5 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. 2.6 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. 2.7 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. 2.8 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data. 2.9 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. 2.10 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.	43 44 48 51 52 53

LIST OF TABLES

Sr.No.	Table Name	Page No.
1	Original sample of the dataset	5
2	Summary of the dataset	6
3	Models with their corresponding RMSE values	25
4	Original sample of the dataset	28
5	Summary of the dataset	29
6	Models with their corresponding RMSE values	51

LIST OF FIGURES

Sr.No.	Figure Name	Page No.
1	Time Series Plot of Shoe sales	5
2	Yearly boxplot for the shoe sales.	6
3	Monthly boxplot for the Shoesales taking all the years intoaccount	6

4	Monthly plot of the given Time Series.	7
5	Time Series according to different months for different years	7
6	Additive Decomposition	7
7	Multiplicative Decomposition	8
8	Time Series plot after train and test split	10
9	Comparing models built with test data	11
10	Comparing models built with test data	12
11	Comparing models built with test data	13
12	Comparing models built with test data	14
13	Comparing Linear Regression model built with test data	16
14	Comparing Naïve model built with test data	17
15	Comparing Simple Average model built with test data	18
16	Stationary Series	19
17	ACF & PACF Plot	23
18	Observed & Forecasted Sales plot	26
19	Time Series Plot of Soft drink Production	28
20	Yearly boxplot for the Soft drink Production	29
21	Monthly boxplot for the Soft Drinks production taking all the years intoaccount	29
22	Monthly plot of the given Time Series.	30
23	Time Series according to different months for different years	30
24	Additive Decomposition	31
25	Multiplicative Decomposition	32
26	Time Series plot after train and test split	33
27	Comparing models built with test data	35
28	Comparing models built with test data	36
29	Comparing models built with test data	37
30	Comparing models built with test data	38
31	Comparing Linear Regression model built with test data	40
32	Comparing Naïve model built with test data	41
33	Comparing Simple Average model built with test data	42
34	Stationary Series	44
35	ACF & PACF Plot	48
36	Observed & Forecasted Sales plot	52

Case Study 1 – Time Series Forecasting on SHOE SALES

Overview:

You are an analyst in the IJK shoe company and you are expected to forecast the sales of the pairs of shoes for the upcoming 12 months from where the data ends. The data for the pair of shoe sales have been given to you from January 1980 to July 1995.

Summary:

This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q1.1) Read the data as an appropriate Time Series data and plot the data.

YearMonth	Shoe_Sales
1980-01-01	85
1980-02-01	89
1980-03-01	109
1980-04-01	95
1980-05-01	91

Table 1: Original Sample of the dataset

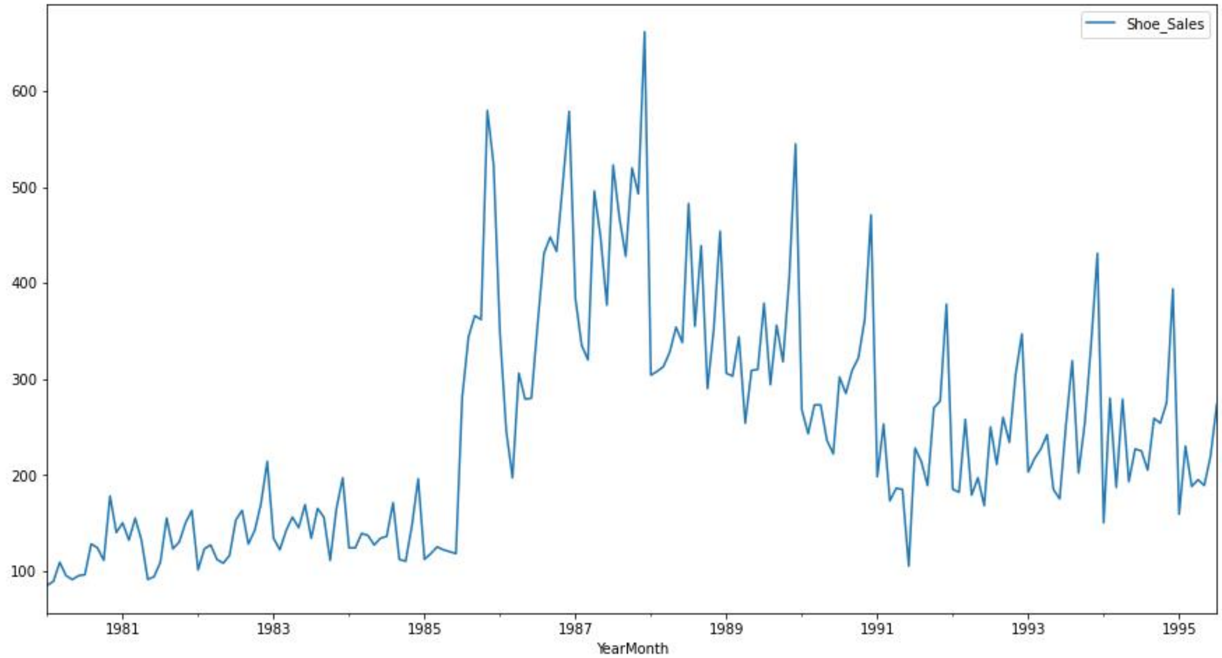


Figure 1: Time Series Plot of Shoe sales

Q1.2) Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

	Shoe_Sales
count	187.000000
mean	245.636364
std	121.390804
min	85.000000
25%	143.500000
50%	220.000000
75%	315.500000
max	662.000000

Table 2: Summary of the dataset

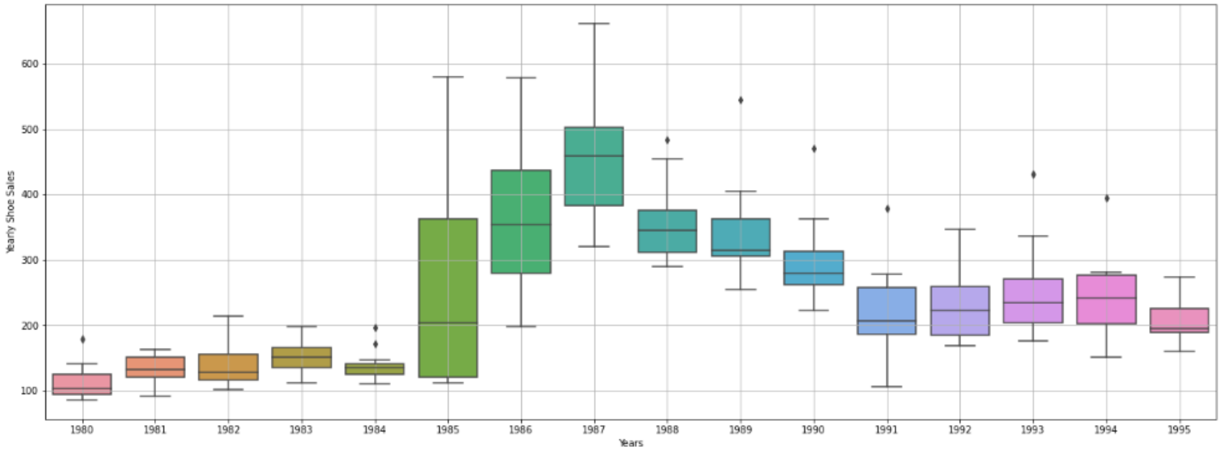


Figure 2: Yearly boxplot for the shoe sales

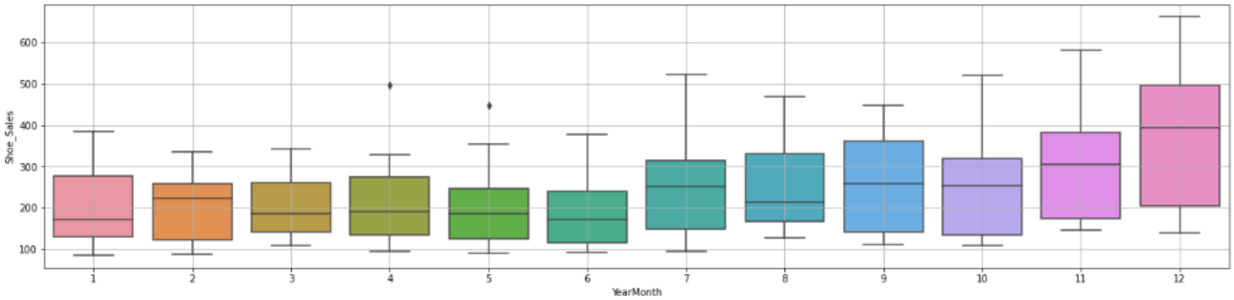


Figure 3: Monthly boxplot for the Shoe sales taking all the years into account

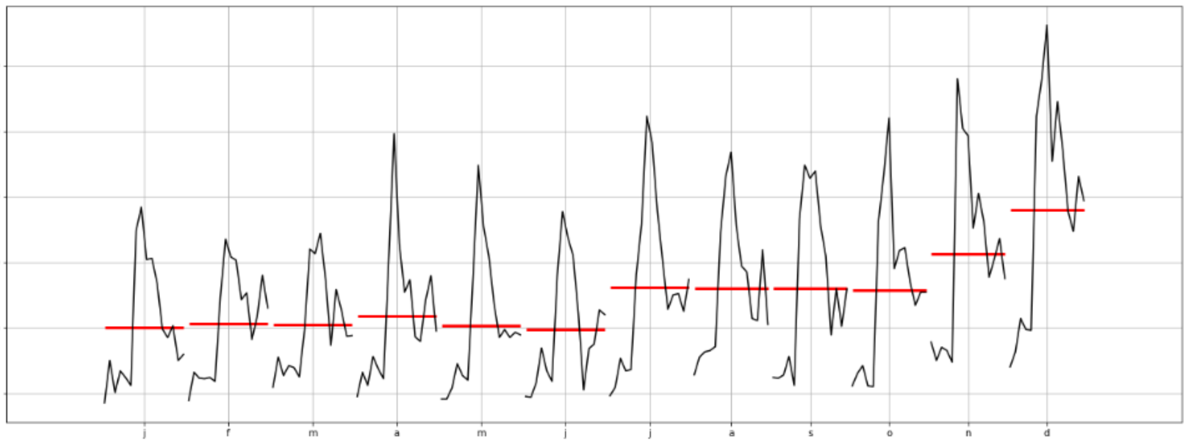


Figure 4: Monthly plot of the given Time Series.

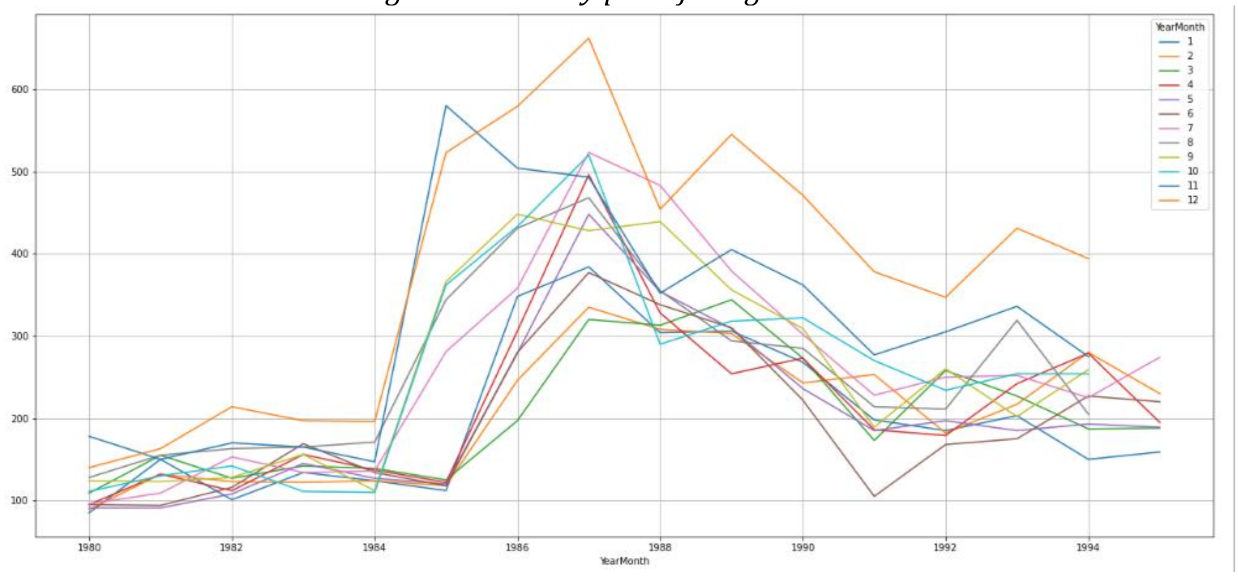


Figure 5: Time Series according to different months for different years

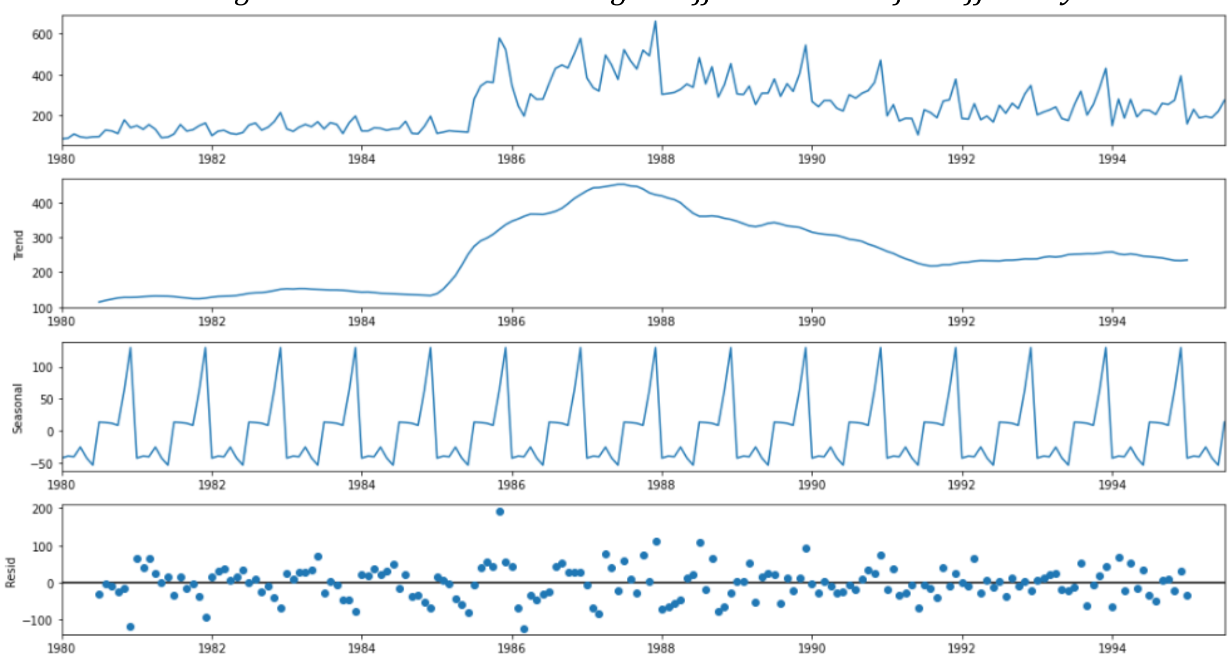


Figure 6: Additive Decomposition

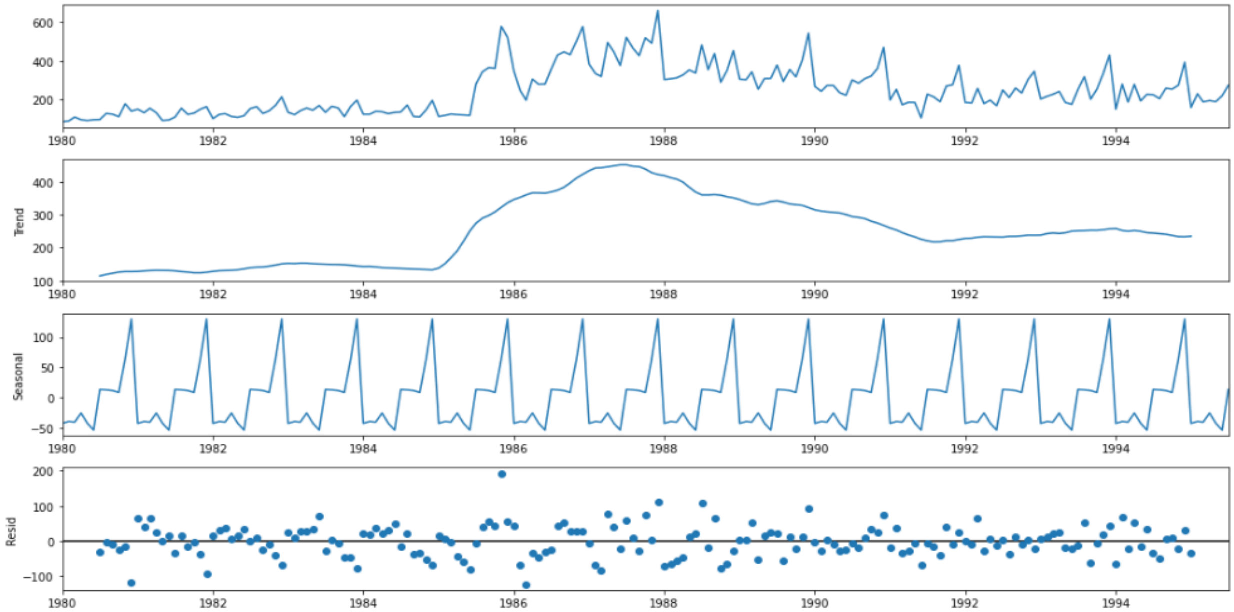


Figure 7: Multiplicative Decomposition

Observations –

- The dataset contains 187 rows and 1 column.
- The variable 'Shoe Sales' is of int datatype.
- The dataset does not contain any null values & dirty or false values in the dataset.
- From the plots above we can see the sales distributions over the years and months.
- We can say that the sales are more in the months towards the end of the year or the last quarter of the year.
- The variation of sales each year for every month is not very large.
- But we can say that year 1985 year shoe sales are increased till year 1987. Which is due to some reasons or due to some tread which occurred.

