AirPassengers-Copy1 (1)

August 19, 2022

[1]:
```
import pandas as pd
import numpy as np
from datetime import datetime as dt
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller, acf, pacf
from statsmodels.tsa.arima_model import ARIMA
```

[2]:
```
data = pd.read_csv('AirPassengers.csv')
data.head()
```

[2]:

	Month	#Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121

Correct the date format. (Complete it in the dd-mm-yyyy format)

[3]:
```
#Adding the date to my Time index i.e. YYYY-MM-DD
data['Month'] = data['Month'].apply(lambda x : dt(int(x[:4]), int(x[5:]), 15))
data.head()
```

[3]:

	Month	#Passengers
0	1949-01-15	112
1	1949-02-15	118
2	1949-03-15	132
3	1949-04-15	129
4	1949-05-15	121

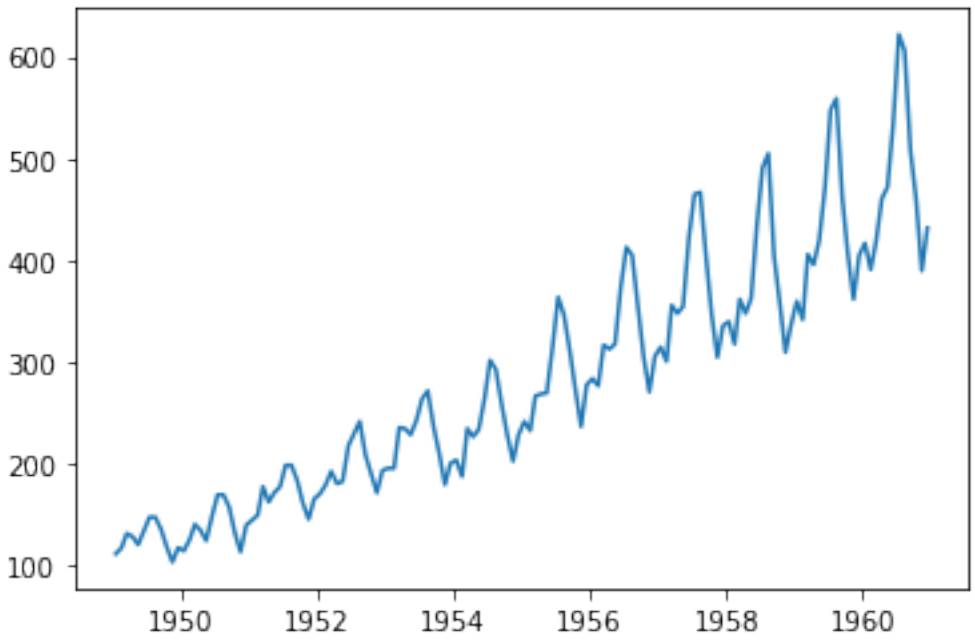
[4]:
```
#Setting the Month column as index. In TS the x axis or the independent␣
↪variable is Time
data.set_index('Month',inplace = True)
```

[5]:
```
data.head()
```

[5]:

Month	#Passengers
1949-01-15	112
1949-02-15	118
1949-03-15	132
1949-04-15	129
1949-05-15	121

[6]:
```
plt.plot(data);
```

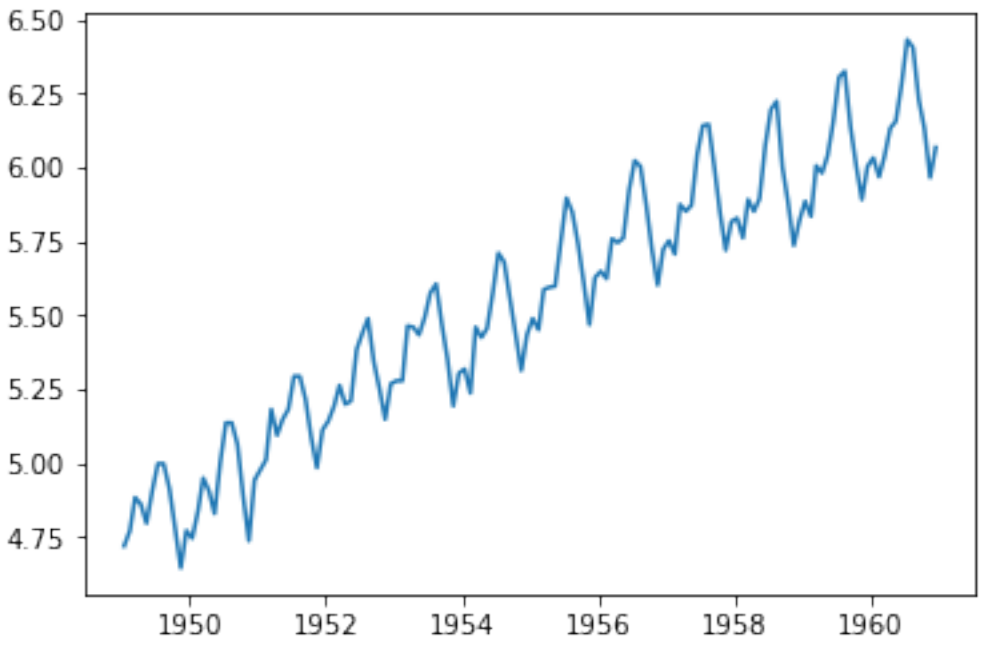


This is not a stationary time series as the mean is not constant. Let us try to make it stationary Apply transformation. Log transformation

[ ]:
```
ts_data = data['#Passengers']
```

[ ]:
```
ts_data_log = np.log(ts_data)
```

[ ]:
```
plt.plot(ts_data_log);
```

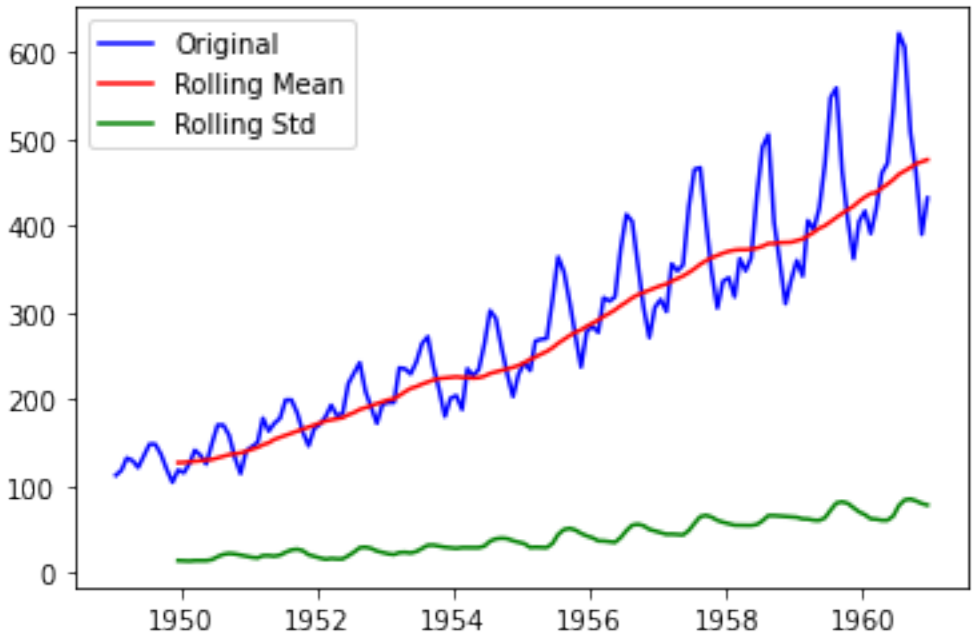


[ ]:
```
def test_stationarity(timeseries):

    #Calculate rolling mean and rolling standard deviation
    rolmean = timeseries.rolling(window = 12).mean()
    rolstd = timeseries.rolling(window = 12).std()

    original = plt.plot(timeseries, color = 'blue', label = 'Original')
    mean = plt.plot(rolmean, color = 'red', label = 'Rolling Mean')
    std = plt.plot(rolstd, color = 'green', label = 'Rolling Std')
    plt.legend(loc='best')


    print('Results of Dickey-Fuller Test :\n')
    df_test = adfuller(timeseries)
    my_output = pd.Series(df_test[:4], index =['Test Statistic', 'p-value',␣
↪'#Lags Used', 'Number of Observations Used'])
    for i,j in df_test[4].items():
        my_output['Critical Value (%s)'%i] = j
    print(my_output)
```

[18]:
```
test_stationarity(data['#Passengers'])
```

```
Results of Dickey-Fuller Test :

Test Statistic                  0.815369
p-value                         0.991880
```

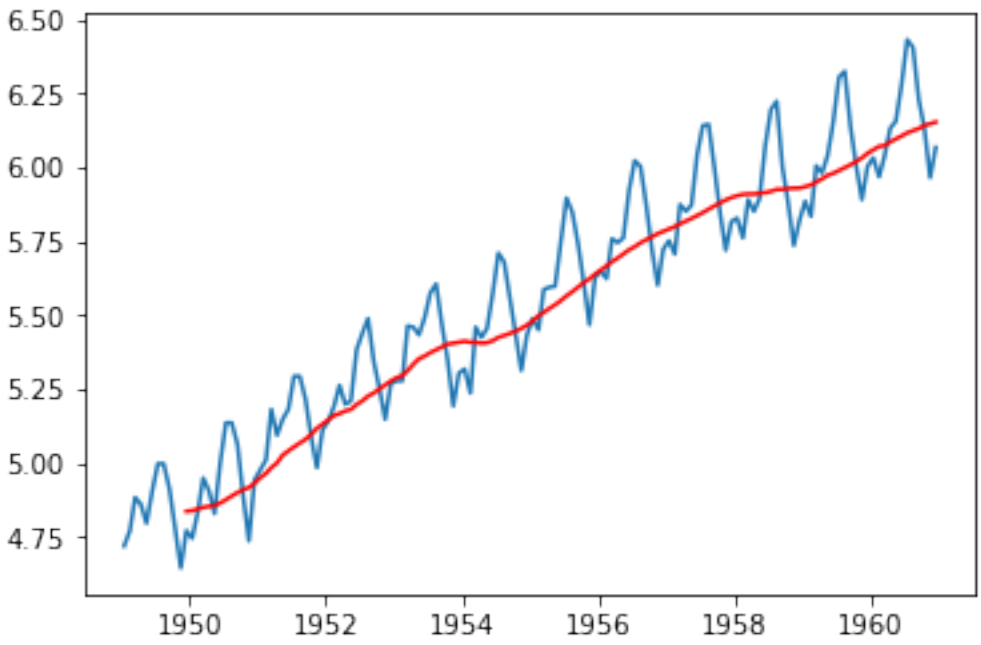
```
#Lags Used                          13.000000
Number of Observations Used        130.000000
Critical Value (1%)                 -3.481682
Critical Value (5%)                 -2.884042
Critical Value (10%)                -2.578770
dtype: float64
```



