Forecasting on Air Passengers Dataset

Objective

- Objective - Build a model to forecast the demand(passenger traffic) in Airplanes.

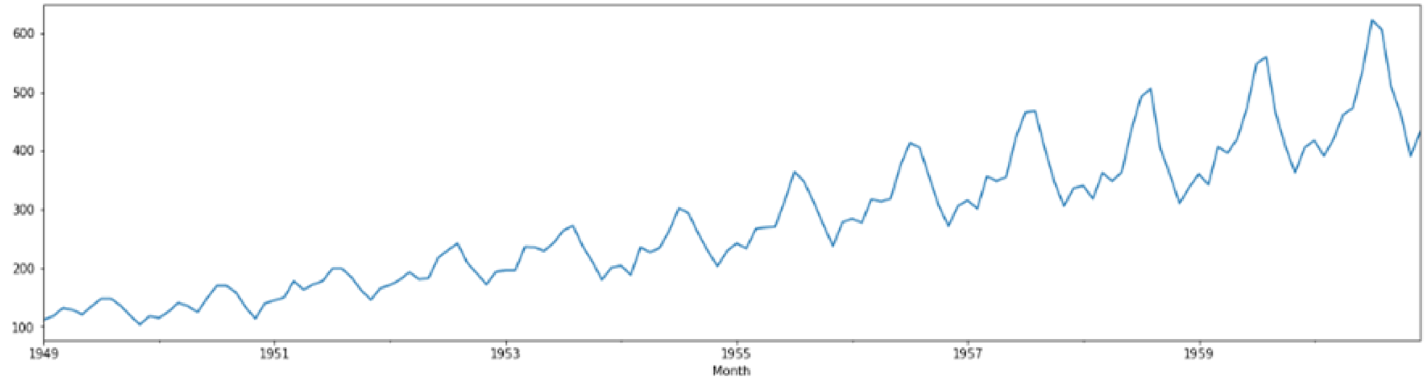
About the Dataset:

- Number of observations: 144
- Variables in the Dataset - Passengers.
- The data is classified in date and the passengers travelling per month.
- There are no missing values in the dataset.

Visualizing Passenger Data:

```
data['Passengers'].plot(figsize =(20,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ccab2a7408>
```



- From the above plot we can observe that, there is some trend and seasonality in the time series
- X - axis: Months
- Y - axis: Number of Passengers

Checking for Stationarity:

Rolling Mean

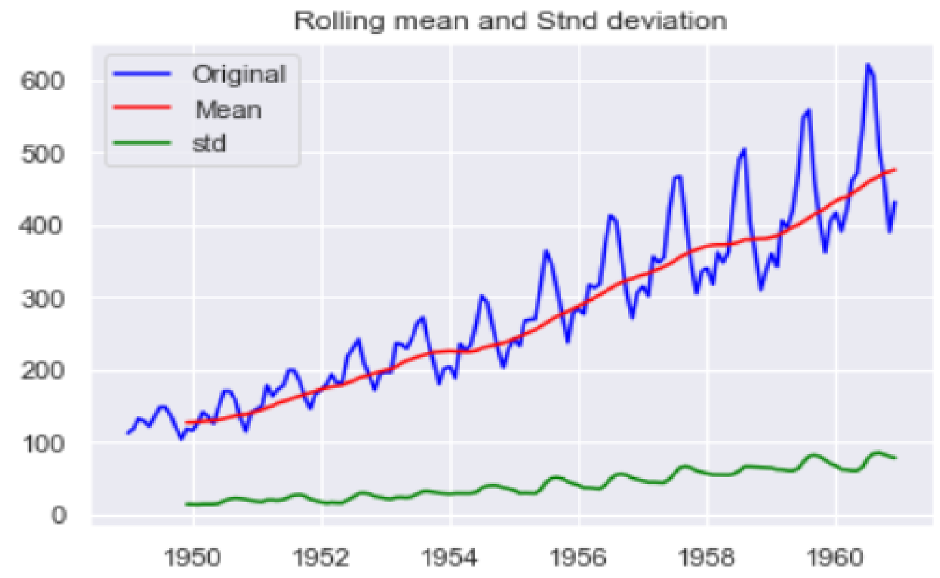
Rolling mean is the test of stationarity in the data.

```
rolmean = IndexedDF.rolling(window=12).mean()
rolstd = IndexedDF.rolling(window=12).std()
print(rolmean,rolstd)
```

```
            Passengers
Month
1949-01-01         NaN
1949-02-01         NaN
1949-03-01         NaN
1949-04-01         NaN
1949-05-01         NaN
...                ...
1960-08-01  463.333333
1960-09-01  467.083333
1960-10-01  471.583333
1960-11-01  473.916667
1960-12-01  476.166667
```