Q1.3) Split the data into training and test. The test data should start in 1991.

```
Int64Index([1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
            1991, 1992, 1993, 1994, 1995],
           dtype='int64', name='YearMonth')
```

Above are the unique year values. So now, we split the data such that our test data begins from 1991. Below is the shape and sample of the data split along with the time series plot.

```
Shape of the Training Data:  (132, 1)
Shape of the Testing Data:  (55, 1)
```

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

YearMonth	Shoe_Sales
1980-01-01	85
1980-02-01	89
1980-03-01	109
1980-04-01	95
1980-05-01	91

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

YearMonth	Shoe_Sales
1990-08-01	285
1990-09-01	309
1990-10-01	322
1990-11-01	362
1990-12-01	471

YearMonth	Shoe_Sales
1991-01-01	198
1991-02-01	253
1991-03-01	173
1991-04-01	186
1991-05-01	185

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

YearMonth	Shoe_Sales
1995-03-01	188
1995-04-01	195
1995-05-01	189
1995-06-01	220
1995-07-01	274

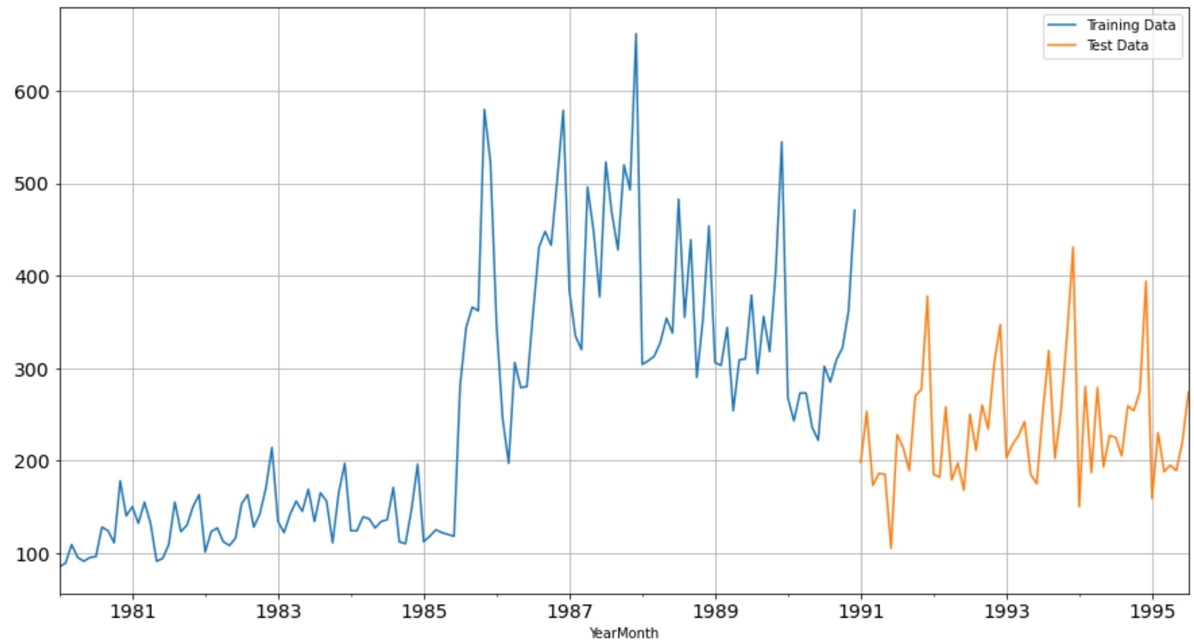


Figure 8: Time Series plot after train and test split

Q1.4) 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.

Model 1: Simple Exponential Smoothing Model

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.605049221658923,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 88.83028430097019,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

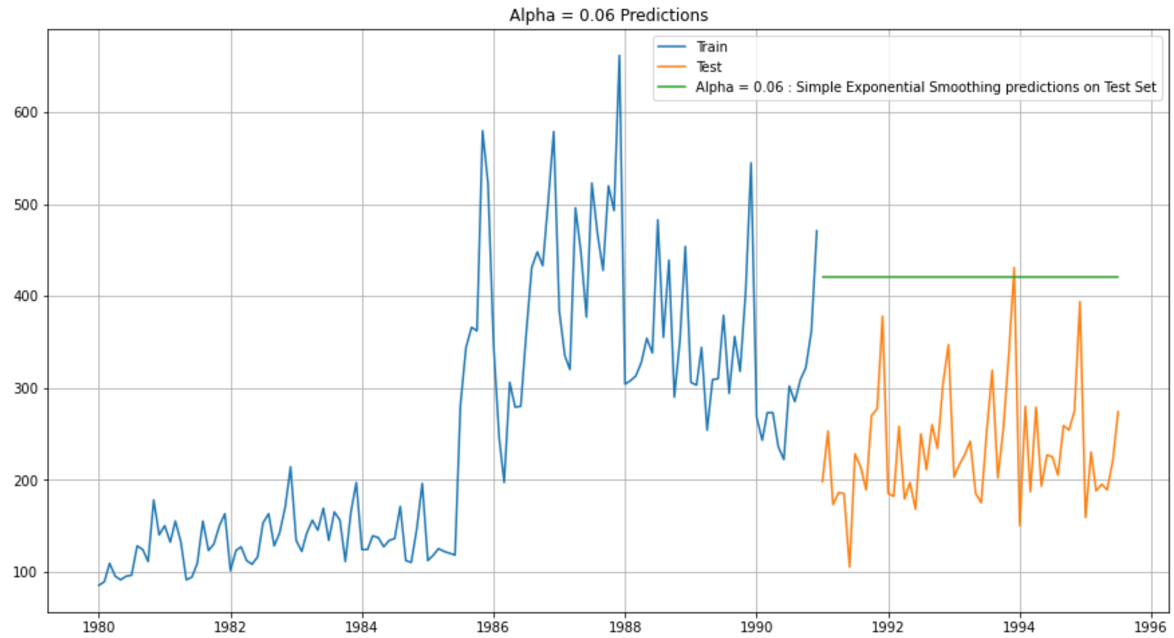


Figure 9: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
SES RMSE: 196.404836419672
```

Model 2: Double Exponential Smoothing Model

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.5948061323729839, 'smoothing_trend': 0.000279646480657923, 'smoothing_seasonal': nan, 'damping_trend': na
n, 'initial_level': 82.93815017865691, 'initial_trend': 2.5254544148321547, 'initial_seasons': array([], dtype=float64), 'use_b
oxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.
4) And then, we visualize and compare the predicted and test data on plot

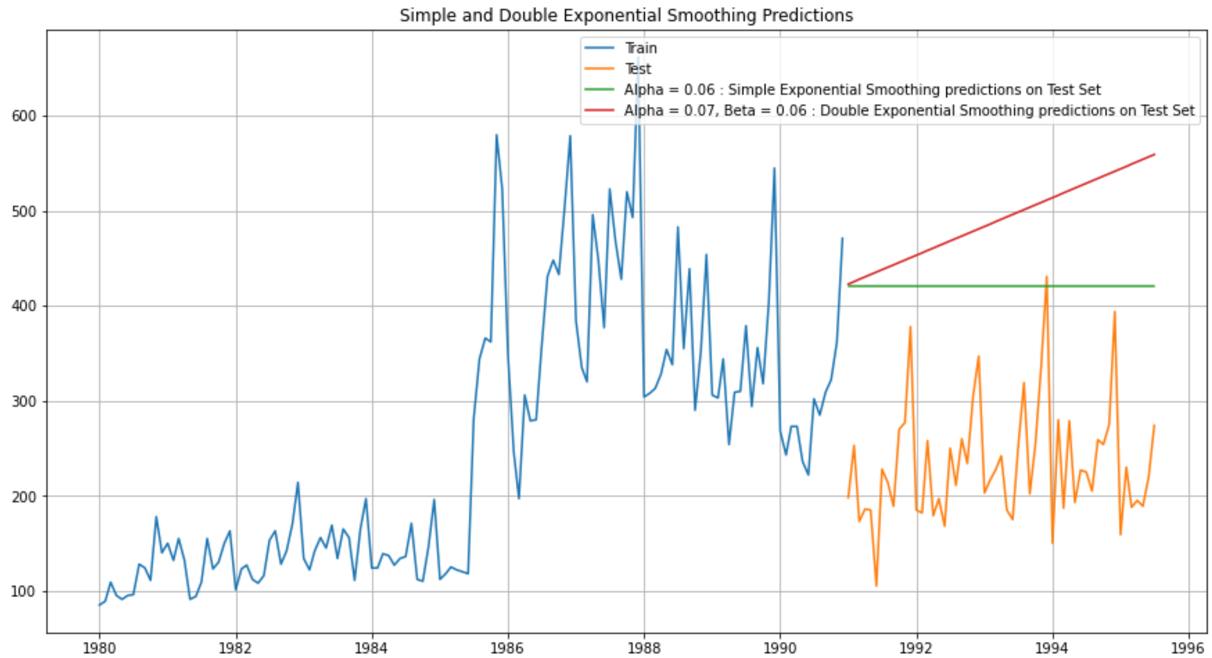


Figure 10: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
DES RMSE: 266.16120808183047
```

Model 3: Triple Exponential Smoothing (additive seasonality)

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.5707142857142857, 'smoothing_trend': 0.0001, 'smoothing_seasonal': 0.29372180451127816, 'damping_trend': nan, 'initial_level': 116.47499999999994, 'initial_trend': 1.6939393939394016, 'initial_seasons': array([-11.20138889, -14.06597222,   1.11111111,  -5.25347222,
       -21.42013889, -11.18055556, -10.83680556,  18.14236111,
        -2.53472222, -12.53472222,  28.90277778,  40.87152778]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

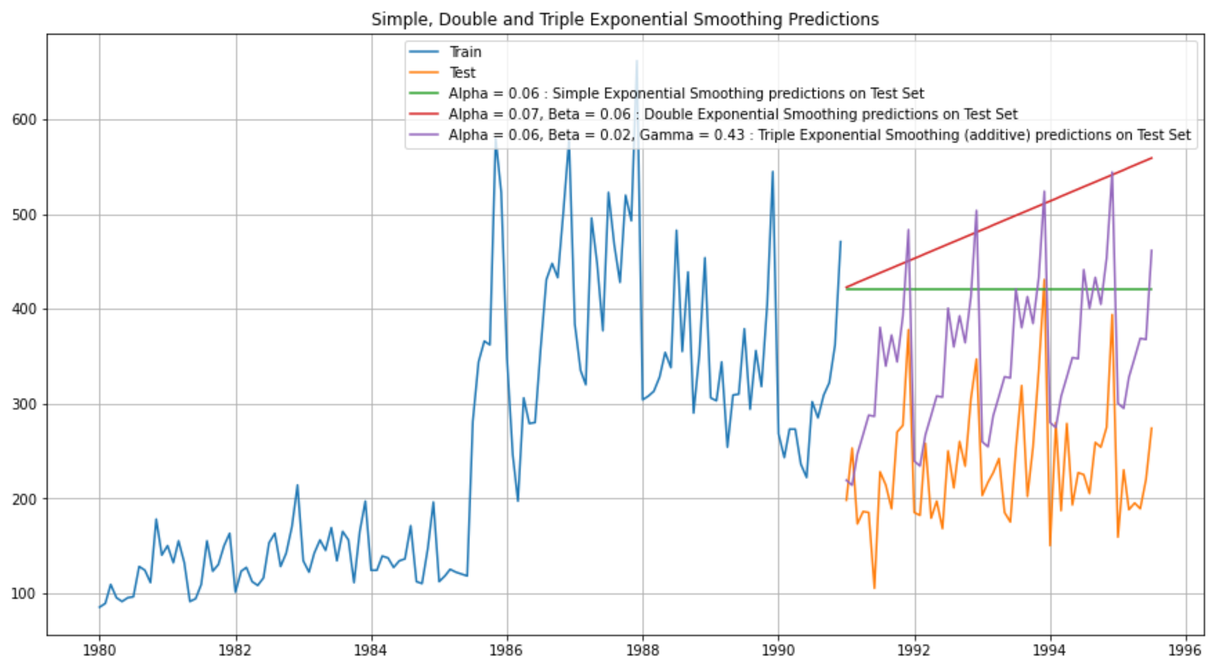


Figure 11: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
TES_add RMSE: 128.99252592312354
```

Model 4: Triple Exponential Smoothing (multiplicative seasonality)

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.5711286329525818, 'smoothing_trend': 0.00014781930867568429, 'smoothing_seasonal': 0.20294733706077994,
'damping_trend': nan, 'initial_level': 116.35529208070726, 'initial_trend': 0.11219854465675648, 'initial_seasons': array([1.05
679343, 1.01130311, 1.2337466 , 1.40663129, 1.32162715,
       1.07936886, 1.18018187, 1.50183082, 1.72369093, 1.4704132 ,
       1.75485304, 1.92101444]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

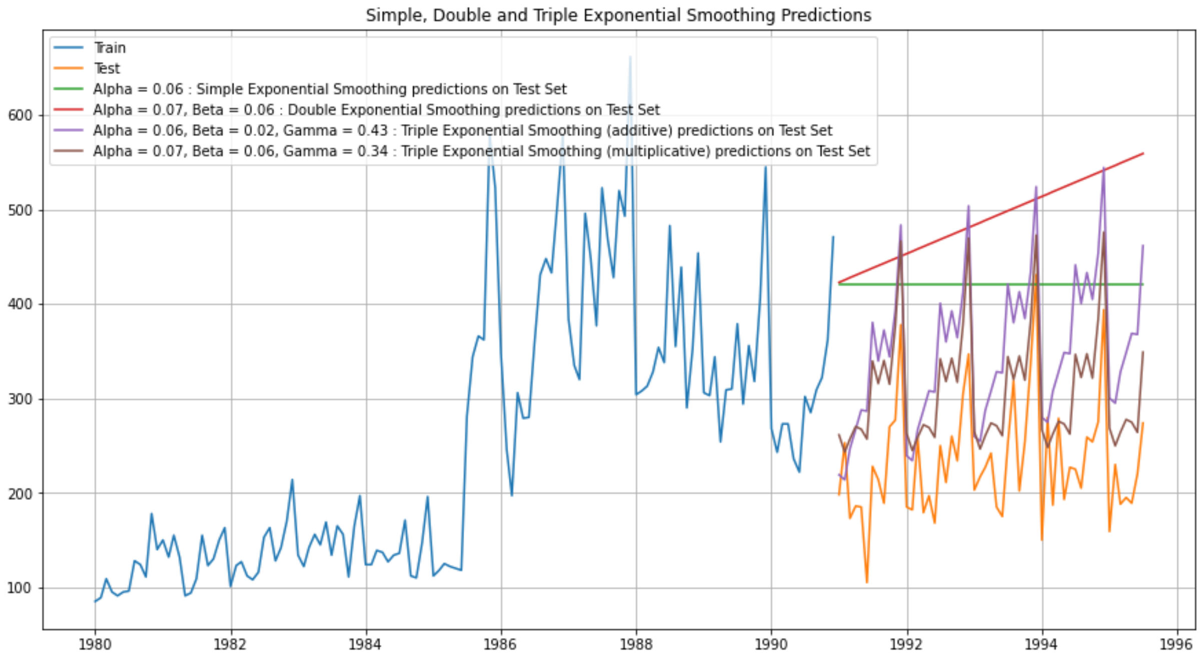


Figure 12: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
TES_mul RMSE: 83.734048494837
```

Model 5: Linear Regression model

1) For this particular linear regression, we are going to regress the 'Shoesales' variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 3
4, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,
124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157,
158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187]
```

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

First few rows of Training Data

YearMonth	Shoe_Sales	time
1980-01-01	85	1
1980-02-01	89	2
1980-03-01	109	3
1980-04-01	95	4
1980-05-01	91	5

Last few rows of Training Data

YearMonth	Shoe_Sales	time
1990-08-01	285	128
1990-09-01	309	129
1990-10-01	322	130
1990-11-01	362	131
1990-12-01	471	132

First few rows of Test Data

First few rows of Test Data

YearMonth	Shoe_Sales	time
1991-01-01	198	133
1991-02-01	253	134
1991-03-01	173	135
1991-04-01	186	136
1991-05-01	185	137

Last few rows of Test Data

YearMonth	Shoe_Sales	time
1995-03-01	188	183
1995-04-01	195	184
1995-05-01	189	185
1995-06-01	220	186
1995-07-01	274	187

2) We then, build, initialize and fit the model on the training data with default parameters.
3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

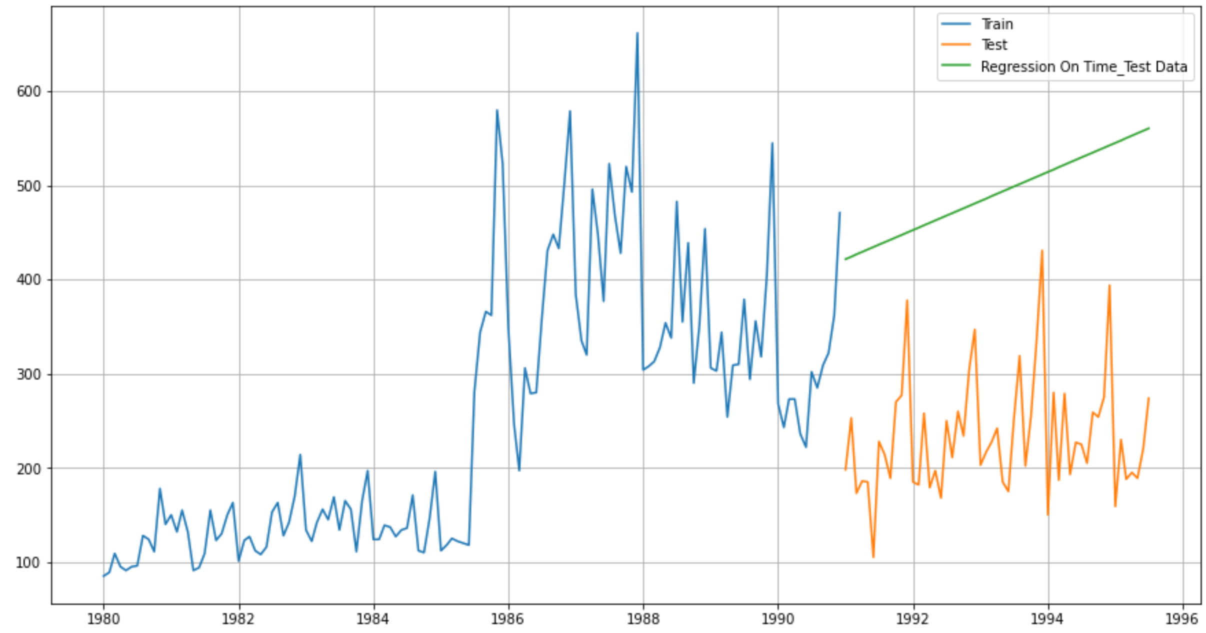


Figure 13: Comparing Linear Regression model built with test data

5) Afterwards, we calculate RMSE value for the model

```
For Regression Model on the Test Data,  RMSE is 266.276
```

Model 6: Naïve Model

1) We build, initialize and fit the model on the training data with default parameters.
2) Once done, we then predict for test dataset.
3) And then, we visualize and compare the predicted and test data on plot

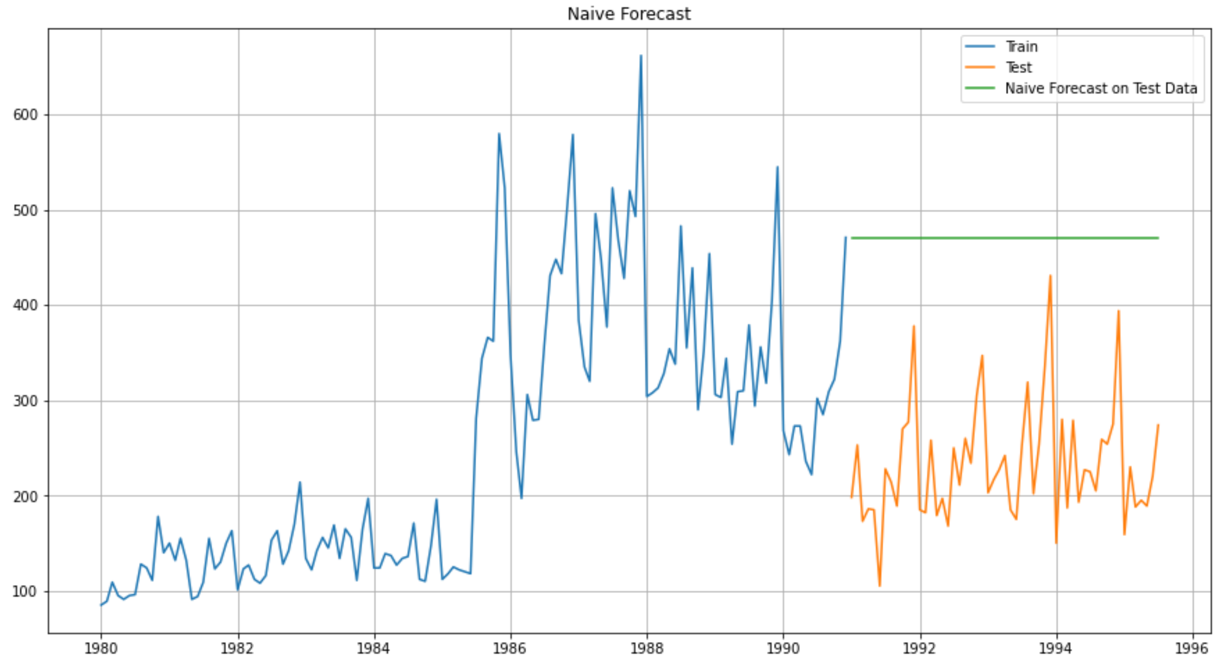


Figure 14: Comparing Naïve model built with test data

4) Afterwards, we calculate RMSE value for the model

```
For Naive Model on the Test Data,  RMSE is 245.121
```

Model 7: Simple Average Model

1) We build, initialize and fit the model on the training data with default parameters.
2) Once done, we then predict for test dataset.
3) And then, we visualize and compare the predicted and test data on plot

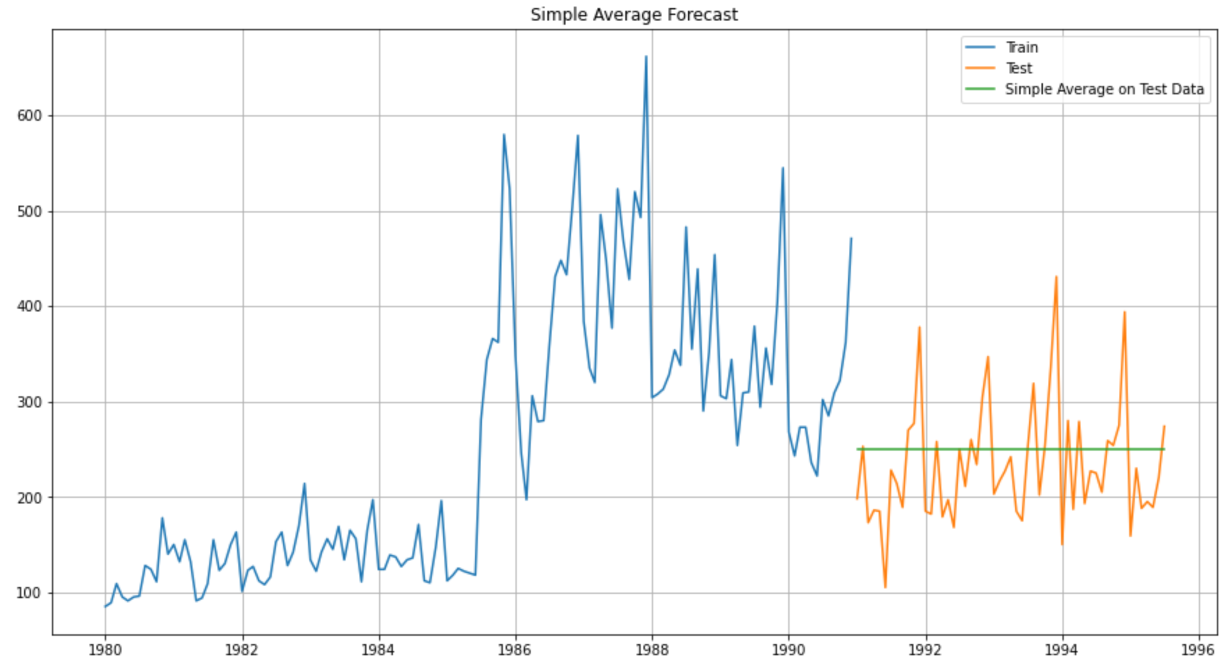


Figure 15: Comparing Simple Average model built with test data

4) Afterwards, we calculate RMSE value for the model

```
For Simple Average forecast on the Test Data,  RMSE is 63.985
```

Q1.5) 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 is a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

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

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

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

```
DF test statistic is -1.749
DF test p-value is 0.7287654522797273
Number of lags used 13
```

We see that at 5% significant level the Time Series is non-stationary.
Let us take one level of differencing to see whether the series becomes stationary.

```
DF test statistic is -3.181
DF test p-value is 0.0882258925591975
Number of lags used 13
```

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

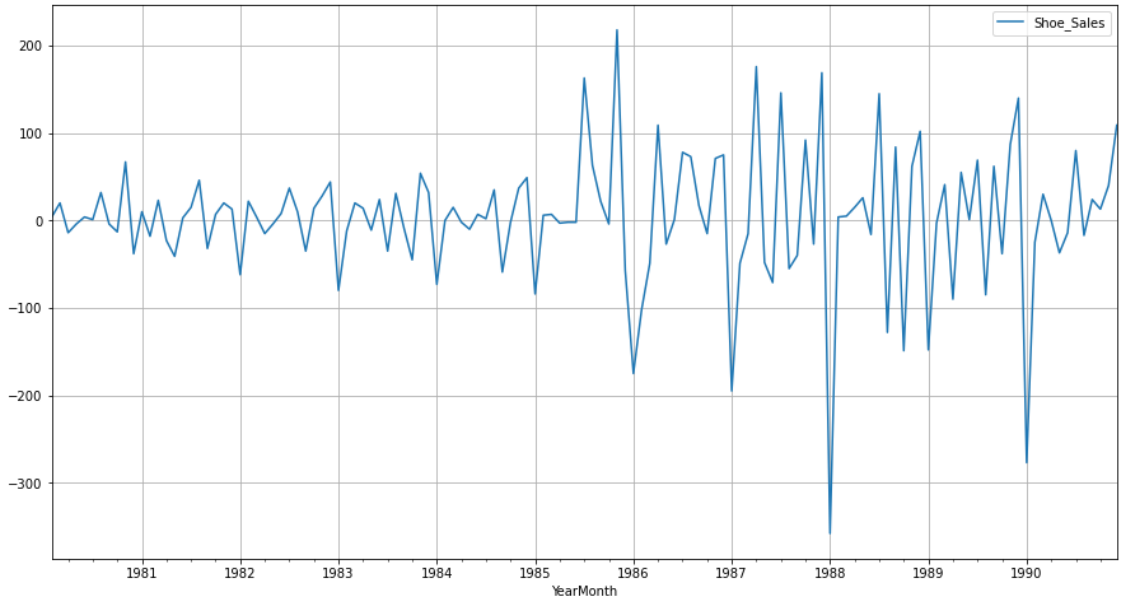
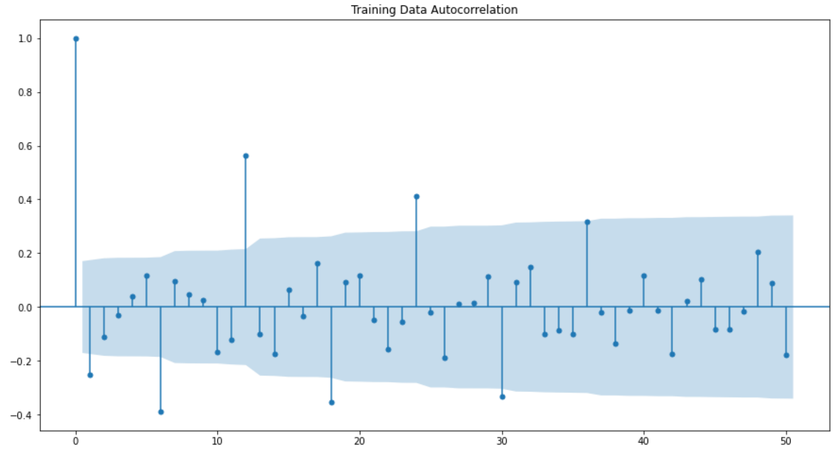


Figure 16: Stationary series

Q1.6) 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.

From decomposition we know that, the data contains seasonality. Hence, we would create SARIMA models as it accounts for seasonality
From the ACF plot we know there is a significant term after every 12 lags



1) We create some combinations for the SARIMA model

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (0, 1, 3)(0, 0, 3, 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: (1, 1, 3)(1, 0, 3, 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)
Model: (2, 1, 3)(2, 0, 3, 12)
Model: (3, 1, 0)(3, 0, 0, 12)
Model: (3, 1, 1)(3, 0, 1, 12)
Model: (3, 1, 2)(3, 0, 2, 12)
Model: (3, 1, 3)(3, 0, 3, 12)
```

2) We find AIC values for all the combinations and find the one with the least AIC value as it is best in terms of performance.

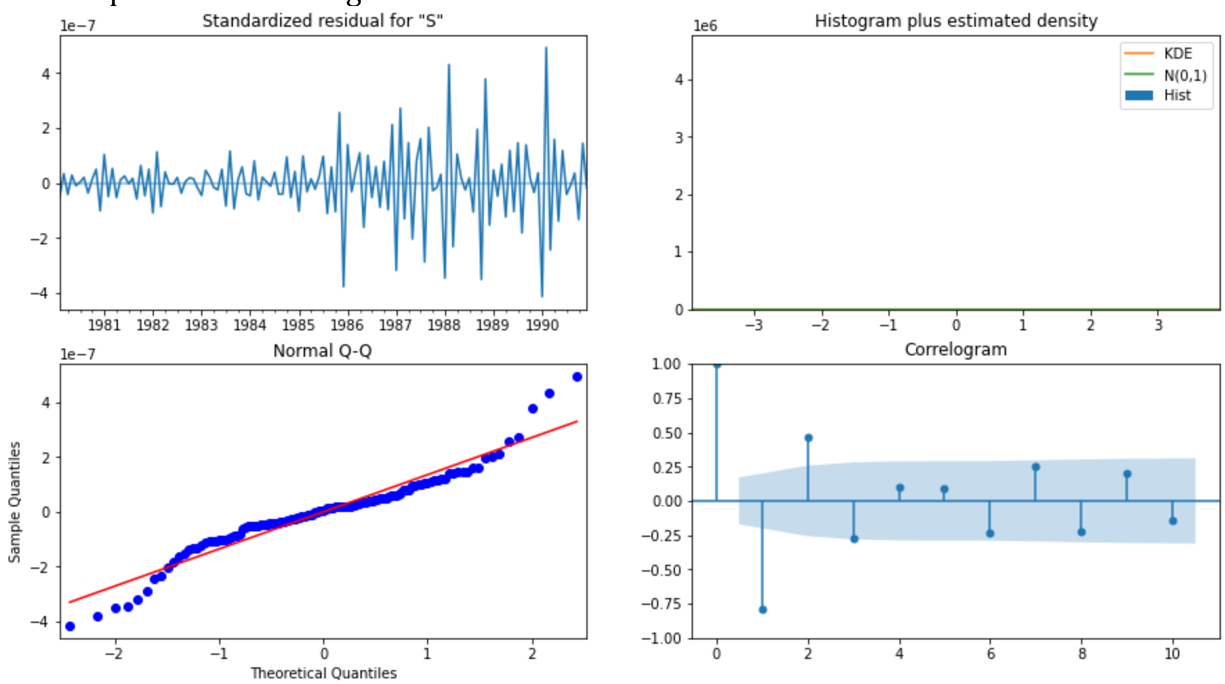
	param	seasonal	AIC
147	(2, 1, 1)	(0, 0, 3, 12)	14.0
211	(3, 1, 1)	(0, 0, 3, 12)	16.0
179	(2, 1, 3)	(0, 0, 3, 12)	18.0
223	(3, 1, 1)	(3, 0, 3, 12)	22.0
255	(3, 1, 3)	(3, 0, 3, 12)	26.0

3) Then we find details of the best combination

```
                                          SARIMAX Results
===================================================================================================
Dep. Variable:                                Shoe_Sales   No. Observations:                  132
Model:             SARIMAX(1, 1, 1)x(0, 0, [1, 2, 3], 12)   Log Likelihood               -2860.188
Date:                                   Thu, 16 Jun 2022   AIC                           5732.376
Time:                                           11:36:51   BIC                           5749.627
Sample:                                       01-01-1980   HQIC                          5739.386
                                            - 12-01-1990
Covariance Type:                                     opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9889   4.04e-21   2.45e+20      0.000       0.989       0.989
ma.L1          1.8802    1.7e-19   1.11e+19      0.000       1.880       1.880
ma.S.L12    2.245e+13   5.65e-34   3.97e+46      0.000    2.24e+13    2.24e+13
ma.S.L24   -8.622e+13   6.23e-34  -1.38e+47      0.000   -8.62e+13   -8.62e+13
ma.S.L36   -1.262e+14   2.02e-33  -6.26e+46      0.000   -1.26e+14   -1.26e+14
sigma2      2.329e-11   1.67e-10      0.139      0.889   -3.05e-10    3.51e-10
===================================================================================
Ljung-Box (L1) (Q):                  84.26   Jarque-Bera (JB):                35.26
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):              12.74   Skew:                             0.06
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.54
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.07e+63. Standard errors may be unstable.
```

4) Then we plot results of diagnostics



5) Then, we find RMSE & MAPE values

```
RMSE: 1512.1692810762256
MAPE: 584.9162430955059
```

Q1.7) 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.