The Test Statistics is not less than any of Critical Value hence I have to accept the Null Hypothesis i.e. TS is not stationary

[12]:
```
movingAverage = ts_data_log.rolling(window = 12).mean()
movingSTD = ts_data_log.rolling(window = 12).std()
plt.plot(ts_data_log)
plt.plot(movingAverage,color='red')
```

[12]:
```
[<matplotlib.lines.Line2D at 0x7f0eebc803d0>]
```



Lets try differencing

[13]:
```
ts_data_log_diff = ts_data_log - movingAverage
ts_data_log_diff.head(15)
```

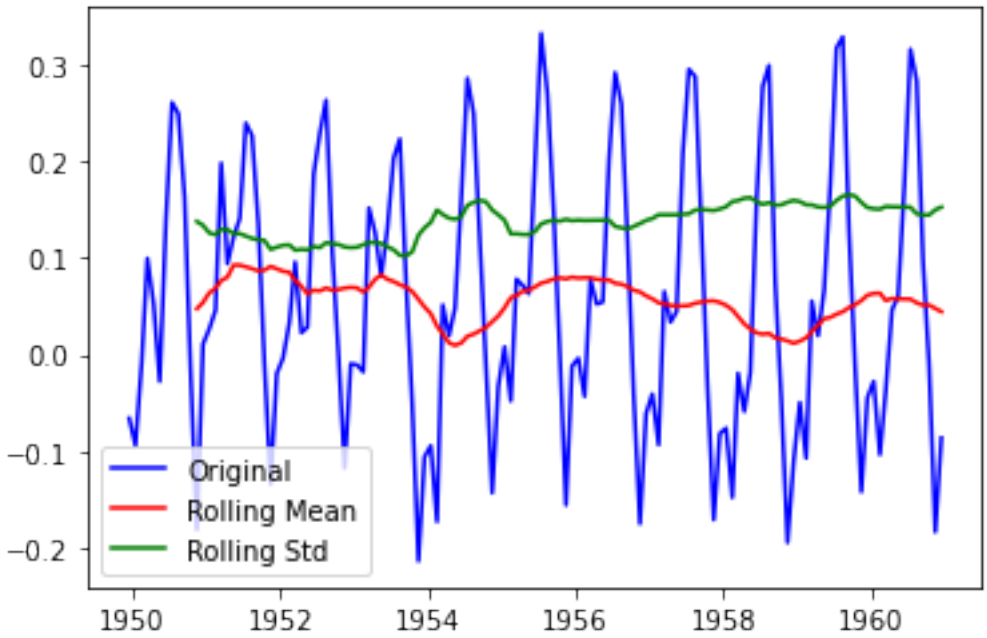
[13]:
```
Month
1949-01-15         NaN
1949-02-15         NaN
1949-03-15         NaN
1949-04-15         NaN
1949-05-15         NaN
1949-06-15         NaN
1949-07-15         NaN
1949-08-15         NaN
1949-09-15         NaN
1949-10-15         NaN
1949-11-15         NaN
1949-12-15   -0.065494
1950-01-15   -0.093449
1950-02-15   -0.007566
1950-03-15    0.099416
Name: #Passengers, dtype: float64
```

[14]:
```
ts_data_log_diff.dropna(inplace=True)
ts_data_log_diff.head()
```

[14]:
```
Month
1949-12-15   -0.065494
1950-01-15   -0.093449
1950-02-15   -0.007566
1950-03-15    0.099416
1950-04-15    0.052142
Name: #Passengers, dtype: float64
```

[15]:
```
test_stationarity(ts_data_log_diff)
```

```
Results of Dickey-Fuller Test :

Test Statistic                  -3.162908
p-value                          0.022235
#Lags Used                      13.000000
Number of Observations Used    119.000000
Critical Value (1%)             -3.486535
Critical Value (5%)             -2.886151
Critical Value (10%)            -2.579896
dtype: float64
```



[ ]: