Chapter 18: Demo Time Series với PMDARIMA

In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv("electric_production.csv",index_col=0)
data.head()
```

Out[2]:

	IPG2211A2N
DATE	
1939-01-01	3.3842
1939-02-01	3.4100
1939-03-01	3.4875
1939-04-01	3.5133
1939-05-01	3.5133

In [3]:
```
data.index = pd.to_datetime(data.index)
```

In [4]:
```
data.index
```

Out[4]:
```
DatetimeIndex(['1939-01-01', '1939-02-01', '1939-03-01', '1939-04-01',
               '1939-05-01', '1939-06-01', '1939-07-01', '1939-08-01',
               '1939-09-01', '1939-10-01',
               ...
               '2017-11-01', '2017-12-01', '2018-01-01', '2018-02-01',
               '2018-03-01', '2018-04-01', '2018-05-01', '2018-06-01',
               '2018-07-01', '2018-08-01'],
              dtype='datetime64[ns]', name='DATE', length=956, freq=None)
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 956 entries, 1939-01-01 to 2018-08-01
Data columns (total 1 columns):
IPG2211A2N    956 non-null float64
dtypes: float64(1)
memory usage: 14.9 KB
```

In [6]:
```
data.columns = ['Energy Production']
```

In [7]:
```
data.head()
```

Out[7]:

	Energy Production
DATE	
1939-01-01	3.3842
1939-02-01	3.4100
1939-03-01	3.4875
1939-04-01	3.5133
1939-05-01	3.5133

In [8]:
```
import matplotlib.pyplot as plt
```

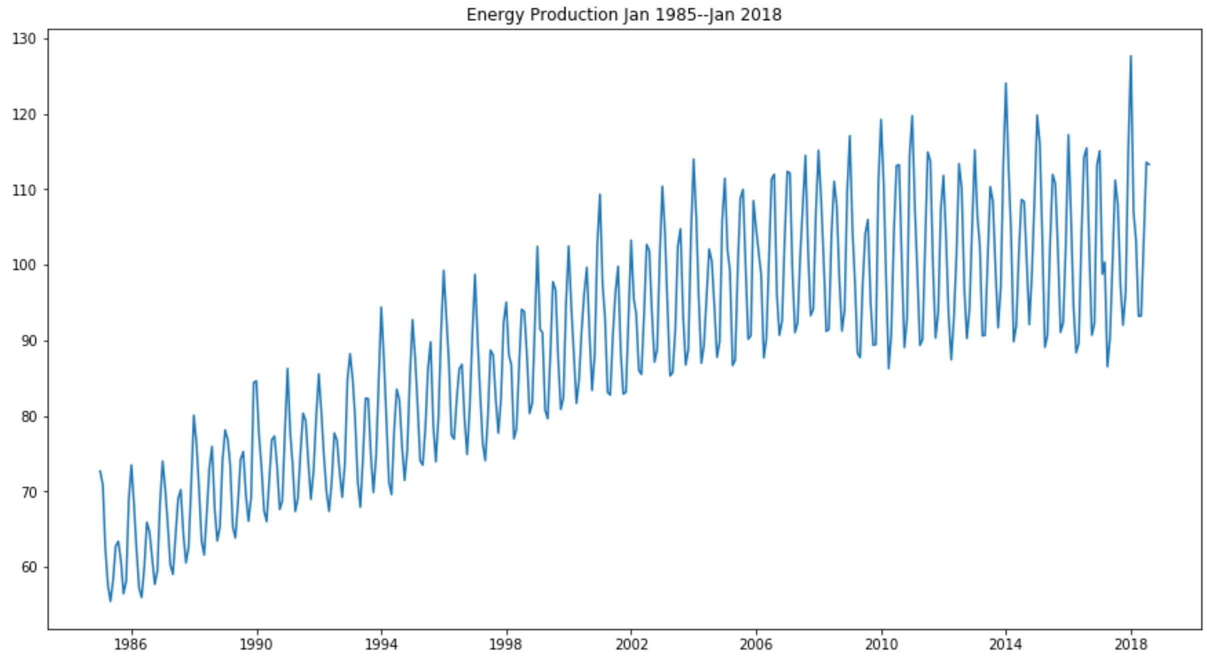
In [9]:
```
data_1985 = data[data.index.year >=int(1985)]
data_1985.head()
```

Out[9]:

	Energy Production
DATE	
1985-01-01	72.6803
1985-02-01	70.8479
1985-03-01	62.6166
1985-04-01	57.6106
1985-05-01	55.4467

In [10]:

