ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 2

Instructor: Luana Lima

Student Name: Yash Doshi

In [1]:
```python
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib as mpl
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_pacf
```

In [2]:
```python
eiats = pd.read_excel("Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sourc
                      sheet_name = "Monthly Data", parse_dates = ['Month'])
eiats
```

Out[2]:

	Month	Wood Energy Production (Trillion Btu)	Biofuels Production (Trillion Btu)	Total Biomass Energy Production (Trillion Btu)	Total Renewable Energy Production (Trillion Btu)	Hydroelectric Power Consumption (Trillion Btu)	Geothermal Energy Consumption (Trillion Btu)	Solar Energy Consumption (Trillion Btu)
0	1973-01-01	129.630	Not Available	129.787	403.981	272.703	1.491	Not Available
1	1973-02-01	117.194	Not Available	117.338	360.900	242.199	1.363	Not Available
2	1973-03-01	129.763	Not Available	129.938	400.161	268.810	1.412	Not Available
3	1973-04-01	125.462	Not Available	125.636	380.470	253.185	1.649	Not Available
4	1973-05-01	129.624	Not Available	129.834	392.141	260.770	1.537	Not Available
...	...	...	...	...	...	...	...	...
569	2020-06-01	180.782	164.111	377.859	1050.542	259.445	17.398	129.862
570	2020-07-01	185.357	180.789	401.014	1006.388	247.114	18.120	139.094
571	2020-08-01	188.216	179.379	402.983	965.785	215.725	18.078	128.03
572	2020-09-01	182.834	175.381	391.618	894.957	170.798	17.585	108.597
573	2020-10-01	186.346	184.232	406.115	949.990	163.392	17.659	100.881

574 rows × 14 columns

QUESTION 1

You will work only with the following columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption. Create a data frame structure with these three time series only. Use the command head() to verify your data.

In [3]:
```python
eiats = eiats.loc[:,['Month', 'Total Biomass Energy Production (Trillion Btu)', 'Total
                     'Hydroelectric Power Consumption (Trillion Btu)']]
eiats
```

Out[3]:

	Month	Total Biomass Energy Production (Trillion Btu)	Total Renewable Energy Production (Trillion Btu)	Hydroelectric Power Consumption (Trillion Btu)
0	1973-01-01	129.787	403.981	272.703
1	1973-02-01	117.338	360.900	242.199
2	1973-03-01	129.938	400.161	268.810
3	1973-04-01	125.636	380.470	253.185
4	1973-05-01	129.834	392.141	260.770
...	...	...	...	...
569	2020-06-01	377.859	1050.542	259.445
570	2020-07-01	401.014	1006.388	247.114
571	2020-08-01	402.983	965.785	215.725
572	2020-09-01	391.618	894.957	170.798
573	2020-10-01	406.115	949.990	163.392

574 rows × 4 columns

QUESTION 2

Transform your data frame in a time series object and specify the starting point and frequency of the time series using the function ts().

In [10]:
```python
eiats['Month'] = pd.to_datetime(eiats['Month'])
```

In [11]:
```python
eiats.dtypes
```

Out[11]:
```
Month                                              datetime64[ns]
Total Biomass Energy Production (Trillion Btu)            float64
Total Renewable Energy Production (Trillion Btu)          float64
Hydroelectric Power Consumption (Trillion Btu)            float64
dtype: object
```

QUESTION 3

Compute mean and standard deviation for these three series.

MEAN and STANDARD DEVIATION

Mean and standard deviation of Total Biomass Energy Production

Mean

In [ ]:
```python
a_mean = np.mean(eiats['Total Biomass Energy Production (Trillion Btu)'])
a_mean
```

Standard Deviation

In [ ]:
```python
a_stdev = np.std(eiats['Total Biomass Energy Production (Trillion Btu)'])
a_stdev
```

Mean and standard deviation of Total Renewable Energy Production

Mean

In [ ]:
```python
b_mean = np.mean(eiats['Total Renewable Energy Production (Trillion Btu)'])
b_mean
```

Standard Deviation

In [ ]:
```python
b_stdev = np.std(eiats['Total Renewable Energy Production (Trillion Btu)'])
b_stdev
```

Mean and standard deviation of Hydroelectric Power Consumption

Mean

In [ ]:
```python
c_mean = np.mean(eiats['Hydroelectric Power Consumption (Trillion Btu)'])
c_mean
```

Standard Deviation

In [ ]:
```python
c_stdev = np.std(eiats['Hydroelectric Power Consumption (Trillion Btu)'])
c_stdev
```

QUESTION 4

Display and interpret the time series plot for each of these variables. Try to make your plot as informative as possible by writing titles, labels, etc. For each plot add a horizontal line at the mean of each series in a different color.

In [ ]:
```python
plt.figure(figsize = (16,5))
plt.plot(eiats['Month'], eiats['Total Biomass Energy Production (Trillion Btu)'], colo
plt.xlabel('Time')
plt.ylabel('Total Biomass Energy Production (Trillion Btu)')
plt.title('Plot for Total Biomass Energy Production (Trillion Btu)')
plt.axhline(a_mean, color = 'midnightblue', linestyle = '--')
```

In [ ]:
```python
plt.figure(figsize = (16,5))
plt.plot(eiats['Month'], eiats['Total Renewable Energy Production (Trillion Btu)'], co
plt.title('Plot for Total Renewable Energy Production (Trillion Btu)')
plt.xlabel('Time')
plt.ylabel('Total Renewable Energy Production (Trillion Btu)')
plt.axhline(b_mean, color = 'midnightblue', linestyle = '--')
```

In [ ]:
```python
plt.figure(figsize=(16, 5))

plt.plot(eiats['Month'], eiats['Hydroelectric Power Consumption (Trillion Btu)'], colo
plt.xlabel('Time')
plt.ylabel('Hydroelectric Power Consumption (Trillion Btu)')
plt.title('Plot for Hydroelectric Power Consumption (Trillion Btu)')
plt.axhline(c_mean, color = 'midnightblue', linestyle = '--')
```

QUESTION 5

Compute the correlation between these three series. Are they significantly correlated? Ex- plain your answer.