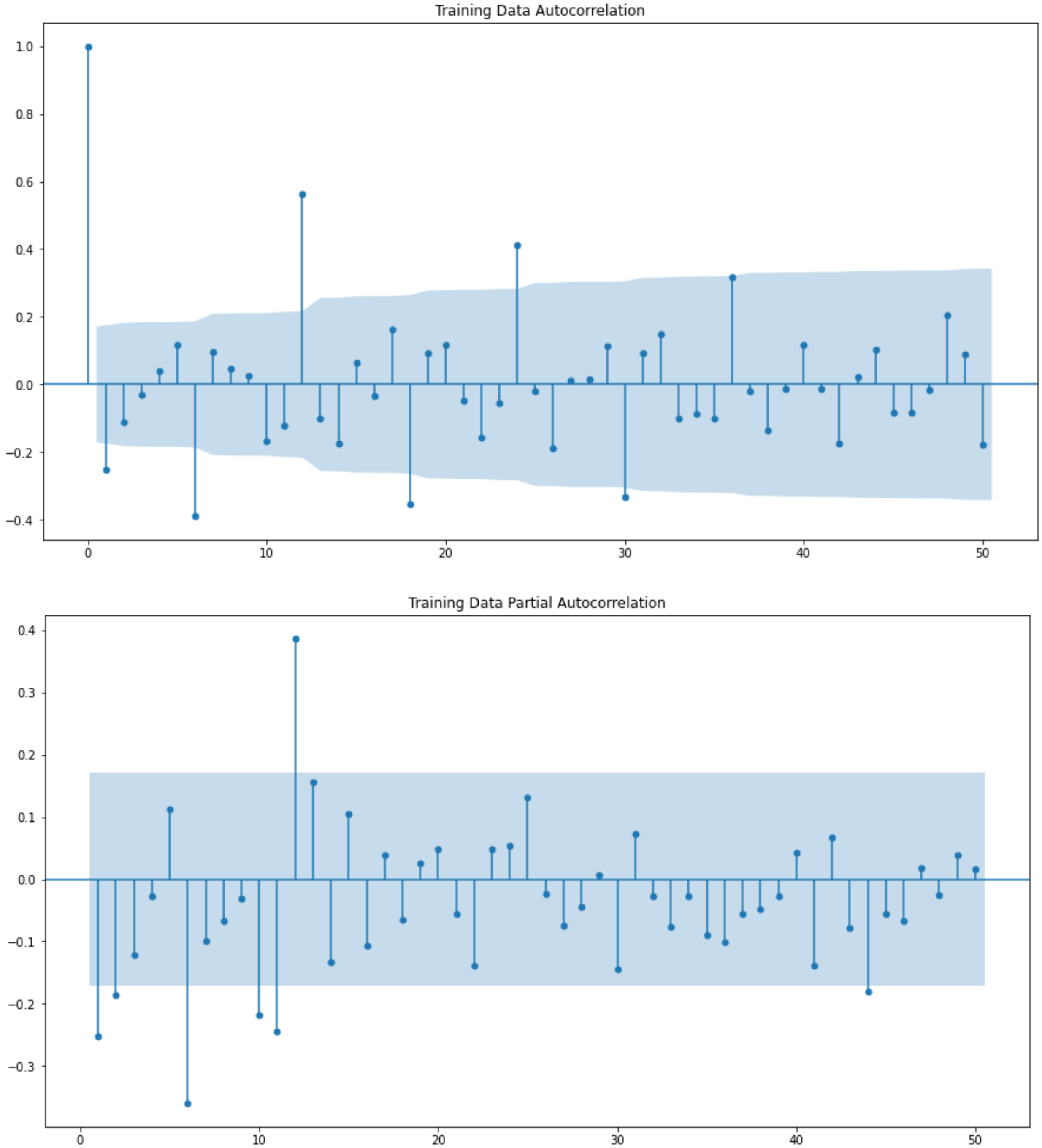


Figure 17: ACF & PACF Plot

From the ACF and PACF plots above, we can say that,

Cut-off of ACF & PACF both is 0 (i.e. 'p' & 'q')

Therefore, the order = (01,0). Since, the differencing done is of first order.

Also, seasonal cut-offs are seen after every 12 lags.

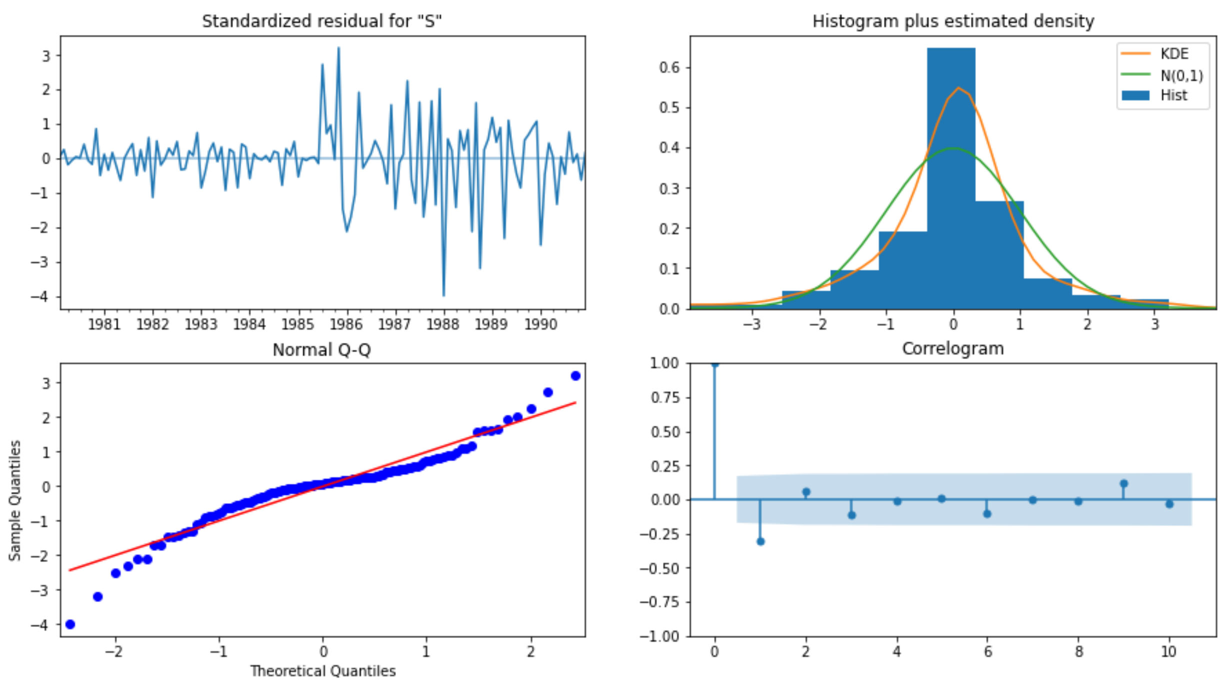
Therefore, the seasonal order = (0,0,12,12)

Results of combination decided.

```
                                        SARIMAX Results
==============================================================================================================
=
Dep. Variable:                                                   Shoe_Sales   No. Observations:            13
2
Model:           SARIMAX(0, 1, 0)x(0, 0, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], 12)   Log Likelihood    -718.74
4
Date:                                                      Thu, 16 Jun 2022   AIC                     1463.48
8
Time:                                                              11:37:39   BIC                     1500.86
6
Sample:                                                          01-01-1980   HQIC                    1478.67
7
                                                               - 12-01-1990
Covariance Type:                                                        opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.S.L12       0.5447    498.023      0.001      0.999    -975.562     976.652
ma.S.L24       0.2462    393.339      0.001      1.000    -770.684     771.176
ma.S.L36       0.3406     17.661      0.019      0.985     -34.274      34.955
ma.S.L48       0.4085    190.545      0.002      0.998    -373.053     373.870
ma.S.L60       0.6702    192.965      0.003      0.997    -377.534     378.874
ma.S.L72       0.6736    171.846      0.004      0.997    -336.138     337.485
ma.S.L84       0.2098    323.542      0.001      0.999    -633.920     634.340
ma.S.L96       0.6880     33.060      0.021      0.983     -64.107      65.483
ma.S.L108      0.0395     62.840      0.001      0.999    -123.124     123.203
ma.S.L120     -0.1149     55.855     -0.002      0.998    -109.588     109.358
ma.S.L132      0.0964     18.603      0.005      0.996     -36.365      36.558
ma.S.L144      0.0648     19.383      0.003      0.997     -37.925      38.054
sigma2      1954.1250      0.790   2473.770      0.000    1952.577    1955.673
===================================================================================
Ljung-Box (L1) (Q):                  12.56   Jarque-Bera (JB):                48.45
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               9.74   Skew:                            -0.48
Prob(H) (two-sided):                     0.00   Kurtosis:                            5.82
===========================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.29e+22. Standard errors may be unstable.
```

Plotting results of diagnostics



RMSE & MAPE value

```
RMSE: 91.53438270942351
MAPE: 31.502963661665124
```

Q1.8) Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

	Test RMSE
Alpha = 0.06 : SES	196.404836
Alpha = 0.07, Beta = 0.06 : DES	266.161208
Alpha = 0.06, Beta = 0.02, Gamma = 0.43 : TES_add	128.992526
Alpha = 0.07, Beta = 06, Gamma = 0.34 : TES_mul	83.734048
Regression Model	266.276472
Naive Model	245.121306
Simple Average Model	63.984570
SARIMA_Auto(1,1,1)(0,0,3,12)	1512.169281
SARIMA_Manual(0,1,0)(0,0,12,12)	91.534383

Table 3: Models with their corresponding RMSE values

From the table above we can conclude that,

Since, the RMSE value of Triple Exponential Smoothing model (additive seasonality) is the least, it is the best performing model. Hence, we would use that model to predict or forecast further sales.

Q1.9) 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.

By comparing the RMSE values of all the models built, we find that, Triple Exponential Smoothing model as it has the lowest RMSE value, which means best performance as we know.

So, we fit the entire data into that model for the entire data

```
{'smoothing_level': 0.4072309571087021, 'smoothing_trend': 0.03229263935904019, 'smoothing_seasonal': 0.22061599693785627, 'dam
ping_trend': nan, 'initial_level': 123.79194018265252, 'initial_trend': -0.11837936617161127, 'initial_seasons': array([-41.725
47063, -45.69189807, -25.68845247, -21.16373101,
       -30.07213165, -17.03805695,  10.49661784,  37.94128517,
        22.47110609,  12.80029182,  57.41106955,  47.57474517]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Then, we forecast for 12 months in the future and calculate RMSE value.

```
RMSE of the Full Model 52.554842406466456
```

Now, we plot the forecast to visualize it

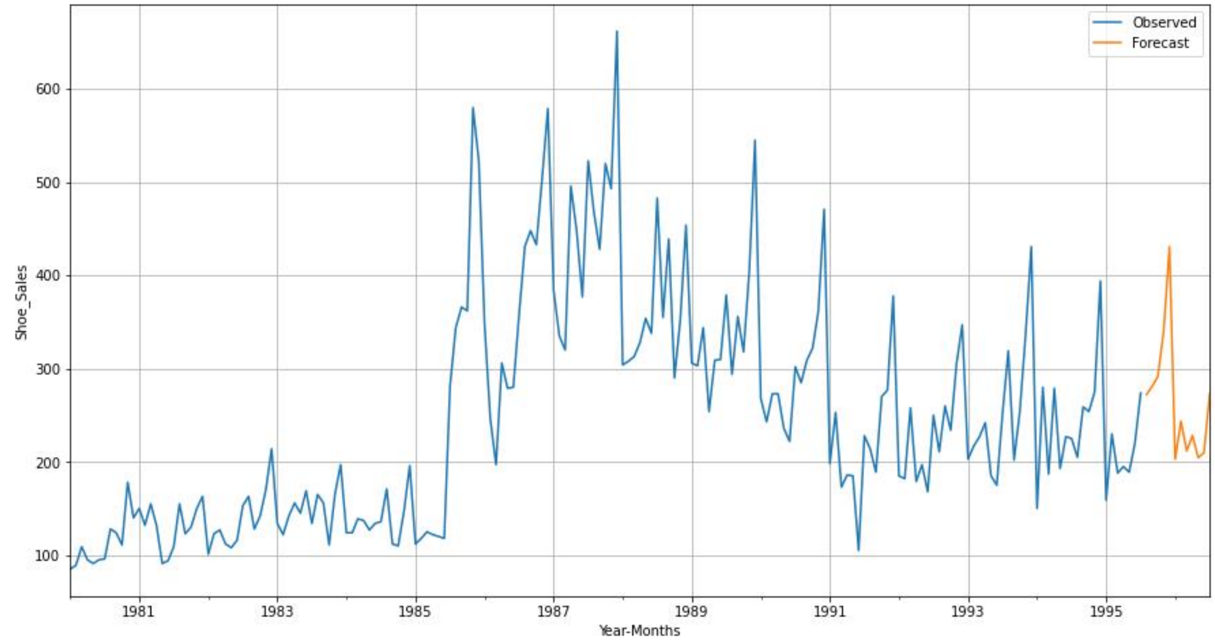


Figure 18: Observed & Forecasted Sales plot

Q1.10) Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

- From the final model used to forecast future 12 months sale suggest that, the sales will be more than the previous 12 months.
- Model performs very well and can be used to make business decisions which could further increase sales or used to devise new marketing strategies.
- The trend observed from the dataset provided suggests that, the last quarter of every year performs best in sales compared to the other quarters which have average sales. So, we need to decide which part of the year's sale we need to boost to maximize the overall profit.
- We should also understand the average market sales of Shoe & our leading competitors' sales to understand our performance in the market much better.
- Also, we should conduct a survey which helps us in understanding why the sales in the first 3 quarters is relatively low and use that survey data to improve and devise new marketing strategies to increase sales in the other 3 quarters as well.

Case Study 2 – Time Series Forecasting on SoftDrink Production.

Overview:
You are an analyst in the RST soft drink company and you are expected to forecast the sales of the production of the soft drink for the upcoming 12 months from where the data ends. The data for the production of soft drink has been given to you from January 1980 to July 1995.

Summary:
This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q2.1) Read the data as an appropriate Time Series data and plot the data.

YearMonth	SoftDrinkProduction
1980-01-01	1954
1980-02-01	2302
1980-03-01	3054
1980-04-01	2414
1980-05-01	2226

Table 4: Original Sample of the dataset

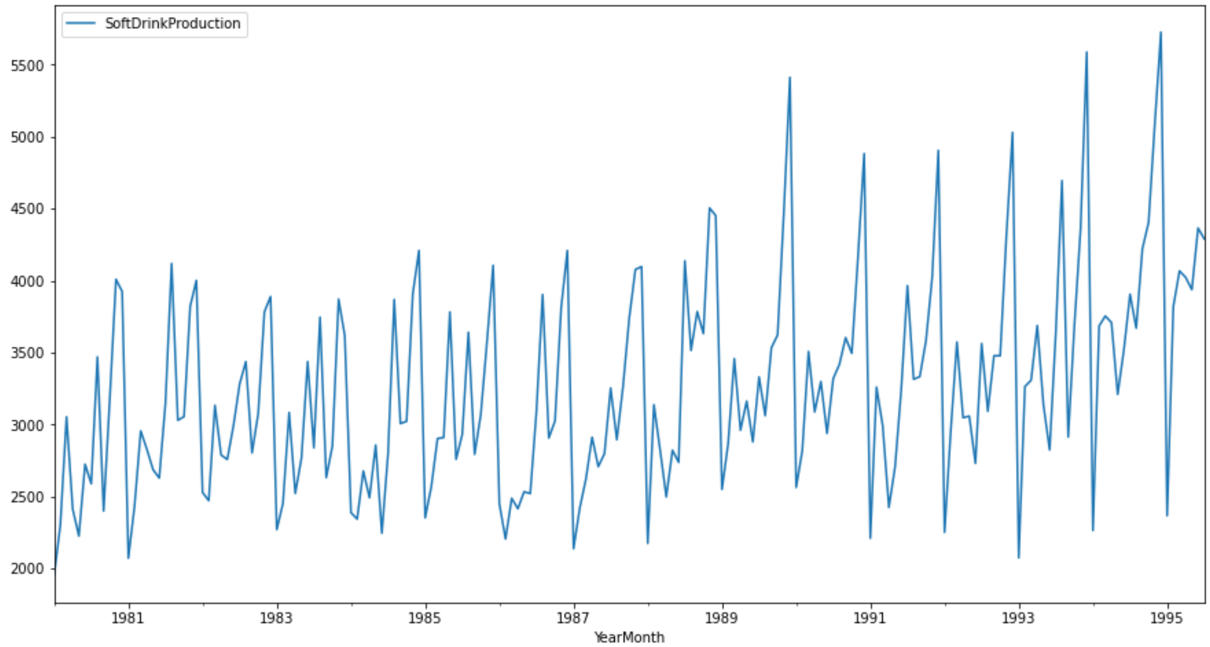


Figure 19: Time Series Plot of Soft Drink Production

Q2.2) Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

	SoftDrinkProduction
count	187.000000
mean	3262.609626
std	728.357367
min	1954.000000
25%	2748.000000
50%	3134.000000
75%	3741.000000
max	5725.000000

Table 5: Summary of the dataset

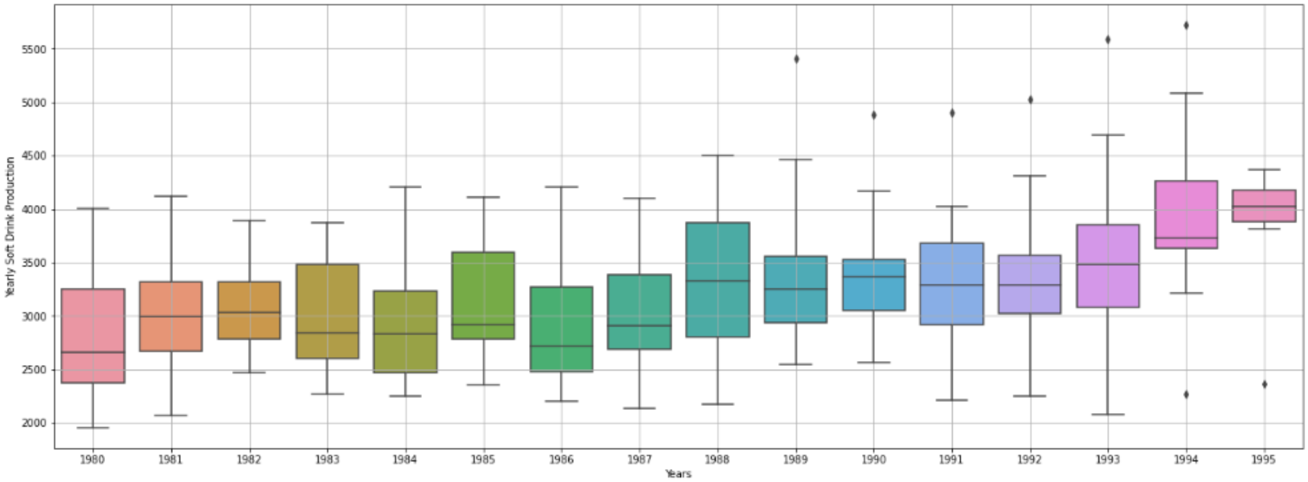


Figure 20: Yearly boxplot for the soft drink production

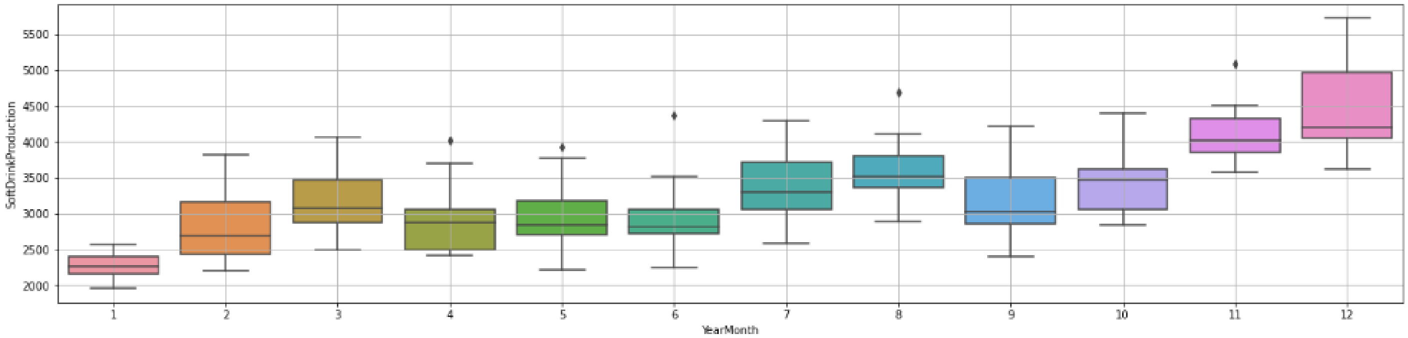


Figure 21: Monthly boxplot for the Soft Drinks production taking all the years into account

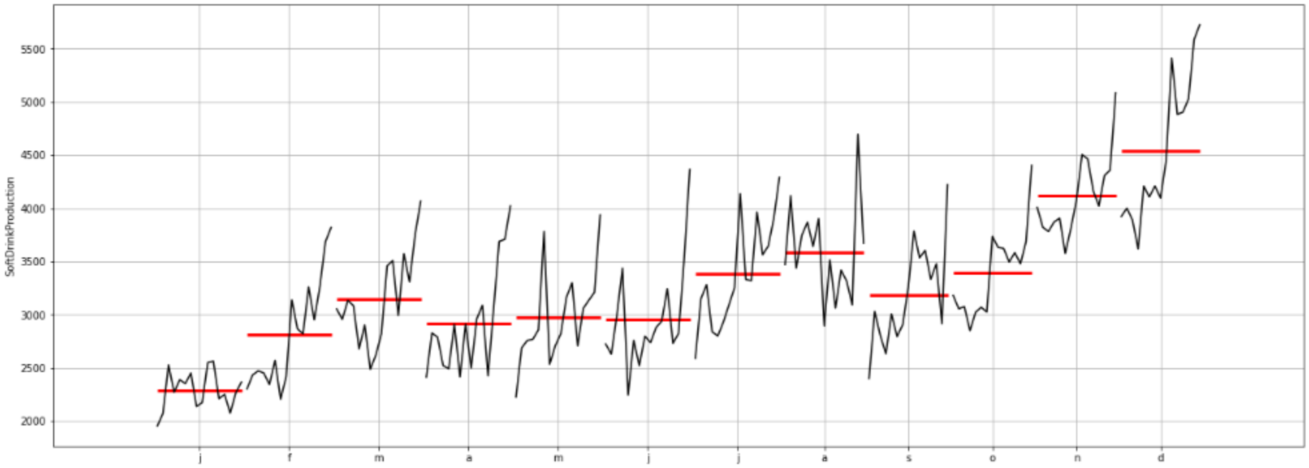


Figure 22: Monthly plot of the given Time Series.

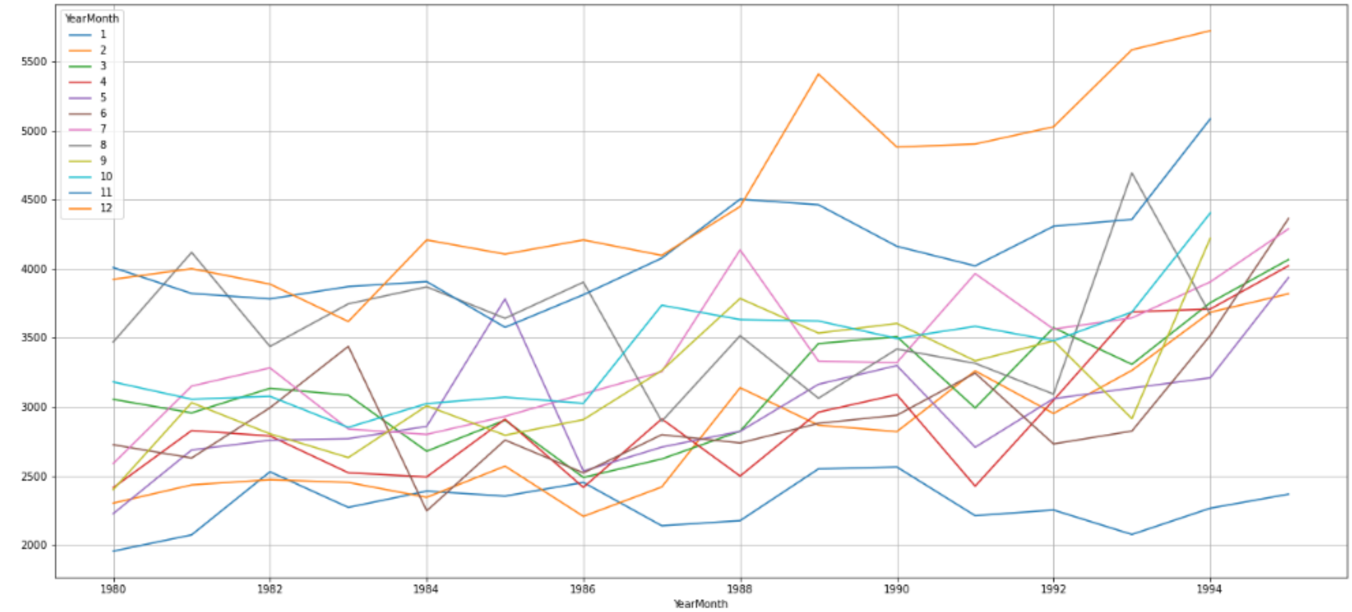


Figure 23: Time Series according to different months for different years

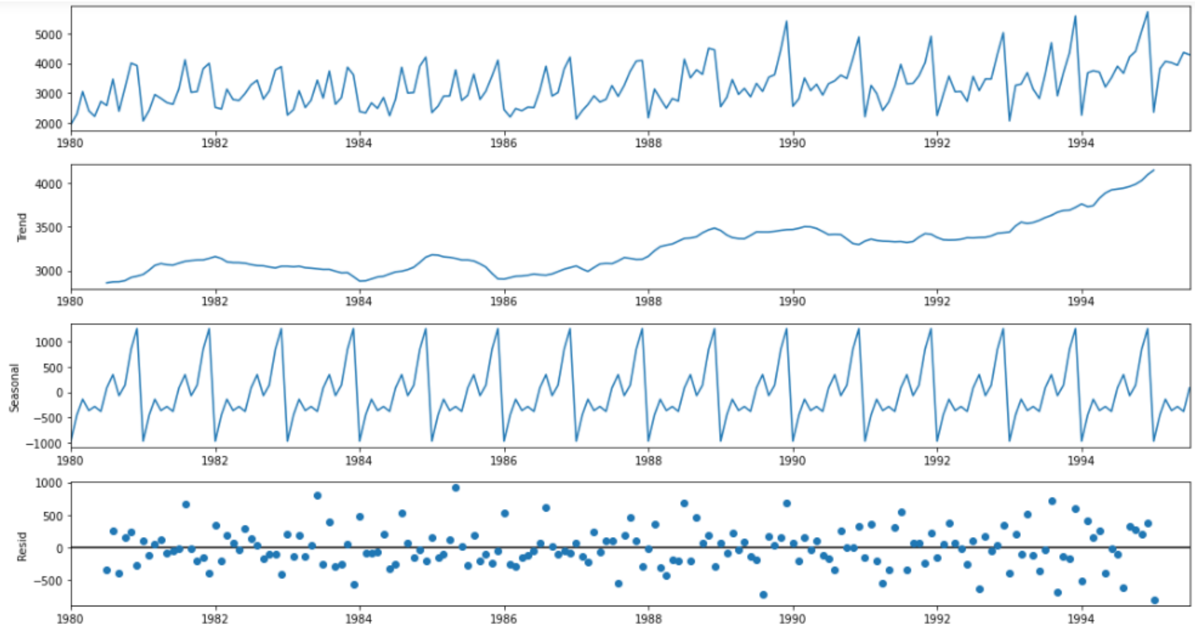


Figure 24: Additive Decomposition

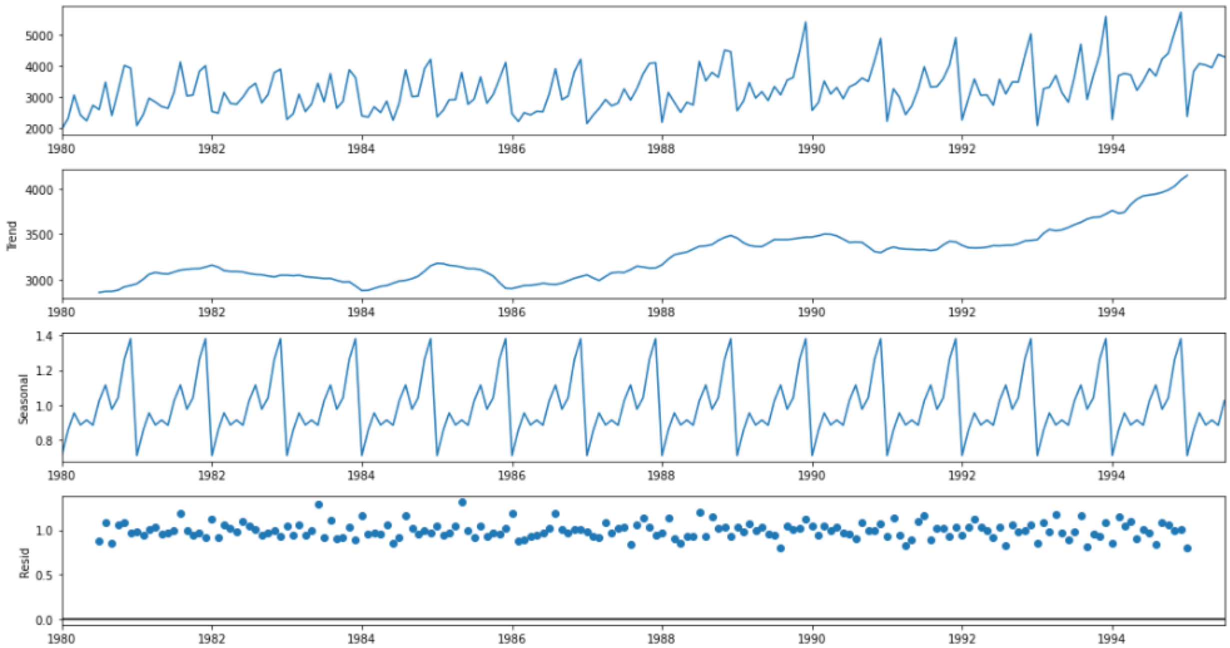


Figure 25: Multiplicative Decomposition

Observations –

- The dataset contains 187 rows and 1 column before imputation of missing values.
- The variable 'Softdrinkproduction' is of int datatype.
- The dataset does not contain any null values & dirty or false values in the dataset.
- The dataset contains 187 rows and 1 column.
- From the plots above we can see the sales distributions over the years and months.
- We can say that the production Exponentially increased every month, But more in the months towards the end of the year.
- The trend shows that production have increased every year.

Q2.3) Split the data into training and test. The test data should start in 1991.

```
Int64Index([1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
            1991, 1992, 1993, 1994, 1995],
           dtype='int64', name='YearMonth')
```

Above are the unique year values. So now, we split the data such that our test data begins from 1991. Below is the shape and sample of the data split along with the time series plot.

```
Shape of the Training Data:  (132, 1)
Shape of the Testing Data:  (55, 1)
```

```
First few rows of Training Dat
```

YearMonth	SoftDrinkProduction
1980-01-01	1954
1980-02-01	2302
1980-03-01	3054
1980-04-01	2414
1980-05-01	2226

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

YearMonth	SoftDrinkProduction
1990-08-01	3418
1990-09-01	3604
1990-10-01	3495
1990-11-01	4163
1990-12-01	4882

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

YearMonth	SoftDrinkProduction
1991-01-01	2211
1991-02-01	3260
1991-03-01	2992
1991-04-01	2425
1991-05-01	2707

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

YearMonth	SoftDrinkProduction
1995-03-01	4067
1995-04-01	4022
1995-05-01	3937
1995-06-01	4365
1995-07-01	4290

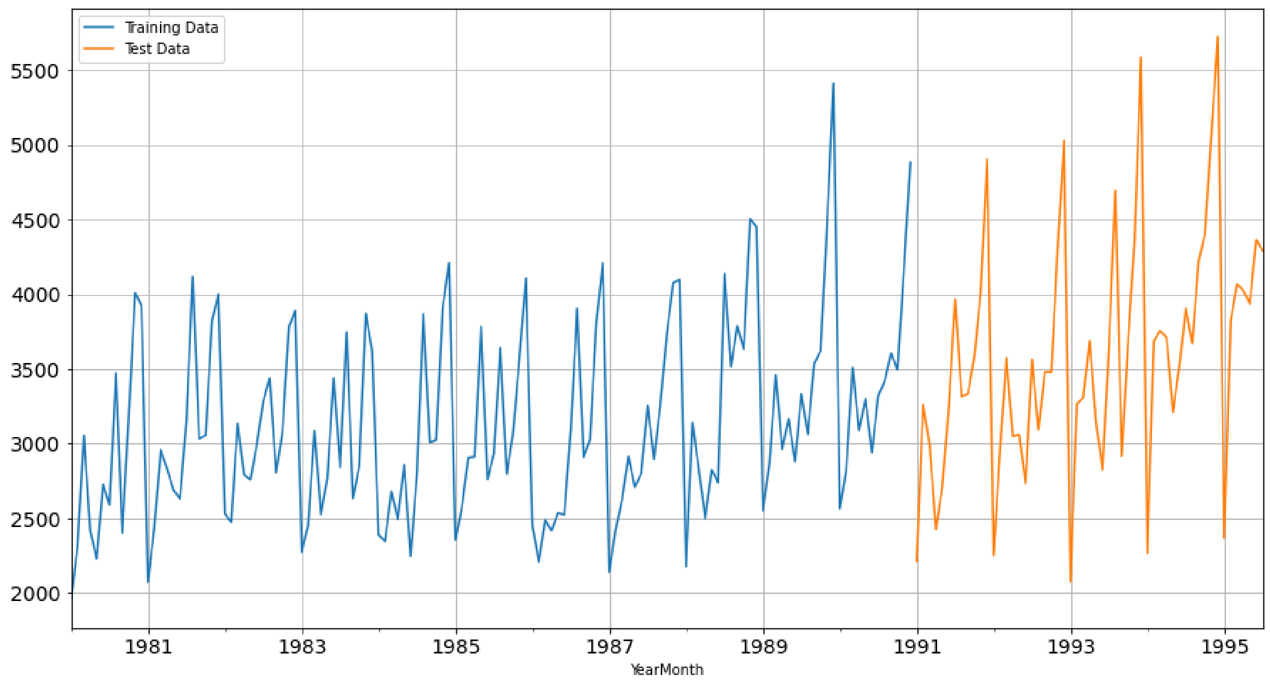


Figure 26: Time Series plot after train and test split

Q2.4) 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.

Model 1: Simple Exponential Smoothing Model

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.11907309094689855,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 2573.0166666666655,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

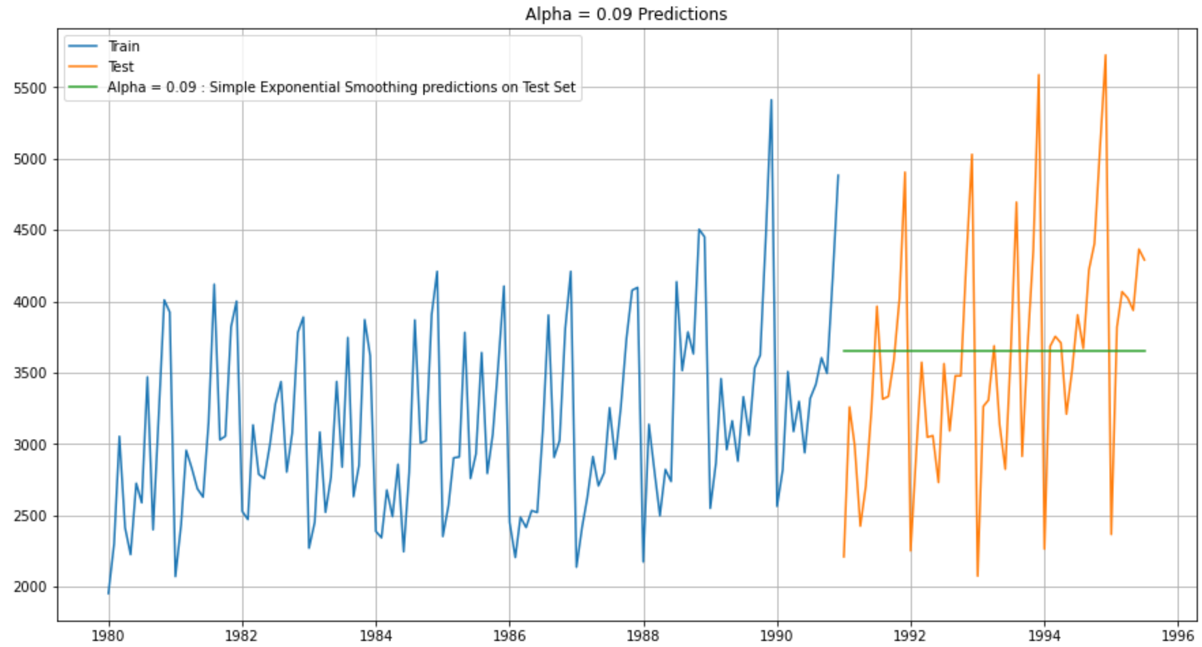


Figure 27: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
SES RMSE: 809.5016403931278
```

Model 2: Double Exponential Smoothing Model

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.1242286864966588, 'smoothing_trend': 0.10769076164072929, 'smoothing_seasonal': nan, 'damping_trend': na
n, 'initial_level': 2142.9200400852947, 'initial_trend': 42.27465415028941, 'initial_seasons': array([], dtype=float64), 'use_b
oxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.
4) And then, we visualize and compare the predicted and test data on plot

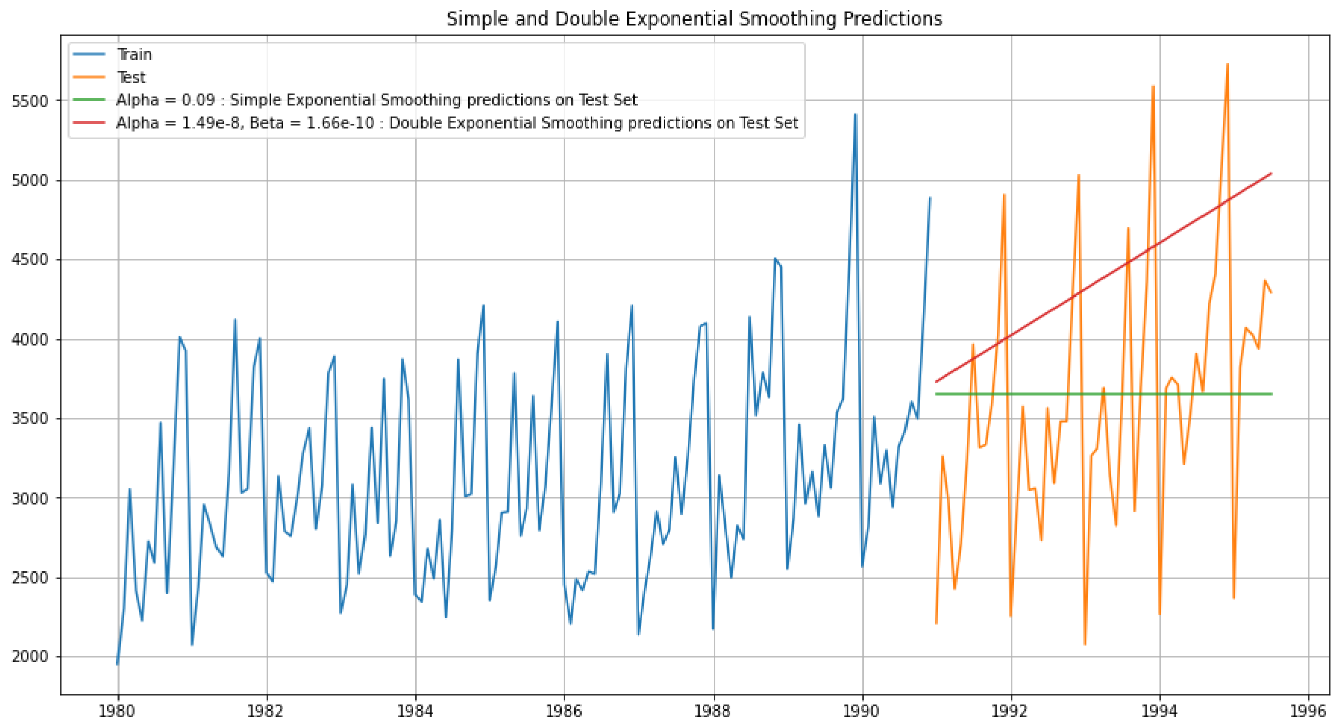


Figure 28: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
DES RMSE: 1074.3291531501832
```

Model 3: Triple Exponential Smoothing (additive seasonality)

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.14628214287204402, 'smoothing_trend': 0.03985523474431963, 'smoothing_seasonal': 0.2624197351602548, 'dam
ping_trend': nan, 'initial_level': 2803.214611111109, 'initial_trend': 7.179638888889087, 'initial_seasons': array([-687.298965
28, -582.87175694,  -55.66104861, -365.74079861,
       -253.26738194, -196.41738194,  -32.54725694,  690.31611806,
       -282.20021528,   44.75545139,  867.40386806,  853.53236806]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

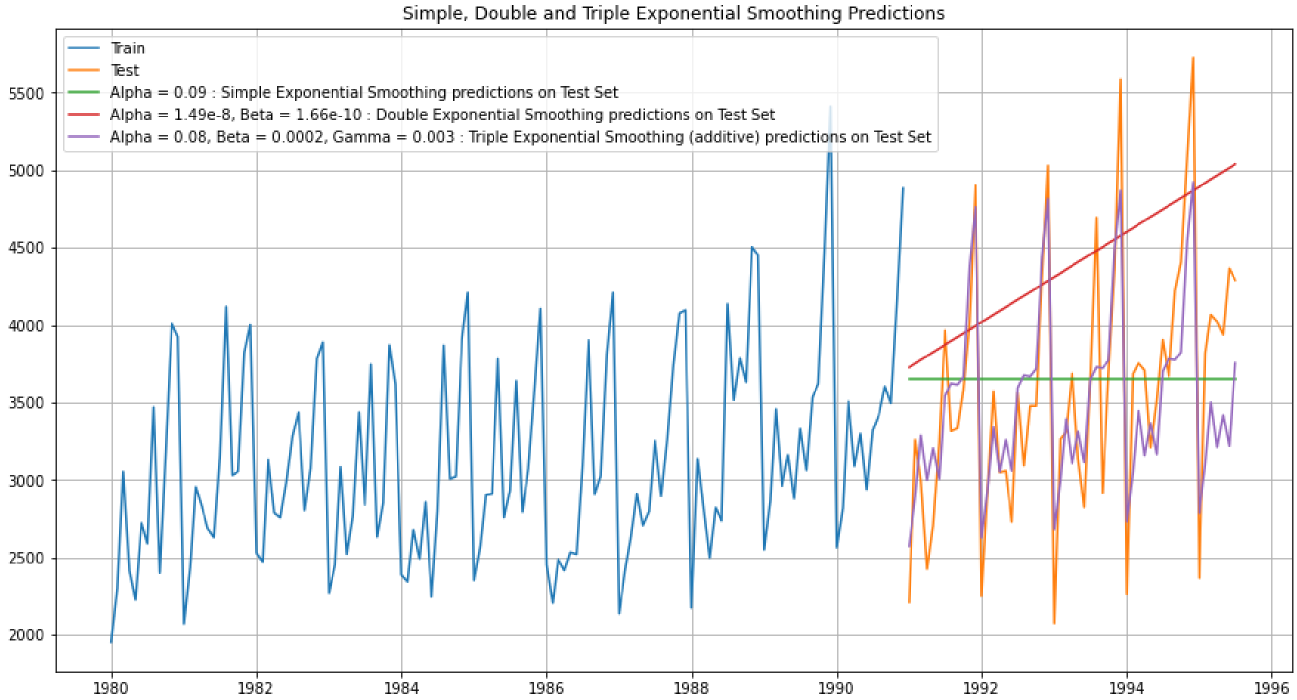


Figure 29: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
TES_add RMSE: 458.9653920540907
```

Model 4: Triple Exponential Smoothing (multiplicative seasonality)

1) We build and initialize the model on training data.

- method = estimated

2) Then we fit the model. Below, are the model parameters.

```
{'smoothing_level': 0.11128429736328378, 'smoothing_trend': 0.04947326762762311, 'smoothing_seasonal': 0.23037194388521623, 'da
mping_trend': nan, 'initial_level': 2803.0168193984414, 'initial_trend': 10.486286228443715, 'initial_seasons': array([0.802840
01, 0.86968748, 1.08266033, 0.93954787, 0.96331944,
       0.98854326, 1.0654188 , 1.28504436, 1.0083707 , 1.0929922 ,
       1.36460606, 1.41709466]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

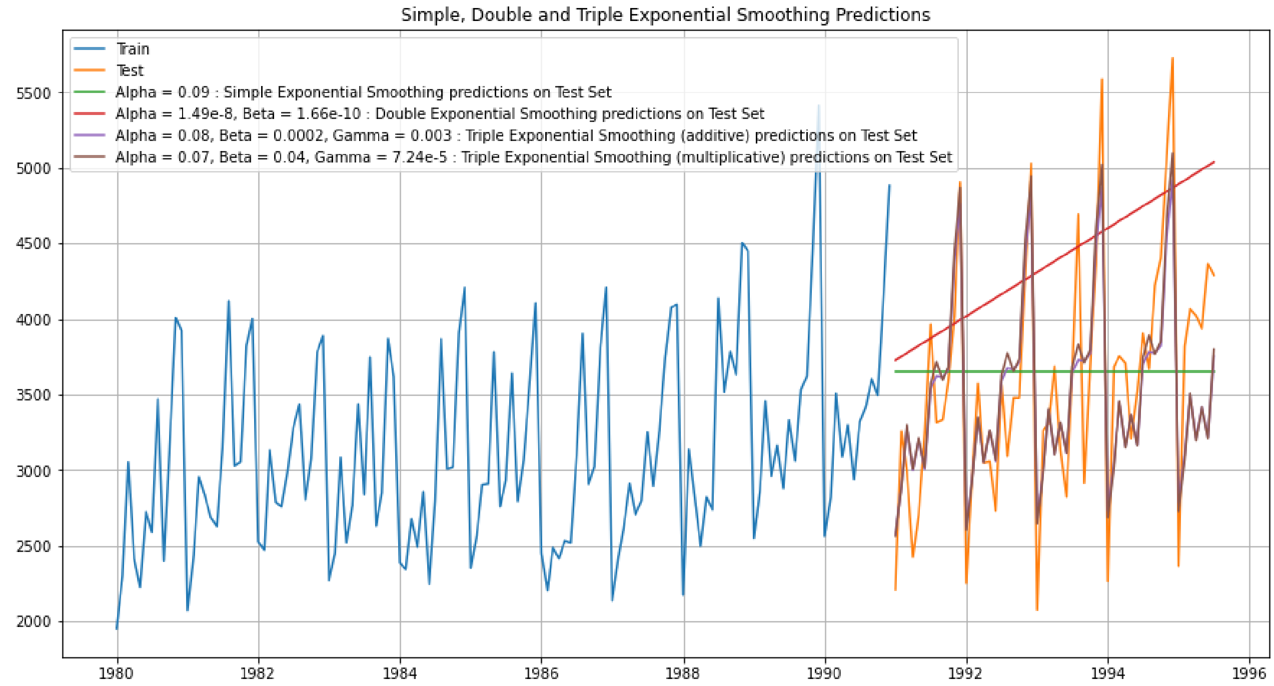


Figure 30: Comparing models built with test data

5) Afterwards, we calculate RMSE value for the model

```
TES_mul RMSE: 447.7225807439294
```

Model 5: Linear Regression model

1) For this particular linear regression, we are going to regress the 'SoftDrink' variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 3
4, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,
124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157,
158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187]
```

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

First few rows of Training Data

YearMonth	SoftDrinkProduction	time
1980-01-01	1954	1
1980-02-01	2302	2
1980-03-01	3054	3
1980-04-01	2414	4
1980-05-01	2226	5

First few rows of Test Data

YearMonth	SoftDrinkProduction	time
1991-01-01	2211	133
1991-02-01	3260	134
1991-03-01	2992	135
1991-04-01	2425	136
1991-05-01	2707	137

Last few rows of Training Data

YearMonth	SoftDrinkProduction	time
1990-08-01	3418	128
1990-09-01	3604	129
1990-10-01	3495	130
1990-11-01	4163	131
1990-12-01	4882	132

Last few rows of Test Data

YearMonth	SoftDrinkProduction	time
1995-03-01	4067	183
1995-04-01	4022	184
1995-05-01	3937	185
1995-06-01	4365	186
1995-07-01	4290	187

2) We then, build, initialize and fit the model on the training data with default parameters.
3) Once done, we then predict for test dataset.