```
plt.figure(figsize=(15,8))
plt.plot(data_1985)
plt.title("Energy Production Jan 1985--Jan 2018")
plt.show()
```



In [11]:

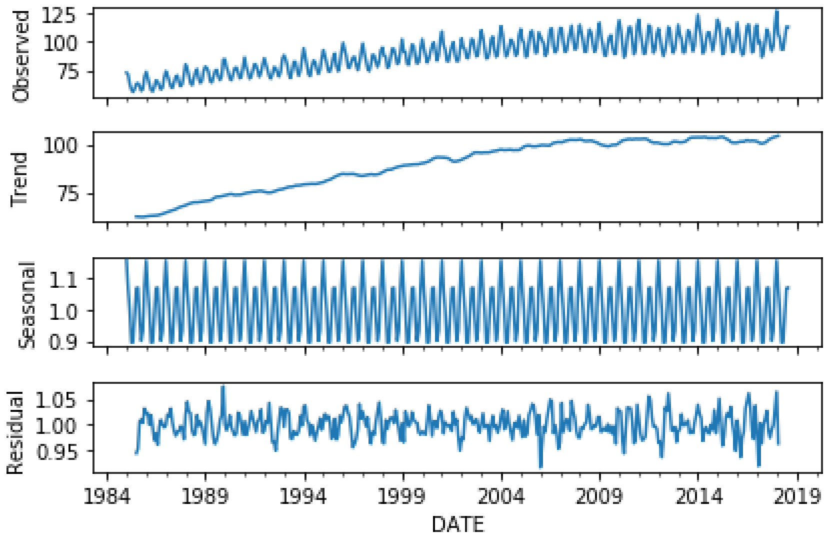
```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(data_1985, model='multiplicative')
result
```

Out[11]:

```
<statsmodels.tsa.seasonal.DecomposeResult at 0x1fb9276c2b0>
```

In [12]:

```
result.plot()
plt.show()
```



- Với kết quả trên, ta có thể thấy rõ tính seasonal component của data, và cũng có thể thấy xu hướng dữ liệu ở trên được tách riêng.

- Trend có thể lên hoặc xuống và có thể tuyến tính hoặc phi tuyến tính. Cần phải hiểu tập dữ liệu để biết liệu một khoảng thời gian đáng kể đã trôi qua có thể xác định xu hướng thực tế hay chưa.
- Cũng có thể có biến động bất thường (Irregular fluctuation) là những thay đổi đột ngột ngẫu nhiên và không thể đoán trước

Áp dụng auto_arima để xây dựng mô hình

Cài pip install pmdarima

In [13]:

```
from pmdarima.arima import auto_arima
```

In [14]:

```
stepwise_model = auto_arima(data, start_p=2, start_q=2,
                           max_p=5, max_q=5, m=12,
                           start_P=1, seasonal=True,
                           d=1, D=1, trace=True,
                           error_action='ignore',
                           suppress_warnings=True,
                           stepwise=True)
```

```
Fit ARIMA: order=(2, 1, 2) seasonal_order=(1, 1, 1, 12); AIC=3729.018, BIC=376
7.811, Fit time=12.485 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 1, 0, 12); AIC=4238.962, BIC=424
8.660, Fit time=0.102 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(1, 1, 0, 12); AIC=4058.517, BIC=407
7.914, Fit time=1.288 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(0, 1, 1, 12); AIC=3859.889, BIC=387
9.286, Fit time=2.816 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(0, 1, 1, 12); AIC=3729.631, BIC=376
3.574, Fit time=9.248 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(2, 1, 1, 12); AIC=3707.297, BIC=375
0.938, Fit time=40.735 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(2, 1, 0, 12); AIC=3776.649, BIC=381
5.442, Fit time=25.816 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(2, 1, 2, 12); AIC=3698.512, BIC=374
7.003, Fit time=47.233 seconds
Fit ARIMA: order=(1, 1, 2) seasonal_order=(2, 1, 2, 12); AIC=3699.508, BIC=374
3.149, Fit time=35.313 seconds
Fit ARIMA: order=(3, 1, 2) seasonal_order=(2, 1, 2, 12); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(2, 1, 2, 12); AIC=3701.329, BIC=374
4.970, Fit time=49.559 seconds
Fit ARIMA: order=(2, 1, 3) seasonal_order=(2, 1, 2, 12); AIC=3700.293, BIC=375
3.633, Fit time=116.732 seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(2, 1, 2, 12); AIC=3700.580, BIC=373
9.372, Fit time=93.359 seconds
Fit ARIMA: order=(3, 1, 3) seasonal_order=(2, 1, 2, 12); AIC=3698.976, BIC=375
7.165, Fit time=136.558 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(1, 1, 2, 12); AIC=3723.193, BIC=376
6.834, Fit time=84.143 seconds
Total fit time: 655.396 seconds
```

In [15]:

```
print(stepwise_model.aic())
```

```
3698.5121729262005
```

In [16]:

```
train = data.loc['1985-01-01':'2016-12-01']
test = data.loc['2015-01-01':]
```

In [17]:

```
test.head()
```

Out[17]:

	Energy Production
DATE	
2015-01-01	119.8260
2015-02-01	116.0253
2015-03-01	103.9265
2015-04-01	89.0847
2015-05-01	90.6408

In [18]:

```
len(test)
```

Out[18]:

```
44
```

Bước 2: Fit mô hình

In [19]:

```
stepwise_model.fit(train)
```

Out[19]:

```
ARIMA(callback=None, disp=0, maxiter=None, method=None, order=(2, 1, 2),
   out_of_sample_size=0, scoring='mse', scoring_args={},
   seasonal_order=(2, 1, 2, 12), solver='lbfgs', start_params=None,
   suppress_warnings=True, transparams=True, trend=None,
   with_intercept=True)
```

Bước 3: Dự đoán kết quả

In [20]:

```
future_forecast = stepwise_model.predict(n_periods=len(test))
```

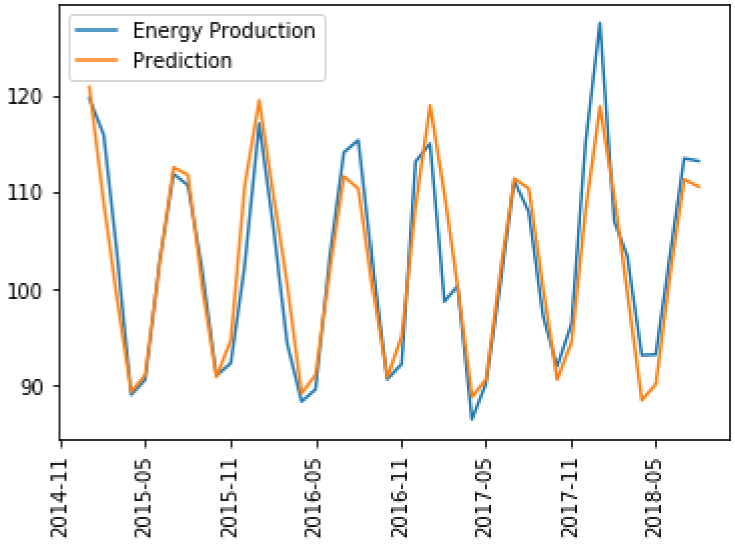
```
In [21]: future_forecast
```