As we can see from the diagram that the rolling mean and Standard Deviation increase with time, we can conclude that the time series is not stationary.

```
orig = plt.plot(IndexedDF,color = 'blue',label = 'Original')
mean =  plt.plot(rolmean,color = 'red',label = 'Mean')
std =  plt.plot(rolstd,color = 'green',label = 'std')
plt.legend(loc = 'best')
plt.title('Rolling mean and Stnd deviation')
plt.show(block = False)
```



Checking for Stationarity:

ACDF Test:

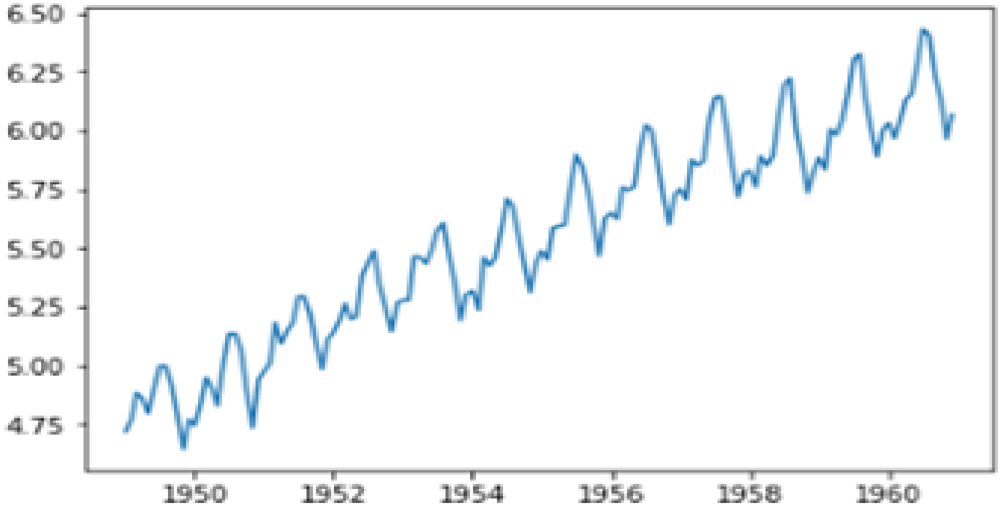
- We can clearly observe that the p value is greater than 0.05, suggesting that the data is not stationary
- The underlying principle is to model or estimate the trend and seasonality in the series and remove those from the series to get a stationary series.

```
Results of the ACDF test :
Test statistics              0.815369
p-value                      0.991880
#lags used                  13.000000
# observations used        130.000000
Critical Value (1%)         -3.481682
Critical Value (5%)         -2.884042
Critical Value (10%)        -2.578770
dtype: float64
```

Estimating the Trend

```
data_log = np.log(data.Passengers)
plt.plot(data_log)
```

```
[<matplotlib.lines.Line2D at 0x1cca9d91cc8>]
```

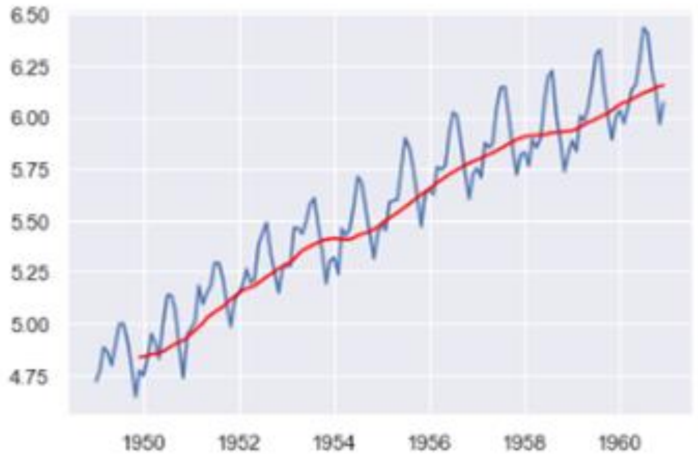


Taking the log of the dependant variable is a simple way of lowering the rate at which rolling mean increases.