4) And then, we visualize and compare the predicted and test data on plot

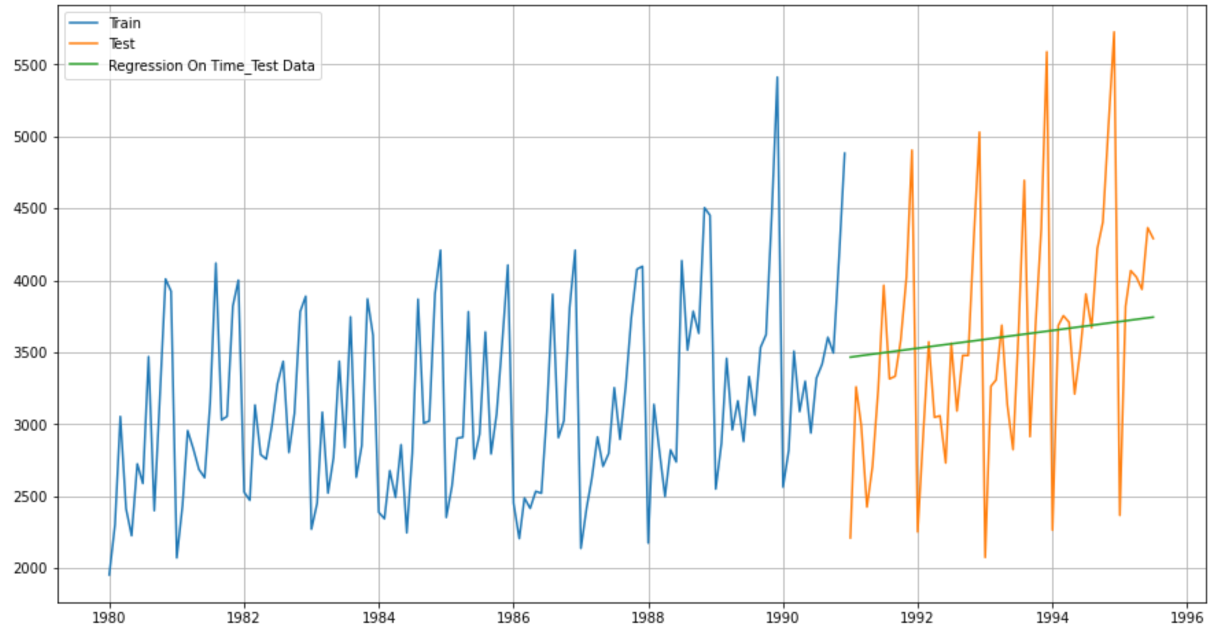


Figure 31: Comparing Linear Regression model built with test data

5) Afterwards, we calculate RMSE value for the model

```
For Regression Model on the Test Data,  RMSE is 775.808
```

Model 6: Naïve Model

1) We build, initialize and fit the model on the training data with default parameters.
2) Once done, we then predict for test dataset.
3) And then, we visualize and compare the predicted and test data on plot

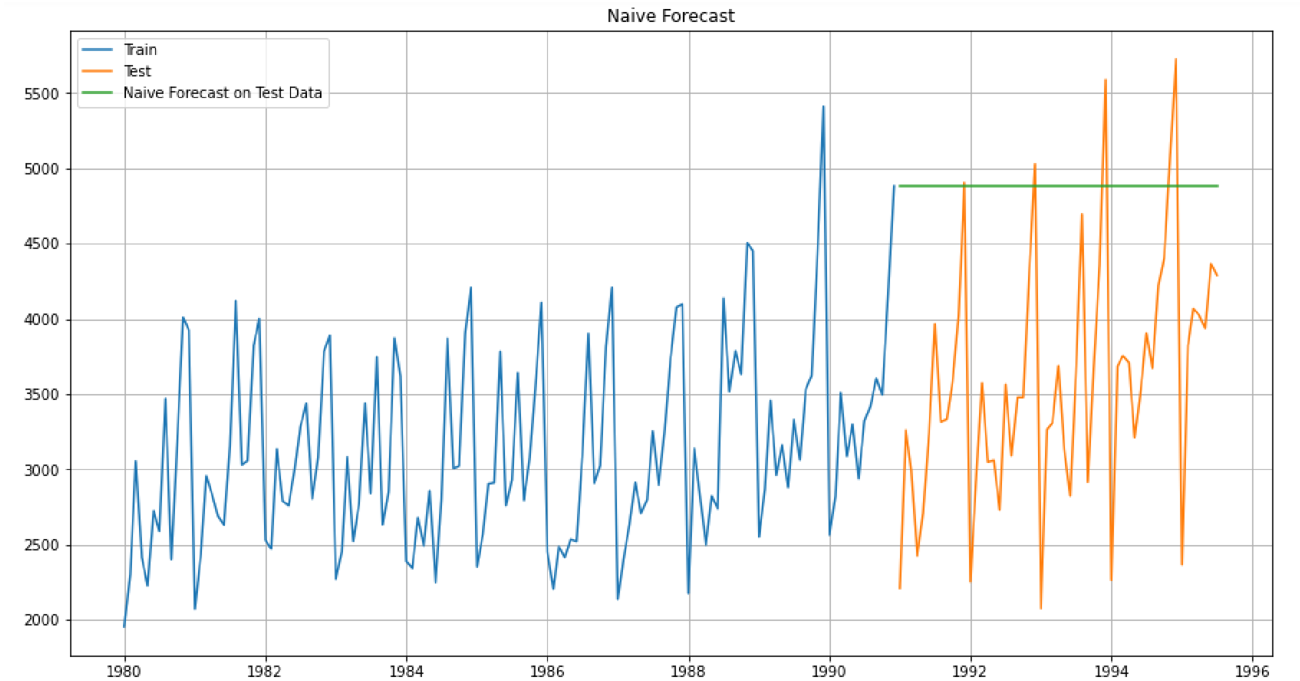


Figure 32: Comparing Naïve model built with test data

4) Afterwards, we calculate RMSE value for the model

```
For Naive model on the Test Data,  RMSE is 1519.259
```

Model 7: Simple Average Model

1) We build, initialize and fit the model on the training data with default parameters.
2) Once done, we then predict for test dataset.
3) And then, we visualize and compare the predicted and test data on plot

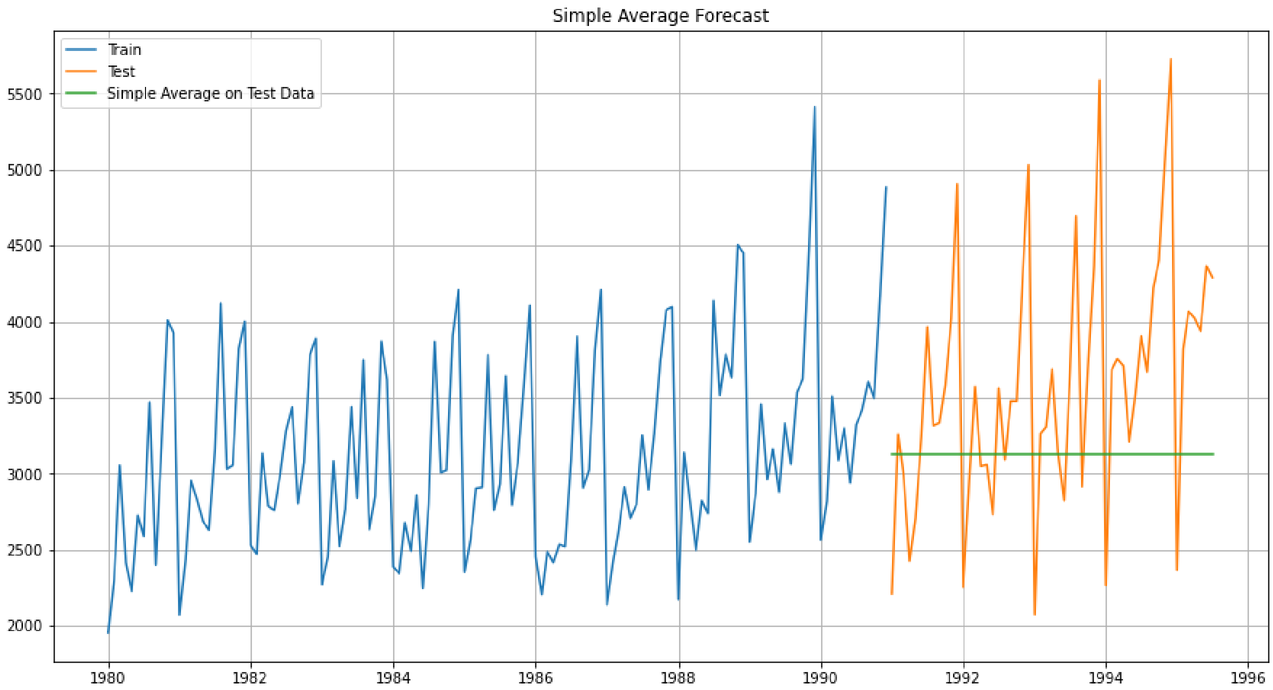


Figure 33: Comparing Simple Average model built with test data

4) Afterwards, we calculate RMSE value for the model

```
For Simple Average forecast on the Test Data,  RMSE is 934.353
```

Q2.5) 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. Stationarity should be checked at alpha = 0.05.

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

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

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

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

```
DF test statistic is -1.649
DF test p-value is 0.7726647141271693
Number of lags used 12
```

We see that at 5% significant level the Time Series is non-stationary.
Let us take one level of differencing to see whether the series becomes stationary.

```
DF test statistic is -7.271
DF test p-value is 3.4205181049970745e-09
Number of lags used 11
```

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

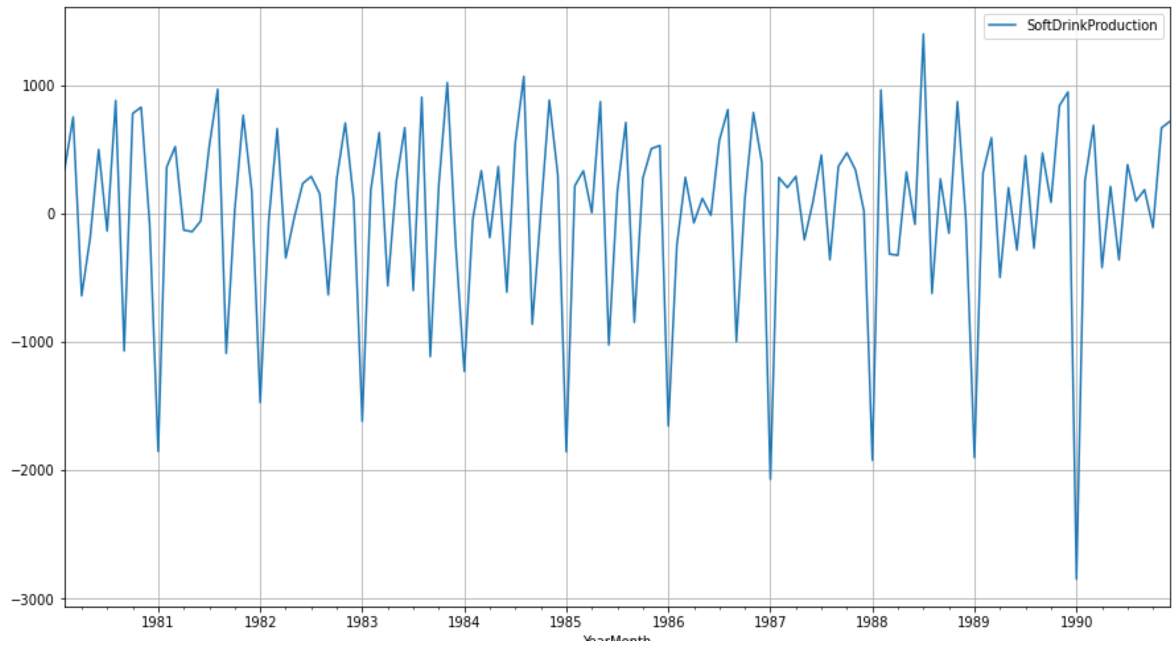
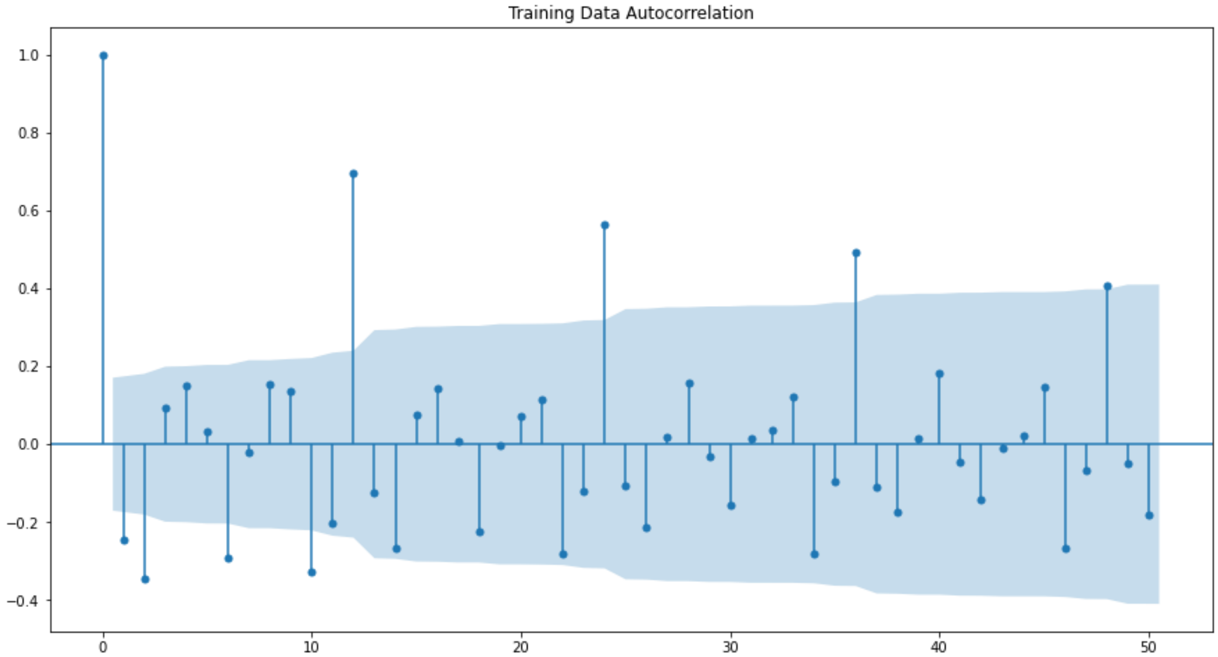


Figure 34: Stationary series

Q2.6) 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.

From decomposition we know that, the data contains seasonality. Hence, we would create SARIMA models as it accounts for seasonality
From the ACF plot we know there is a significant term after every 12 lags



