AIM: Implementation of Time Series Analysis.

Tool : Jupyter Notebook

Roll No. 412039 ¶

In [1]:

```
import pandas as pd
df = pd.read_csv("Downloads/AirPassengers.csv")
```

In [2]:

```
df.head()
```

Out[2]:

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

In [4]:

```
df['Month'] = pd.to_datetime(df['Month'], format='%Y-%m')
df.index = df['Month']
del df['Month']
print(df.head())
```

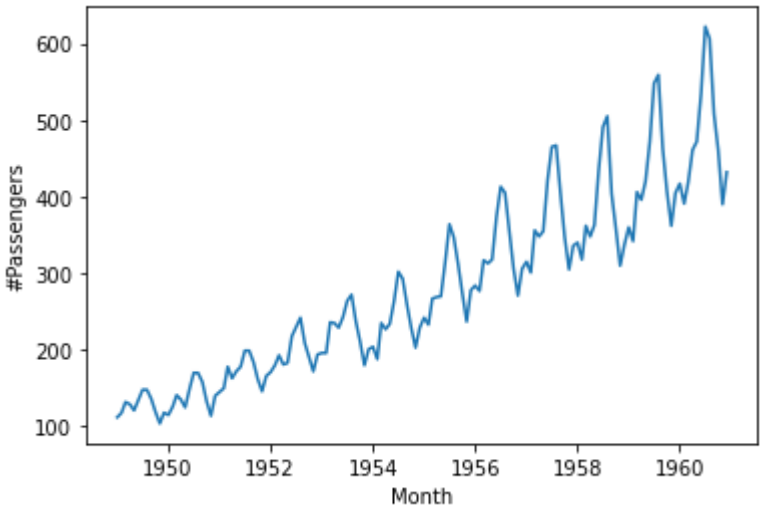
```
            #Passengers
Month
1949-01-01          112
1949-02-01          118
1949-03-01          132
1949-04-01          129
1949-05-01          121
```

In [19]:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.lineplot(x='Month', y='#Passengers', data=df)
```

Out[19]:

```
<AxesSubplot:xlabel='Month', ylabel='#Passengers'>
```



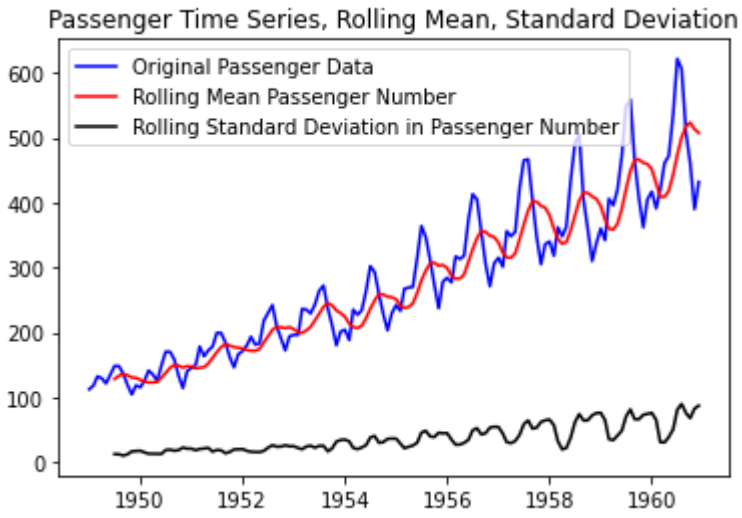
Stationarity Checking

In [20]:

```
rolling_mean = df.rolling(7).mean()
rolling_std = df.rolling(7).std()
plt.plot(df, color="blue",label="Original Passenger Data")
plt.plot(rolling_mean, color="red", label="Rolling Mean Passenger Number")
plt.plot(rolling_std, color="black", label = "Rolling Standard Deviation in Passenger Nu
plt.title("Passenger Time Series, Rolling Mean, Standard Deviation")
plt.legend(loc="best")
```

Out[20]:

```
<matplotlib.legend.Legend at 0x25ca247d430>
```



In [21]:

```
from statsmodels.tsa.stattools import adfuller
adft = adfuller(df,autolag="AIC")
output_df = pd.DataFrame({"Values":[adft[0],adft[1],adft[2],adft[3], adft[4]['1%'], adft
                                                    "critical value (1%)", "critical
print(output_df)
```

```
       Values                        Metric
0    0.815369                Test Statistics
1    0.991880                       p-value
2   13.000000               No. of lags used
3  130.000000  Number of observations used
4   -3.481682           critical value (1%)
5   -2.884042           critical value (5%)
6   -2.578770          critical value (10%)
```

Autocorrelation:

if our passenger data has strong autocorrelation, we can assume that high passenger numbers

today suggest a strong likelihood that they will be high tomorrow as well

In [22]:

```
autocorrelation_lag1 = df['#Passengers'].autocorr(lag=1)
print("One Month Lag: ", autocorrelation_lag1)
```

```
One Month Lag:  0.9601946480498522
```

Conclusion: Thus, we have see how the time series analysis is done using adfuller library for Stationarity and to find autocorrelation using autocorr function.