Data Stationarity Test - Rolling Mean Method

```
movingAverage = IndexedDF_logScale.rolling(window = 12).mean()
movingstd = IndexedDF_logScale.rolling(window = 12).std()
plt.plot(IndexedDF_logScale)
plt.plot(movingAverage,color = 'red')
```

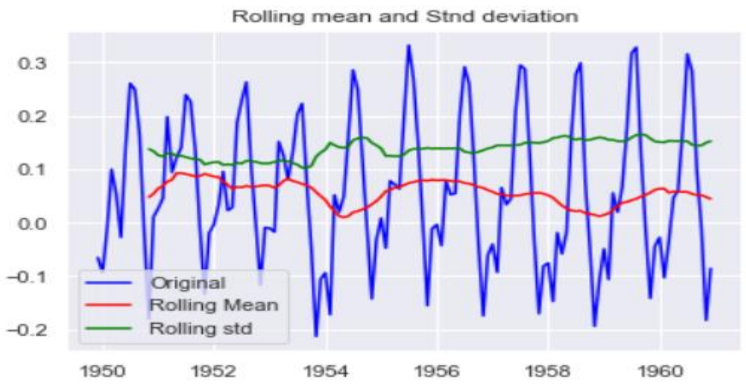
```
[<matplotlib.lines.Line2D at 0x1c6dc34b188>]
```



- For the graph on the left, we can see that although rolling mean is not stationery, it is still better than the previous case where no transformation were applied to series.
- We know from the graph that both time series with log scale as well as moving average have a trend component. Subtracting one from the other should remove the trend component in both.

```
datasetLogScaleMinusMovingAverage = IndexedDF_logScale - movingAverage
datasetLogScaleMinusMovingAverage .head(12)
```

```
#remove the NaN values
datasetLogScaleMinusMovingAverage.dropna(inplace = True)
datasetLogScaleMinusMovingAverage .head(10)
```




```
Results of the ACDF test :
Test statistics            -3.162908
p-value                     0.022235
#lags used                 13.000000
# observations used       119.000000
Critical Value (1%)        -3.486535
Critical Value (5%)        -2.886151
Critical Value (10%)       -2.579896
dtype: float64
```

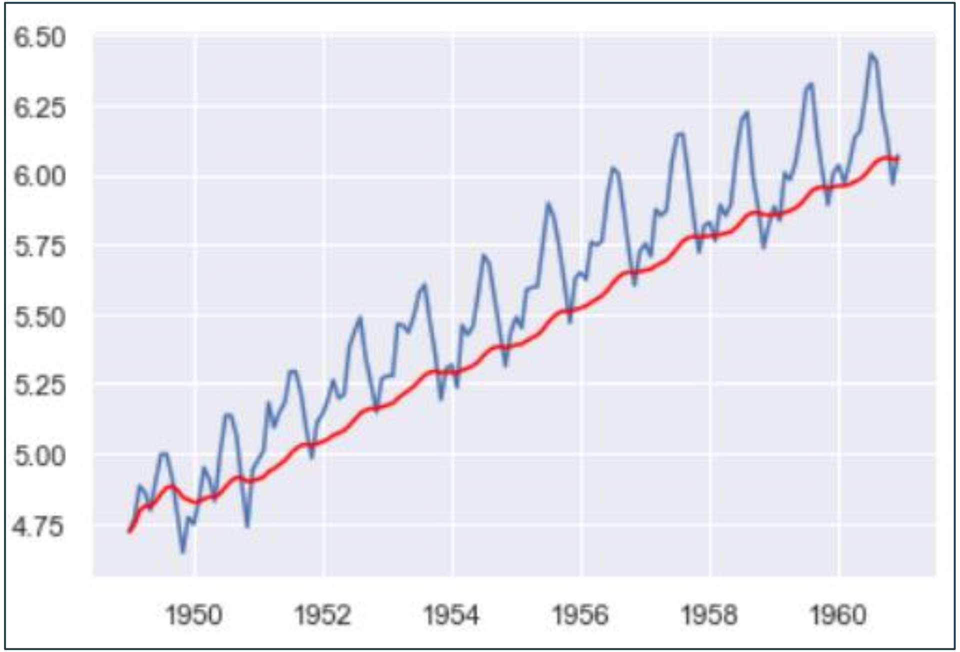
- We found that our assumption of subtracting two related series of similar trend components will make the result stationery was true.

1. P value has reduced from 0.99 to 0.022
2. ADF Test Statistic is close to the critical values

- Thus, we can say that the given series is stationary.

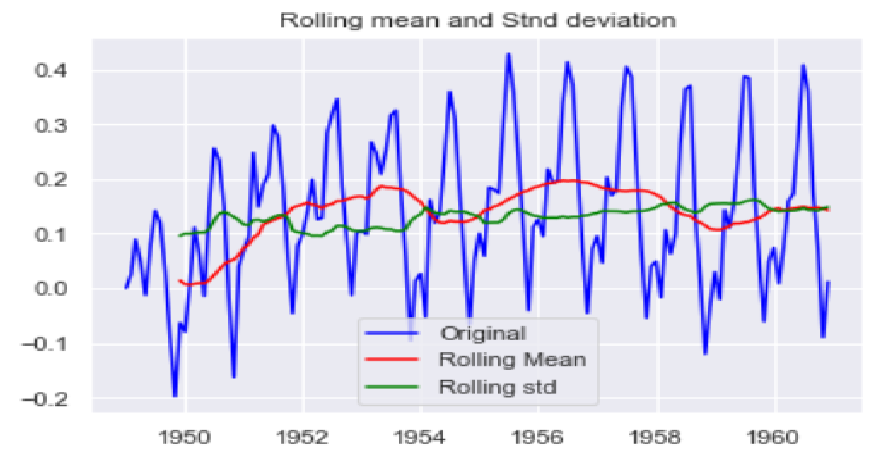
Making Data Stationary - Exponential Decay Method

```
exponentialDecayingWeightedAverage = IndexedDF_logScale.ewm(halflife=12,min_periods=0, adjust = True).mean()
plt.plot(IndexedDF_logScale)
plt.plot(exponentialDecayingWeightedAverage, color = 'red')
```



- For the given graph, it seems that the given method is not showing any advantage over log scale as corresponding curves are similar.
- Since no concrete inference can be drawn, we perform ADF test on the decay series.

```
datasetlogScaleMinusexponentialDecayingWeightedAverage = IndexedDF_logScale - exponentialDecayingWeightedAverage
test_stationarity(datasetlogScaleMinusexponentialDecayingWeightedAverage)
```




```
Results of the ACDF test :
Test statistics              -3.601262
p-value                       0.005737
#lags used                   13.000000
# observations used         130.000000
Critical Value (1%)          -3.481682
Critical Value (5%)          -2.884042
Critical Value (10%)         -2.578770
dtype: float64
```

We observe that the time series is stationary and the series for moving average and std deviation is almost parallel to x-axis, thus they also have no trend.

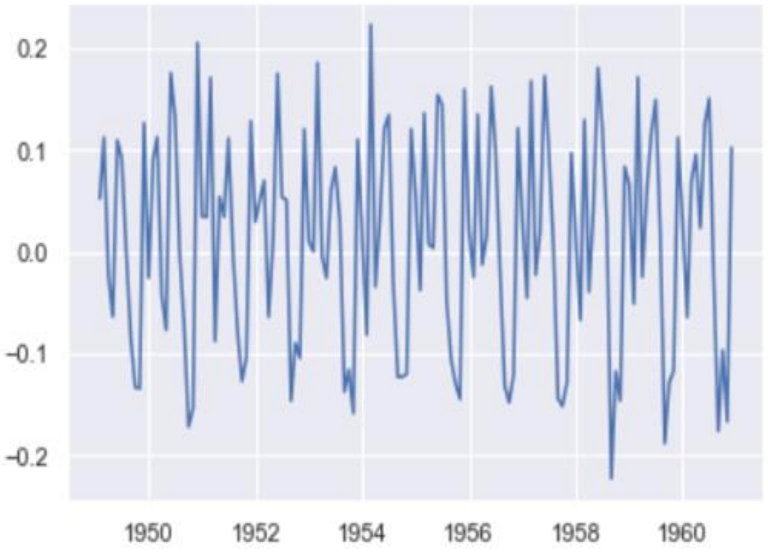
Additionally,

1. P value decreased from 0.022 to 0.005
2. Test statistic is closer to critical values.

Making Data Stationary - Differencing Method

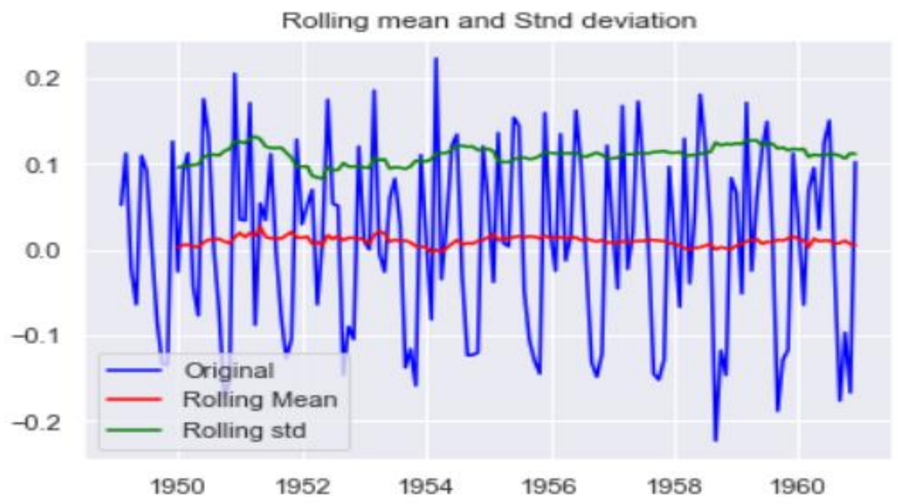
```
datasetlogDiffShifting = IndexedDF_logScale - IndexedDF_logScale.shift()
plt.plot(datasetlogDiffShifting)
```

[<matplotlib.lines.Line2D at 0x1d7ce504608>]




```
datasetlogDiffShifting.dropna(inplace = True)
test_stationarity(datasetlogDiffShifting)
```

	Passengers
Month	
1949-01-01	NaN
1949-02-01	NaN
1949-03-01	NaN
1949-04-01	NaN
1949-05-01	NaN
...	...
1960-08-01	463.333333
1960-09-01	467.083333
1960-10-01	471.583333
1960-11-01	473.916667
1960-12-01	476.166667




```
Results of the ACDF test :
Test statistics              -2.717131
p-value                       0.071121
#lags used                   14.000000
# observations used         128.000000
Critical Value (1%)          -3.482501
Critical Value (5%)          -2.884398
Critical Value (10%)         -2.578960
dtype: float64
```

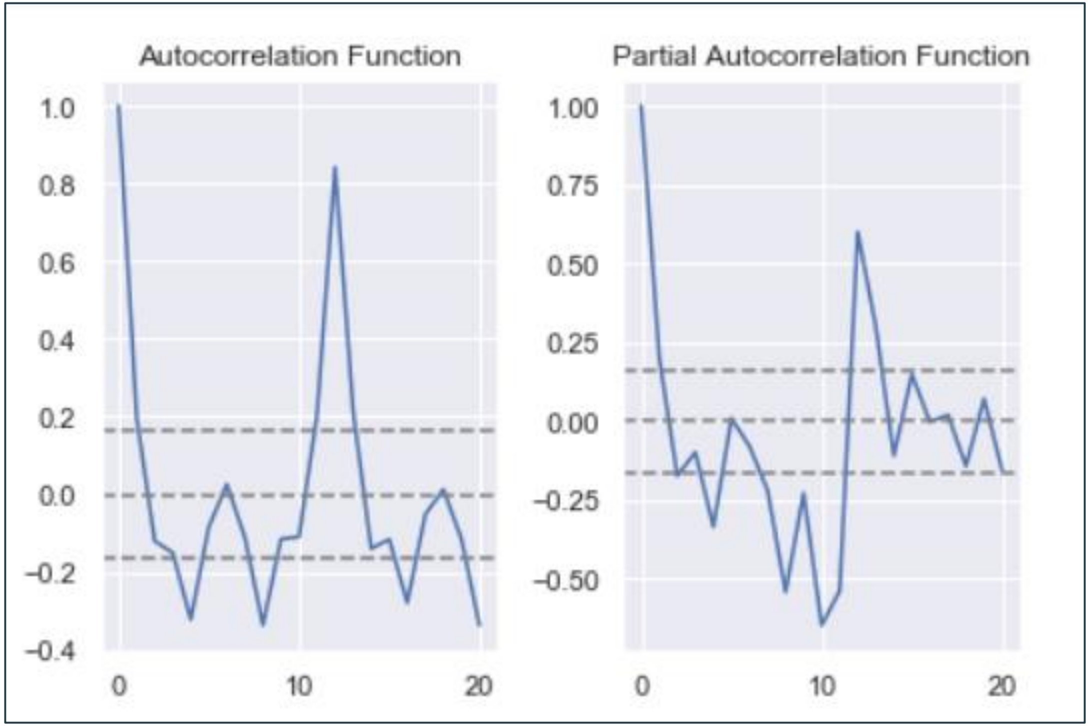
The ACDF Result shows that:

1. P value of 0.07 is not as good as 0.005 of exponential decay.
2. The test statistic value is not as close to the critical values as that for exponential decay.