Correlation between Total Biomass Energy Production and Total Renewable Energy Production

In [ ]:
```python
a_corr = np.corrcoef(eiats['Total Biomass Energy Production (Trillion Btu)'],
                     eiats['Total Renewable Energy Production (Trillion Btu)'])
a_corr
```

A correlation of 0.92 indicates that there is a strong, positive correlation between Total Biomass Energy Production and Total Renewable Energy Production. It means that both the variables move in the same direction together.

Correlation between Total Biomass Energy Production and Hydroelectric Power Consumption

In [ ]:
```python
b_corr = np.corrcoef(eiats['Total Biomass Energy Production (Trillion Btu)'],
                     eiats['Hydroelectric Power Consumption (Trillion Btu)'])
b_corr
```

A correlation of -0.255 indicates that there is a weak, negative correlation between Total Biomass Energy Production and Hydroelectric Power Consumption. It means that they do not move in the same direction together. This makes sense, because one variable is a measure of production , whereas, the other variable is a measure of consumption. Both these variables are of two completely different energies (one is of biomass, and the other is of hydroelectric).

Correlation between Total Renewable Energy Production and Hydroelectric Power Consumption

In [ ]:
```python
c_corr = np.corrcoef(eiats['Total Renewable Energy Production (Trillion Btu)'],
                     eiats['Hydroelectric Power Consumption (Trillion Btu)'])
c_corr
```

A correlation of -0.0027 indicates a very weak, negative correlation between Total Renewable Energy Production and Hydro- electric Power Consumption. It means that they do not move in the same direction together.

QUESTION 6

Compute the autocorrelation function from lag 1 up to lag 40 for these three variables. What can you say about these plots? Do the three of them have the same behavior?

In [ ]:
```python
ax = sm.graphics.tsa.plot_acf(eiats['Total Biomass Energy Production (Trillion Btu)'],
                              title = 'ACF of Total Biomass Energy Production (Trillion Btu)
```

ACF is a measure of dependence between two adjacent values of the same variables. In ACF, we talk about the same variable at different times. This plot tells us how the biomass production at a given time period is related to another time period. For instance, the correlation between first point at lag 1 and second point at lag 2 is 0.972.

Hence, ACF tells us how correlated the points are with each other, based on how many time steps they are separated by. It is how correlated past data points are to the future data points, for different values of time separation.

In [ ]:
```python
ax = sm.graphics.tsa.plot_acf(eiats['Total Renewable Energy Production (Trillion Btu)
                              title = 'ACF of Total Renewable Energy Production (Trillion Bt
```

It can be seen from this plot that there is not much difference in correlation between the total renewable energy production at two separate time periods. In this case, the correlation between first point at lag 1 and second point at lag 2 is 0.966.

If we compare the ACF of total renewable energy production with total biomass energy production, we find that the points at different time intervals are highly autocorrelated in case of biomass energy production than total renewable energy production.

In [ ]:
```python
ax = sm.graphics.tsa.plot_acf(eiats['Hydroelectric Power Consumption (Trillion Btu)'],
                              title = 'ACF of Hydroelectric Power Consumption (Trillion Btu)
```

This ACF plot shows that there is very little correlation between the two variables at different time periods. For instance, the correlation between first point at lag 1 and second point at lag 2 is 0.802, and that between the first point and the third point at lag 3 is 0.550.

It should also be noted that ACF for Hydroelectric Power Consumption shows a seasonality. That is the reason why the autocorrelation factor drops, increases, and then drops again.

The three graphs do not showcase the same behavior. The ACF for total biomass energy production and total renewable energy production show a similar trend. The only thing is that the correlation of total renewable energy production is slightly lower than the total biomass energy production. However, the graph for hydroelectric power consumption is completely different than the other two. The hydroelectric power consumption shows a seasonal trend. That is the reason why it is rising, dropping, and then rising again.

QUESTION 7

Compute the partial autocorrelation function from lag 1 to lag 40 for these three variables. How these plots differ from the ones in Q6?

In [ ]:
```python
ax = plot_pacf(eiats['Total Biomass Energy Production (Trillion Btu)'], lags=40,
       title = 'PACF of Total Biomass Energy Production (Trillion Btu)')
```

In [ ]:
```python
ax = sm.graphics.tsa.plot_pacf(eiats['Total Renewable Energy Production (Trillion Btu)
                               title = 'PACF of Total Renewable Energy Production (Trillion
```

In [ ]:
```python
ax = sm.graphics.tsa.plot_pacf(eiats['Hydroelectric Power Consumption (Trillion Btu)']
                               title = 'PACF of Hydroelectric Power Consumption (Trillion Bt
```

Partial autocorrelation talks about the correlation between two points separated by some time period. PACF does not take into consideration the correlation of the points in-between them. Unlike ACF, the values of PACF try to be as close to zero as possible. PACF is important in order to know how one point is related to some other point in a distant future without the intervening terms.

These plots vary considerably from the plots in Q6. The first plot shows a significant correlation between first and second point, followed by correlations that are not so significant. The pattern of PACF plot for total renewable energy production is similar to the total biomass energy production. The PACF plot for Hydroelectric Power Consumption is the same as its ACF plot.

There is not much we can discover from the PACF plot alone. In order to analyze the data in a better way, we need to use ACF plot too. Together, we can know a lot about the data.