```
Out[21]: array([121.02093743, 108.96541585,  99.25005531,  89.36263872,
                 91.15240549, 102.58023544, 112.64660083, 111.8612734 ,
                100.16103254,  90.92434851,  94.73861988, 110.74551708,
                119.63790847, 109.64710782, 100.63934302,  89.21472724,
                 91.14381236, 102.04387384, 111.72315674, 110.41734412,
                 99.83636462,  90.89462894,  95.18341575, 108.79151397,
                119.09329406, 109.77761568, 100.34402343,  88.85838068,
                 90.65731515, 101.87039198, 111.48665134, 110.42950596,
                100.05382187,  90.64954818,  94.59006641, 108.17197239,
                118.98115851, 109.48192681,  99.59711228,  88.49917059,
                 90.17854912, 101.71774161, 111.40038319, 110.62959389])
```

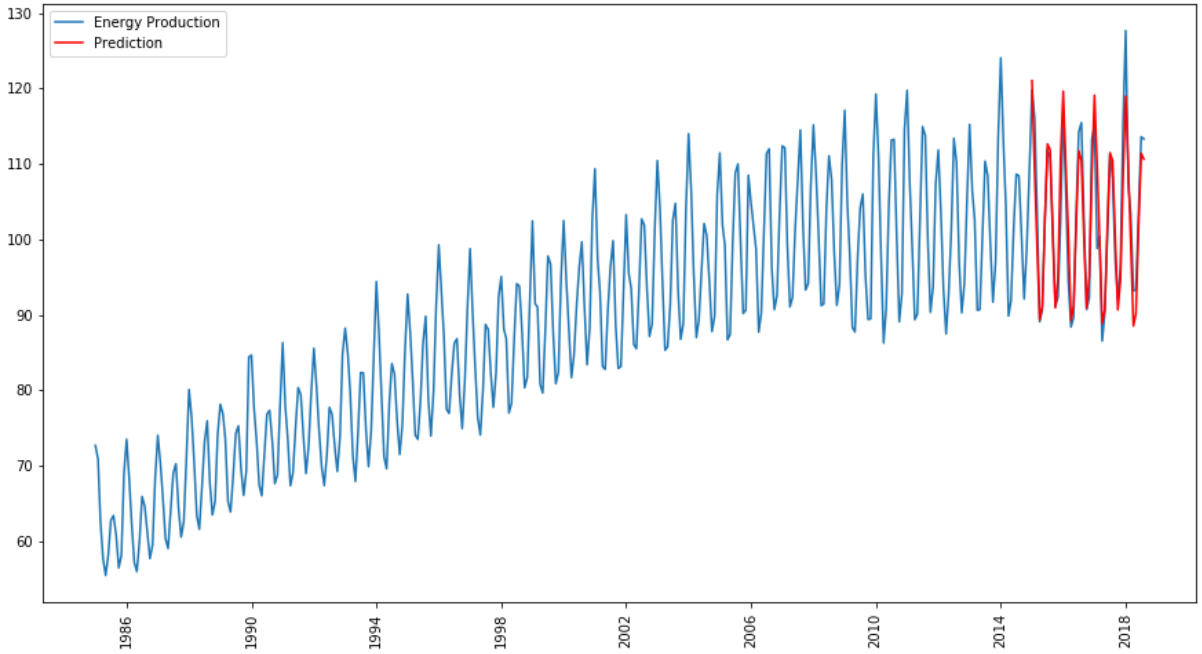
```
In [22]: future_forecast = pd.DataFrame(future_forecast,index = test.index,columns=['Predic
```

Bước 4: Trực quan hóa dữ liệu