ACF and PACF

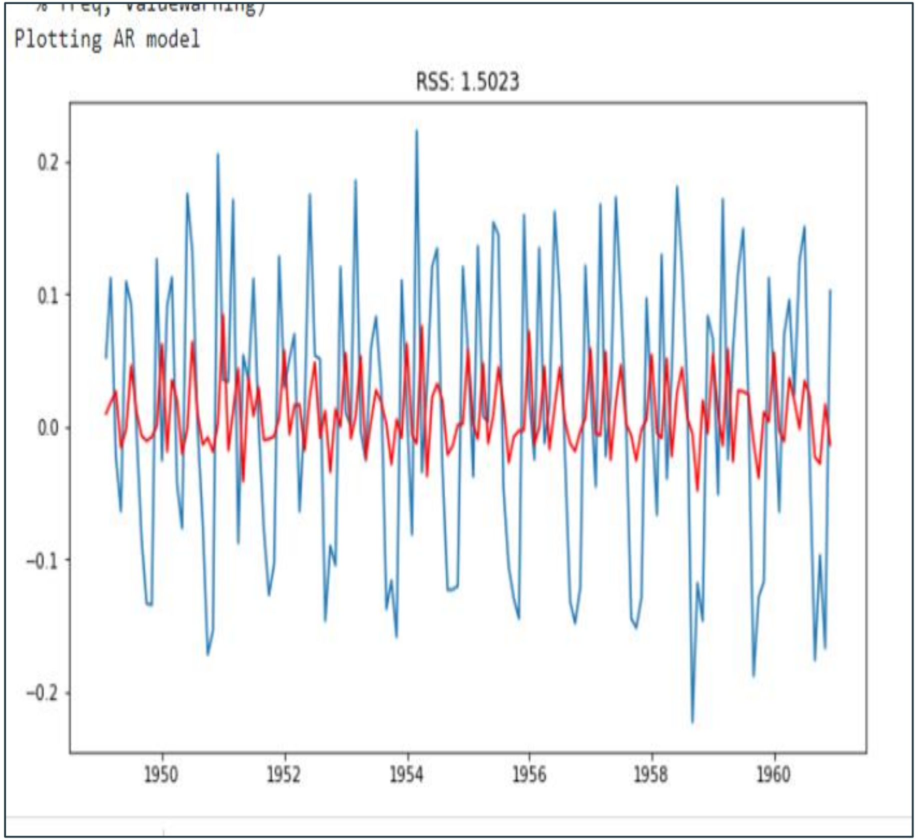
- ACF is auto-correlation between the elements of a series and others from the same series separated from them by a given interval

- PACF gives the partial correlation of a stationary time series with its own lagged values
- The value of q from ACF plot is 2
- The value of p from PACF plot is 2



AR Model

- Before, we see an ARIMA model, let us check the results of the individual AR & MA model. These models will give a value of RSS. Lower RSS values indicate a better model.
- **Residual sum of squares (RSS)/sum of squared residuals (SSR)/sum of squared estimate of errors (SSE)** is a measure of the discrepancy between the data and an estimation model.
- A small RSS indicates a tight fit of the model to the data. It is used as an optimality criterion in parameter selection and model selection
- Here in this AR model gives lower RSS of 1.5023 at order 2,1,0.



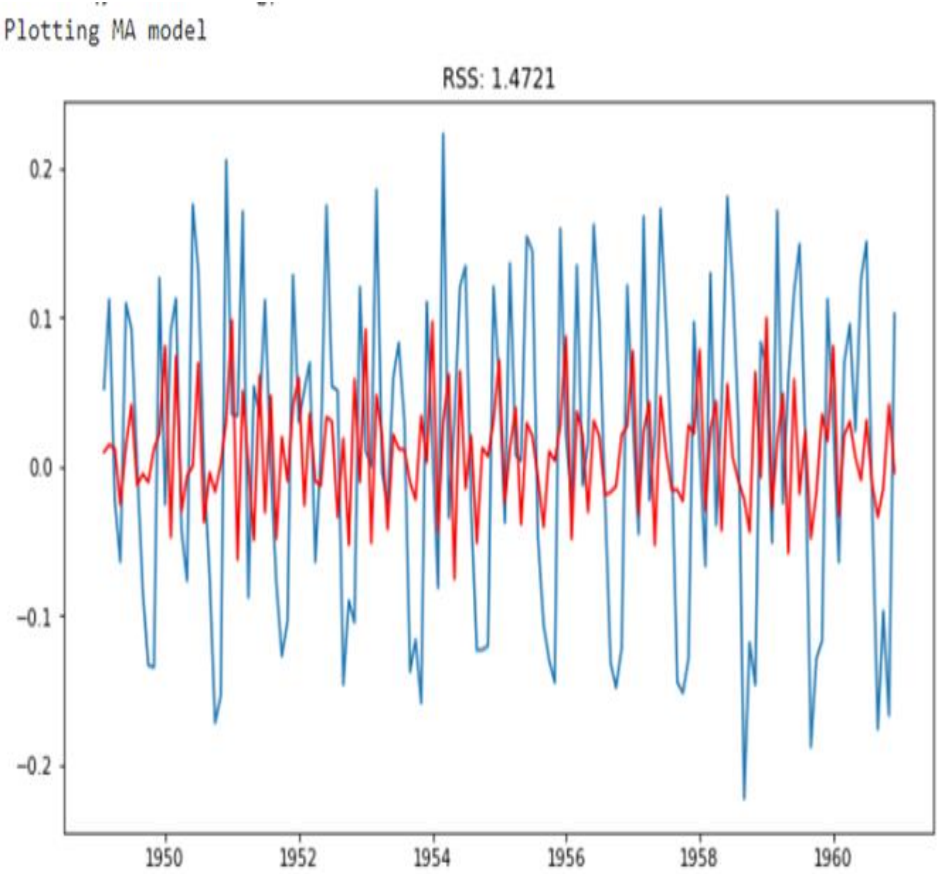
AR Model Summary

```
print(results_AR.summary())

                             ARIMA Model Results
==============================================================================
Dep. Variable:            D.#Passengers   No. Observations:                  143
Model:                   ARIMA(2, 1, 0)   Log Likelihood                 122.802
Method:                         css-mle   S.D. of innovations              0.102
Date:                  Wed, 16 Sep 2020   AIC                           -237.605
Time:                          20:38:35   BIC                           -225.753
Sample:                      02-01-1949   HQIC                          -232.789
                           - 12-01-1960
=======================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------
const                   0.0096      0.009      1.048      0.295      -0.008       0.028
ar.L1.D.#Passengers     0.2359      0.083      2.855      0.004       0.074       0.398
ar.L2.D.#Passengers    -0.1725      0.083     -2.070      0.038      -0.336      -0.009
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.6838           -2.3088j            2.4079           -0.2042
AR.2            0.6838           +2.3088j            2.4079            0.2042
-----------------------------------------------------------------------------
```

- Here we can see that p values for AR L1 and AR L2 is less than 0.05
- Therefore they seem significant. The value for AIC and BIC values are -237.605 and -225.753 respectively.

MA Model



Moving Average Model (MA)

- Assumes the value of the dependent variable on the current day depends on the previous days error terms
- The MR model gives lower RSS of 1.4721 at order 0,1,2
- As in both the models RSS value is comparatively less.

MA Model Summary

```
print(results_MA.summary())

                             ARIMA Model Results
==============================================================================
Dep. Variable:         D.#Passengers   No. Observations:                  143
Model:                ARIMA(0, 1, 2)   Log Likelihood                 124.189
Method:                      css-mle   S.D. of innovations              0.101
Date:               Wed, 16 Sep 2020   AIC                           -240.379
Time:                       20:40:31   BIC                           -228.528
Sample:                   02-01-1949   HQIC                          -235.563
                        - 12-01-1960
=====================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                   0.0096      0.007      1.314      0.189      -0.005      0.024
ma.L1.D.#Passengers     0.2019      0.120      1.688      0.091      -0.033      0.436
ma.L2.D.#Passengers    -0.3409      0.188     -1.814      0.070      -0.709      0.027
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -1.4419           +0.0000j            1.4419            0.5000
MA.2            2.0342           +0.0000j            2.0342            0.0000
-----------------------------------------------------------------------------
```

Here we can see that p values for MA L1 and MA L2 values are more than 0.05. Therefore they are insignificant. The value for AIC and BIC values are -240.379 and -228.528 respectively.

Now we will combine AR and MA model into ARIMA model and see whether the RSS value has decreased or not. The model with the lowest RSS and AIC & BIC value will be used for predictions. We will also look at whether the all components are significant.

ARIMA Model

```
model_ar2ma = ARIMA(data_log, order=(2, 1, 2))
results_ARIMA = model_ar2ma.fit(disp=-1)
plt.plot(data_log_diff)
plt.plot(results_ARIMA.fittedvalues, color='red')
print(results_ARIMA.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:           D.Passengers   No. Observations:                  143
Model:                 ARIMA(2, 1, 2)   Log Likelihood                 149.640
Method:                       css-mle   S.D. of innovations              0.084
Date:                Thu, 17 Sep 2020   AIC                           -287.281
Time:                        00:02:41   BIC                           -269.504
Sample:                    02-01-1949   HQIC                          -280.057
                         - 12-01-1960
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
const                  0.0096      0.003      3.697      0.000       0.005       0.015
ar.L1.D.Passengers     1.6293      0.039     41.868      0.000       1.553       1.706
ar.L2.D.Passengers    -0.8946      0.039    -23.127      0.000      -0.970      -0.819
ma.L1.D.Passengers    -1.8270      0.036    -51.303      0.000      -1.897      -1.757
ma.L2.D.Passengers     0.9245      0.036     25.568      0.000       0.854       0.995
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.9106           -0.5372j            1.0573           -0.0848
AR.2            0.9106           +0.5372j            1.0573            0.0848
MA.1            0.9881           -0.3245j            1.0400           -0.0505
MA.2            0.9881           +0.3245j            1.0400            0.0505
-----------------------------------------------------------------------------
```

- In this case we observe that all our AR and MA components are significant.
- Hence we can consider this as a best fit model.

ARIMA Model	AIC	BIC
(1,1,1)	-241.6	-229.8
(1,1,2)	-265.2	-250.4
(2,1,1)	-270.2	-255.3
(2,1,2)	-287.3	-269.5

Ljung Box test

H0: The model does not show lack of fit
H1: The model exhibits lack of fit

Since the p-value<0.05. We reject the null hypothesis which means this model exhibits lack of fit.

In order to overcome this we will apply auto arima on out dataset to get a best fit model.

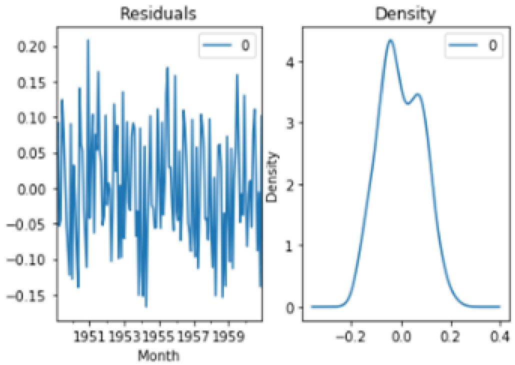
	lb_stat	lb_pvalue
1	0.008330	9.272774e-01
2	5.455342	6.537138e-02
3	5.671274	1.287459e-01
4	11.161776	2.480479e-02
5	13.539662	1.881372e-02
6	23.208052	7.297086e-04
7	23.650007	1.312438e-03
8	36.733352	1.288387e-05
9	38.322395	1.525618e-05
10	52.570518	8.945834e-08
11	54.051012	1.155342e-07
12	129.038668	9.623163e-22

Applying Auto ARIMA

```
Best model:  ARIMA(3,1,3)(0,0,0)[0] intercept
Total fit time: 5.276 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  144
Model:               SARIMAX(3, 1, 3)   Log Likelihood                 146.806
Date:                Thu, 17 Sep 2020   AIC                           -277.612
Time:                        10:44:43   BIC                           -253.909
Sample:                             0   HQIC                          -267.980
                                - 144
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0049      0.002      3.051      0.002       0.002       0.008
ar.L1          0.5566      0.149      3.724      0.000       0.264       0.849
ar.L2          0.5686      0.137      4.153      0.000       0.300       0.837
ar.L3         -0.6247      0.096     -6.506      0.000      -0.813      -0.437
ma.L1         -0.7073      0.172     -4.122      0.000      -1.044      -0.371
ma.L2         -0.9321      0.067    -14.005      0.000      -1.063      -0.802
ma.L3          0.6961      0.149      4.684      0.000       0.405       0.987
sigma2         0.0081      0.002      5.325      0.000       0.005       0.011
===================================================================================
Ljung-Box (Q):                      292.60   Jarque-Bera (JB):                 5.93
Prob(Q):                              0.00   Prob(JB):                         0.05
Heteroskedasticity (H):               1.04   Skew:                             0.06
Prob(H) (two-sided):                  0.88   Kurtosis:                         2.01
===================================================================================
```

Auto Arima model

```
# Plot residual errors
residuals1 = pd.DataFrame(results_ARIMA.resid)
fig, ax = plt.subplots(1,2)
residuals1.plot(title="Residuals", ax=ax[0])
residuals1.plot(kind='kde', title='Density', ax=ax[1])
plt.show()
```




```
acorr_ljungbox(residuals1, lags = 12)
```

```
(array([ 0.29499887,  0.53242258,  0.69969247,  2.68826059,  3.81646287,
        11.27093801, 11.52508214, 18.89257608, 19.33691629, 24.91524491,
        26.43116691, 91.63987923]),
 array([5.87034800e-01, 7.66277206e-01, 8.73276280e-01, 6.11270416e-01,
        5.76131853e-01, 8.03551183e-02, 1.17298667e-01, 1.54448292e-02,
        2.24757836e-02, 5.50849239e-03, 5.59591104e-03, 2.37515033e-14]))
```