1) We create some combinations for the SARIMA model

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (0, 1, 3)(0, 0, 3, 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: (1, 1, 3)(1, 0, 3, 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)
Model: (2, 1, 3)(2, 0, 3, 12)
Model: (3, 1, 0)(3, 0, 0, 12)
Model: (3, 1, 1)(3, 0, 1, 12)
Model: (3, 1, 2)(3, 0, 2, 12)
Model: (3, 1, 3)(3, 0, 3, 12)
```

2) We find AIC values for all the combinations and find the one with the least AIC value as it is best in terms of performance.

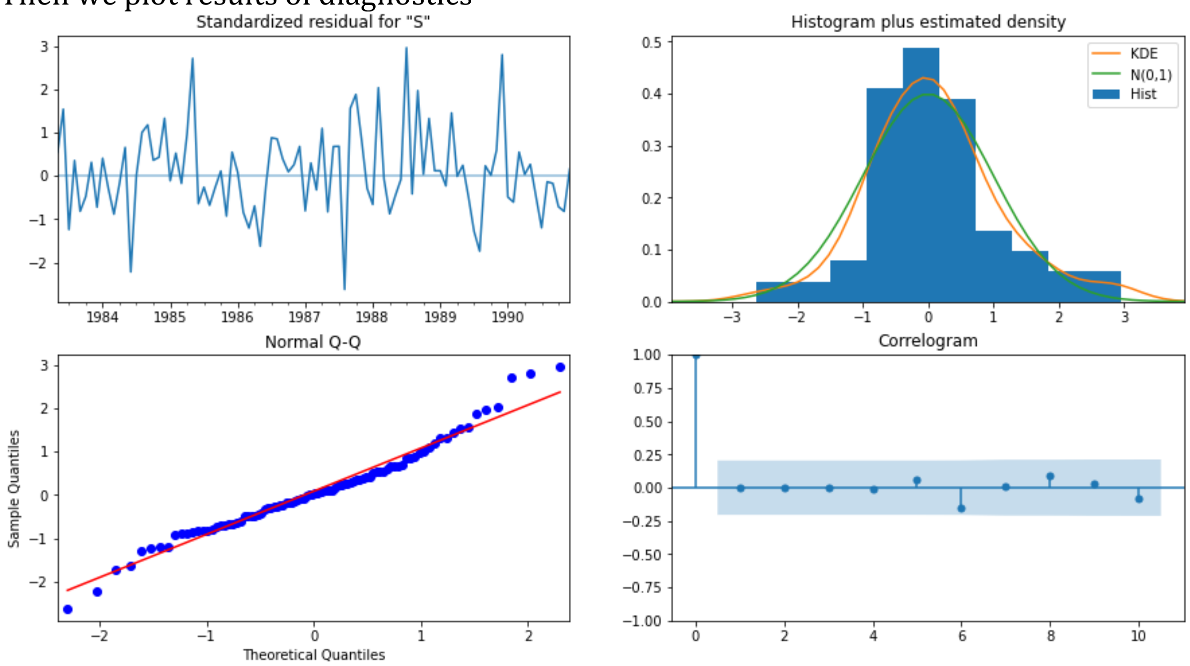
	param	seasonal	AIC
83	(1, 1, 1)	(0, 0, 3, 12)	12.000000
147	(2, 1, 1)	(0, 0, 3, 12)	14.000000
223	(3, 1, 1)	(3, 0, 3, 12)	22.000000
175	(2, 1, 2)	(3, 0, 3, 12)	125.724211
163	(2, 1, 2)	(0, 0, 3, 12)	171.500403

3) Then we find details of the best combination

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                        SoftDrinkProduction   No. Observations:                  132
Model:             SARIMAX(3, 1, 1)x(3, 0, [1, 2], 12)   Log Likelihood                -668.639
Date:                                Thu, 16 Jun 2022   AIC                           1357.277
Time:                                        13:50:50   BIC                           1382.495
Sample:                                    01-01-1980   HQIC                          1367.456
                                         - 12-01-1990
Covariance Type:                                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1000      0.143      0.701      0.483      -0.180       0.380
ar.L2         -0.0261      0.126     -0.207      0.836      -0.273       0.221
ar.L3          0.0777      0.147      0.529      0.597      -0.210       0.366
ma.L1         -0.9213      0.086    -10.775      0.000      -1.089      -0.754
ar.S.L12       0.5940      0.750      0.792      0.429      -0.877       2.065
ar.S.L24       0.3046      0.700      0.435      0.663      -1.067       1.676
ar.S.L36       0.0913      0.243      0.376      0.707      -0.385       0.568
ma.S.L12      -0.2144      0.764     -0.281      0.779      -1.711       1.283
ma.S.L24      -0.1275      0.476     -0.268      0.789      -1.061       0.806
sigma2      1.175e+05   1.66e+04      7.076      0.000      8.5e+04     1.5e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 5.85
Prob(Q):                              0.99   Prob(JB):                         0.05
Heteroskedasticity (H):               1.24   Skew:                             0.41
Prob(H) (two-sided):                  0.55   Kurtosis:                         3.92
===================================================================================

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

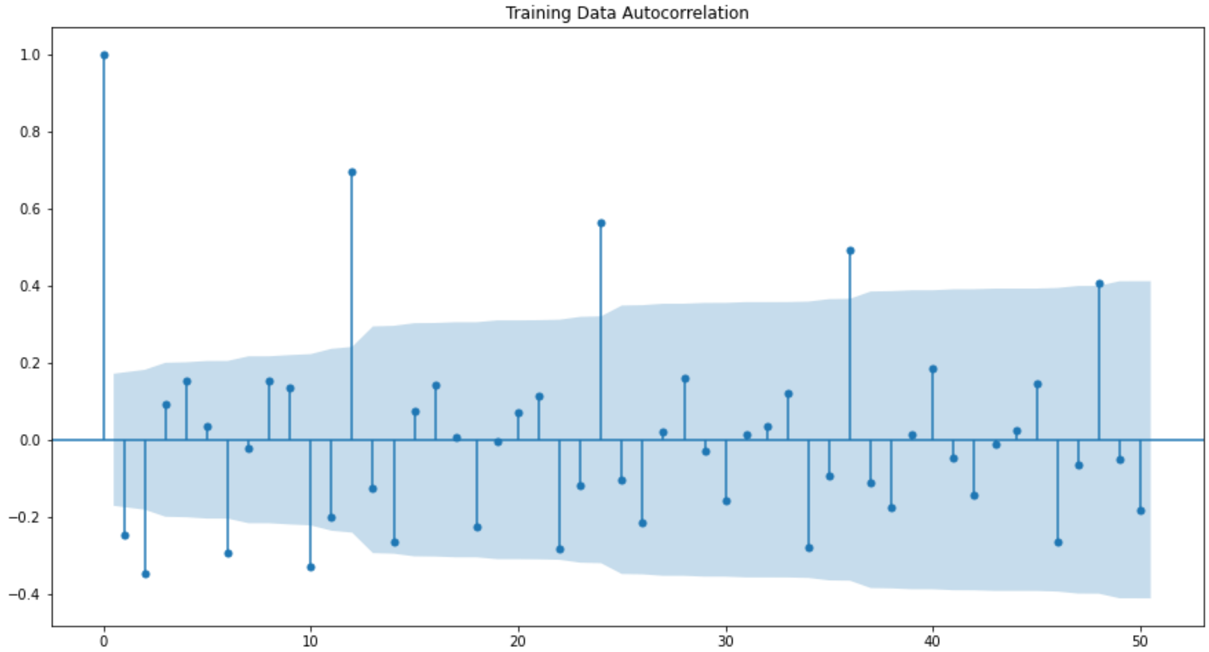
4) Then we plot results of diagnostics



5) Then, we find RMSE & MAPE values

```
RMSE: 427.63406007717424
MAPE: 10.875732384235762
```

Q2.7) 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.



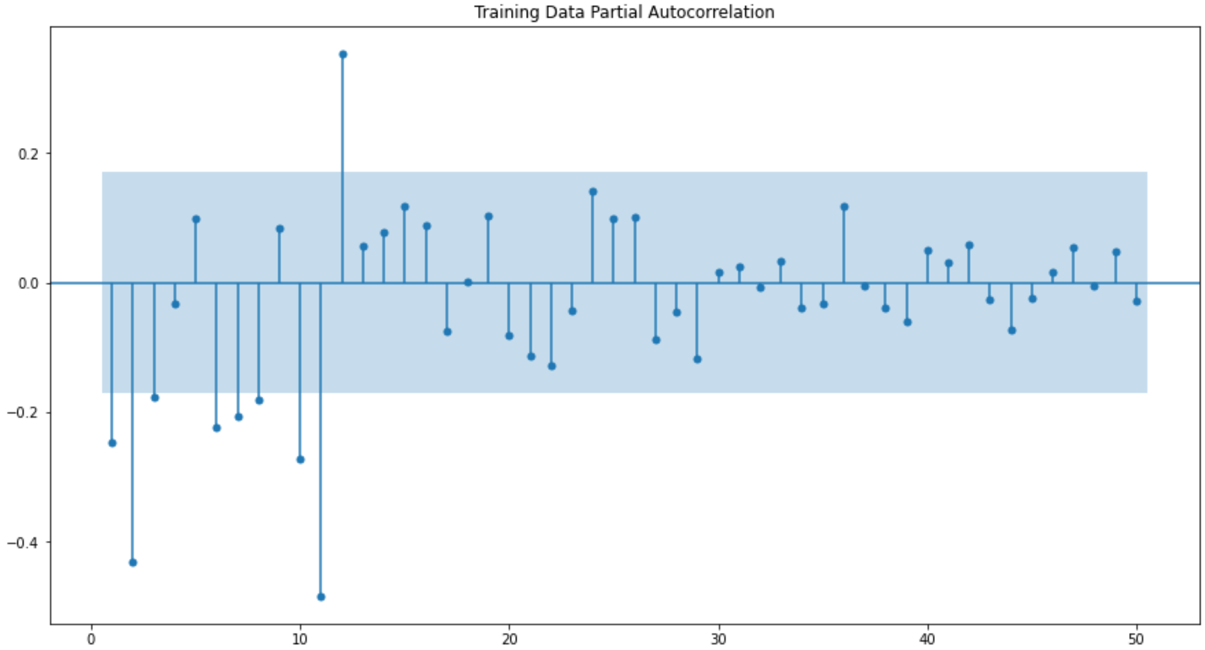


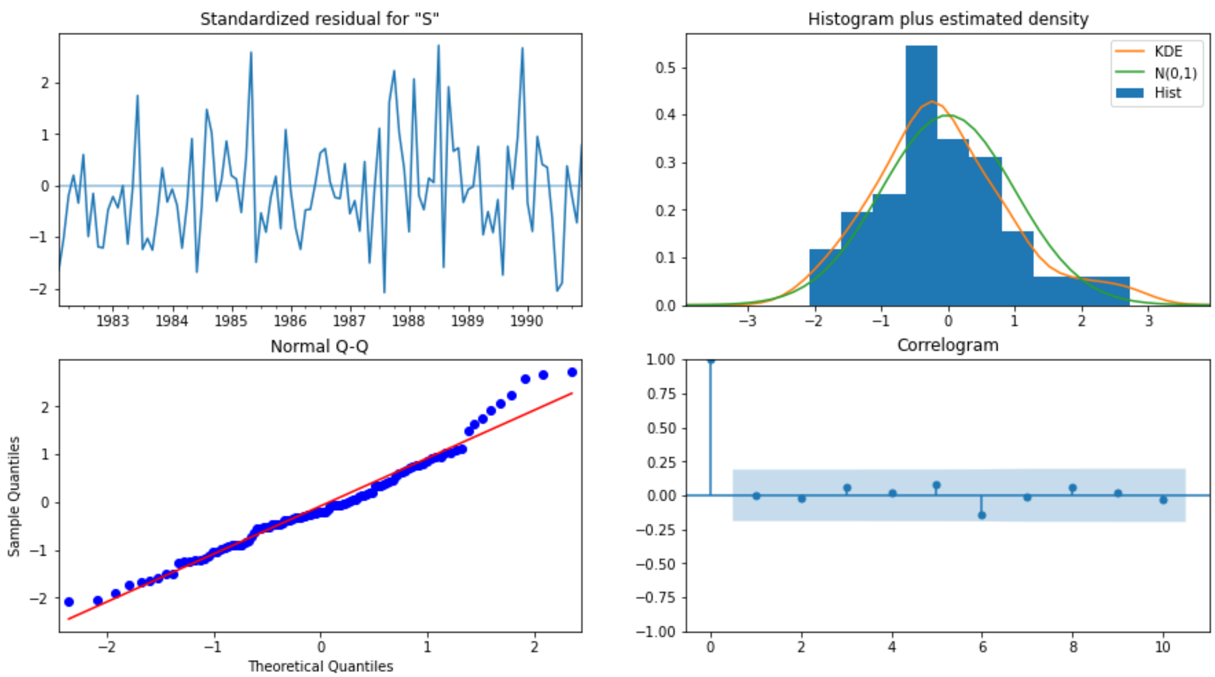
Figure 35: ACF & PACF Plot

From the ACF and PACF plots above, we can say that,
Cut-off of ACF & PACF both is 0 (i.e. 'p' & 'q')
Therefore, the order = (01,0). Since, the differencing done is of first order.
Also, seasonal cut-offs are seen after every 12 lags.
Therefore, the seasonal order = (0,0,12,12)
Results of combination decided.

```
                                                     SARIMAX Results
==========================================================================================================================
=
Dep. Variable:                                                     SoftDrinkProduction   No. Observations:             13
2
Model:             SARIMAX(2, 1, 2)x(0, 1, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], 24)   Log Likelihood             -785.89
7
Date:                                                                 Thu, 16 Jun 2022   AIC                        1605.79
4
Time:                                                                         14:42:28   BIC                        1651.23
2
Sample:                                                                     01-01-1980   HQIC                       1624.21
4
                                                                          - 12-01-1990
Covariance Type:                                                                   opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.0355      2.667     -0.388      0.698      -6.263       4.192
ar.L2         -0.0392      0.152     -0.258      0.797      -0.337       0.259
ma.L1          0.1506     12.292      0.012      0.990     -23.940      24.242
ma.L2         -0.8505     10.447     -0.081      0.935     -21.326      19.625
ma.S.L24      -0.7586    605.384     -0.001      0.999   -1187.289    1185.772
ma.S.L48       0.2140    817.387      0.000      1.000   -1601.835    1602.263
ma.S.L72       0.1338    938.650      0.000      1.000   -1839.586    1839.854
ma.S.L96       0.6544    929.411      0.001      0.999   -1820.958    1822.267
ma.S.L120      0.5339   1547.955      0.000      1.000   -3033.402    3034.470
ma.S.L144     -1.1744   1364.808     -0.001      0.999   -2676.148    2673.799
ma.S.L168      0.9472   1704.069      0.001      1.000   -3338.967    3340.861
ma.S.L192      0.1502    614.474      0.000      1.000   -1204.196    1204.497
ma.S.L216      0.6298   1237.347      0.001      1.000   -2424.526    2425.786
ma.S.L240     -0.3860    877.191     -0.000      1.000   -1719.648    1718.877
ma.S.L264     -0.2467   1196.689     -0.000      1.000   -2345.713    2345.220
ma.S.L288      0.7824   1937.665      0.000      1.000   -3796.972    3798.537
sigma2      2.802e+04      0.033   8.37e+05      0.000     2.8e+04     2.8e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 6.88
Prob(Q):                              0.97   Prob(JB):                         0.03
Heteroskedasticity (H):               1.68   Skew:                             0.58
Prob(H) (two-sided):                  0.13   Kurtosis:                         3.45
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 1.36e+25. Standard errors may be unstable.
```

Plotting results of diagnostics



RMSE & MAPE value

```
RMSE: 608.0069198004215
MAPE: 13.222030959955845
```

Q2.8) Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

	Test RMSE
Alpha = 0.09 : SES	809.501640
Alpha = 1.49e-8, Beta = 1.66e-10 : DES	1074.329153
Alpha = 0.08, Beta = 0.0002, Gamma = 0.003 : TES_add	458.965392
Alpha = 0.07, Beta = 0.04, Gamma = 7.24e-5 : TES_mul	447.722581
Regression Model	775.807810
Naive Model	1519.259233
Simple Average Model	934.353358
SARIMA_Auto(3,1,1)(3,0,2,12)	427.634060
SARIMA_Manual(2,1,2)(0,1,12,24)	608.006920

Table 6: Models with their corresponding RMSE values

From the table above we can conclude that,

Since, the RMSE value of Triple Exponential Smoothing model (additive seasonality) is the least, it is the best performing model. Hence, we would use that model to predict or forecast further sales.

Q2.9) 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.

By comparing the RMSE values of all the models built, we find that, Triple Exponential Smoothing model as it has the lowest RMSE value, which means best performance as we know.

So, we fit the entire data into that model for the entire data

```
{'smoothing_level': 0.11238627876980502, 'smoothing_trend': 0.08744849128598821, 'smoothing_seasonal': 0.296746510658254, 'damp
ing_trend': nan, 'initial_level': 2802.9504126041684, 'initial_trend': 12.129001433987412, 'initial_seasons': array([-687.45296
47 , -582.75987466,  -55.45060893, -365.70168671,
       -253.53771449, -196.42589555,  -32.66701439,  690.26106557,
       -282.32490047,   44.77261006,  867.39344232,  853.63543195]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Then, we forecast for 12 months in the future and calculate RMSE value.

```
RMSE of the Full Model 332.3853901348161
```

Now, we plot the forecast to visualize it

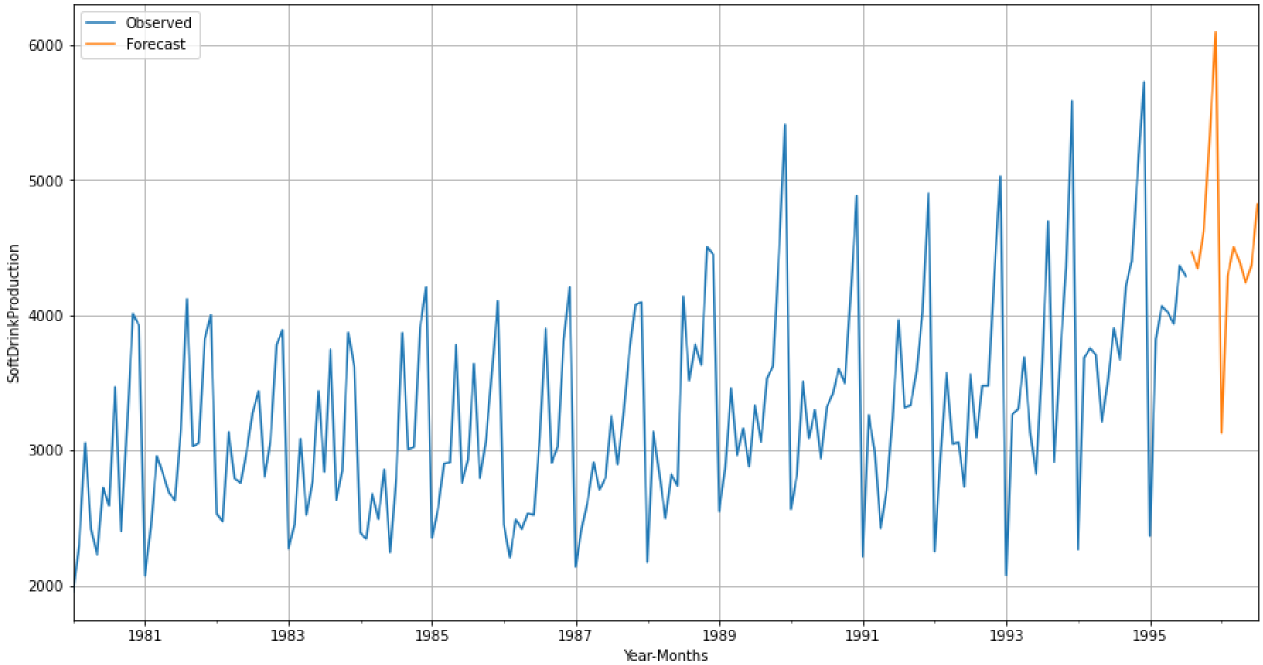


Figure 36: Observed & Forecasted Sales plot

Q2.10) Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

- From the final model used to forecast future 12 months production suggest that, the production will be morethan the previous 12 months.
- Model performs very well and can be used to make business decisions which could further increase production or used to devise new marketing strategies.
- The trend observed from the dataset provided suggests that, the last month has the highest production compared to the other months which have average production. So, we need to decide which partof the year’s production we need to boost to maximize the overall profit.
- We should also understand the average market soft drink production & our leading competitors’ production to understand our performance in the market much better.