```
In [23]: plt.plot(test, label='Energy Production')
         plt.plot(future_forecast, label='Prediction')
         plt.xticks(rotation='vertical')
         plt.legend()
         plt.show()
```



In [24]:
```
plt.figure(figsize=(15,8))
plt.plot(data_1985, label='Energy Production')
plt.plot(future_forecast, label='Prediction', color='red')
plt.xticks(rotation='vertical')
plt.legend()
plt.show()
```



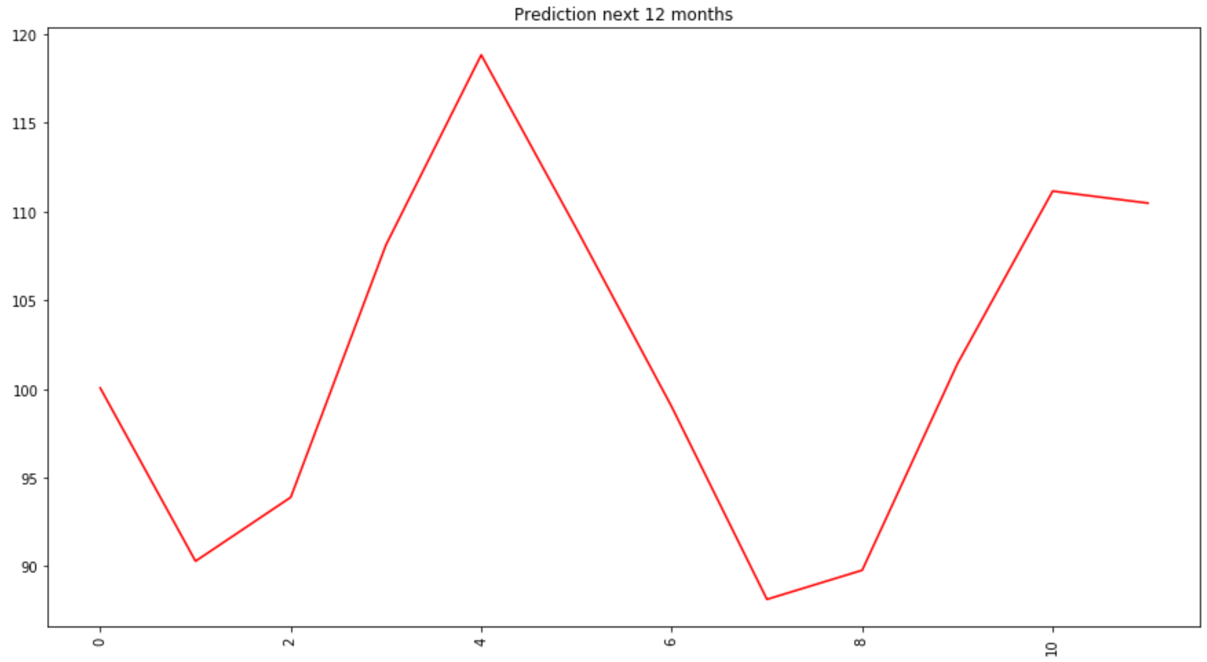
Dự đoán 12 tháng tiếp theo

In [25]:
```
future_forecast = stepwise_model.predict(n_periods=len(test)+12)
future_forecast
```

Out[25]:
```
array([121.02093743, 108.96541585,  99.25005531,  89.36263872,
         91.15240549, 102.58023544, 112.64660083, 111.8612734 ,
        100.16103254,  90.92434851,  94.73861988, 110.74551708,
        119.63790847, 109.64710782, 100.63934302,  89.21472724,
         91.14381236, 102.04387384, 111.72315674, 110.41734412,
         99.83636462,  90.89462894,  95.18341575, 108.79151397,
        119.09329406, 109.77761568, 100.34402343,  88.85838068,
         90.65731515, 101.87039198, 111.48665134, 110.42950596,
        100.05382187,  90.64954818,  94.59006641, 108.17197239,
        118.98115851, 109.48192681,  99.59711228,  88.49917059,
         90.17854912, 101.71774161, 111.40038319, 110.62959389,
        100.05827356,  90.29611652,  93.89578844, 108.12580355,
        118.83579264, 109.03648227,  99.00111142,  88.14316308,
         89.78421857, 101.42313235, 111.15783398, 110.48135914])
```

In [26]:

```
plt.figure(figsize=(15,8))
plt.plot(future_forecast[len(test):], color='red')
plt.xticks(rotation='vertical')
plt.title("Prediction next 12 months")
plt.show()
```



Prediction next 12 months

In [ ]: