Chapter 18 - exercise 2: Python Programming Language Interest

Cho dữ liệu interest Python Programming Language 5 năm tính đến 18/08/2019 trong tập tin sales-of-shampoo-over-a-three-year.csv.

- Thực hiện việc dự báo interest sử dụng thuật toán ARIMA
- Cho biết trong 3 tháng sau 5 năm trên thì interest như thế nào?

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

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

Out[2]:

Week	Interest
8/31/2014	49
9/7/2014	63
9/14/2014	52
9/21/2014	60
9/28/2014	77

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

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

Out[4]:
```
DatetimeIndex(['2014-08-31', '2014-09-07', '2014-09-14', '2014-09-21',
               '2014-09-28', '2014-10-05', '2014-10-12', '2014-10-19',
               '2014-10-26', '2014-11-02',
               ...
               '2019-06-16', '2019-06-23', '2019-06-30', '2019-07-07',
               '2019-07-14', '2019-07-21', '2019-07-28', '2019-08-04',
               '2019-08-11', '2019-08-18'],
              dtype='datetime64[ns]', name='Week', length=260, freq=None)
```

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

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 260 entries, 2014-08-31 to 2019-08-18
Data columns (total 1 columns):
Interest    260 non-null int64
dtypes: int64(1)
memory usage: 4.1 KB
```

In [6]:
```
data.head()
```

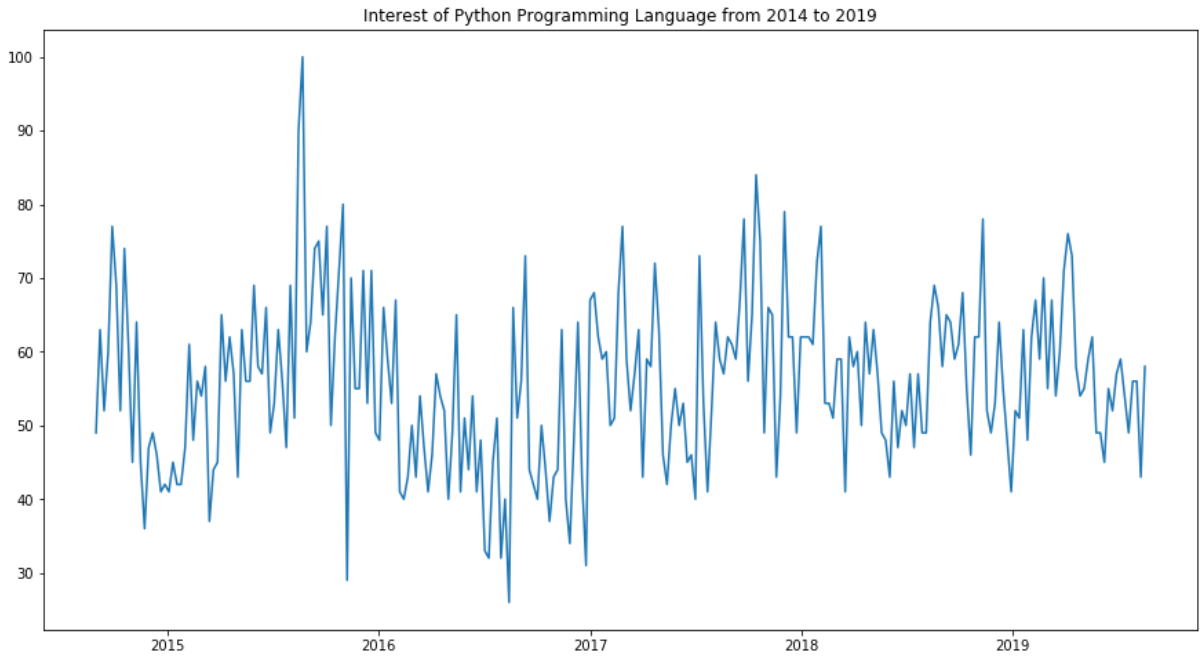
Out[6]:

Week	Interest
2014-08-31	49
2014-09-07	63
2014-09-14	52
2014-09-21	60
2014-09-28	77

In [7]:
```
from datetime import datetime
```

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

In [9]:
```
plt.figure(figsize=(15,8))
plt.plot(data)
plt.title("Interest of Python Programming Language from 2014 to 2019")
plt.show()
```



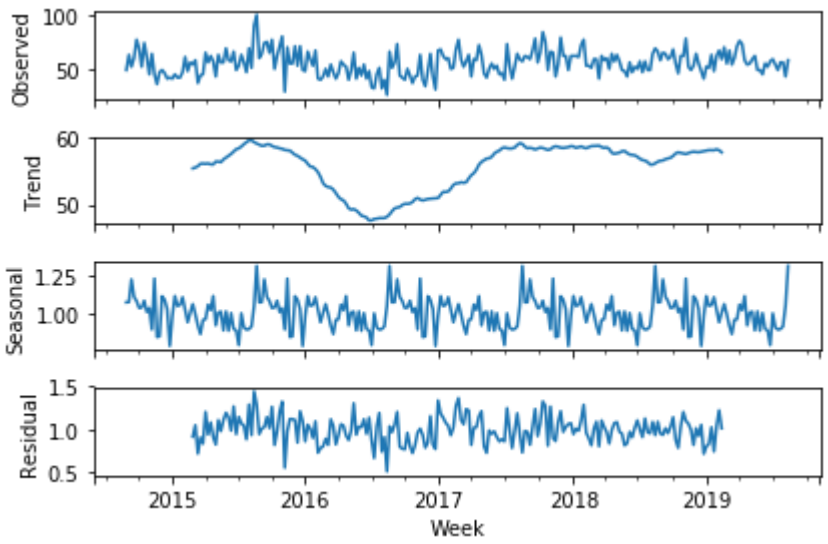
In [10]:
```
type(data)
```

Out[10]:
```
pandas.core.frame.DataFrame
```

In [11]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(x = data, model='multiplicative')
result
```

Out[11]:
```
<statsmodels.tsa.seasonal.DecomposeResult at 0x17f91d339b0>
```

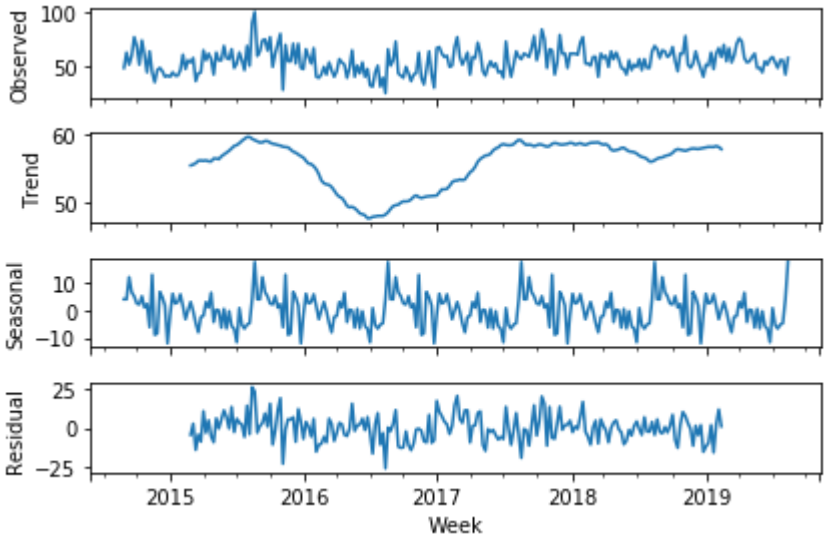
In [12]:
```
result.plot()
plt.show()
```



In [13]:
```
result1 = seasonal_decompose(x = data, model='additive')
result1
```

Out[13]:
```
<statsmodels.tsa.seasonal.DecomposeResult at 0x17f9991d320>
```

In [14]:
```
result1.plot()
plt.show()
```



In [15]:
```
from pmdarima.arima import auto_arima
```

In [16]:
```
stepwise_model = auto_arima(data, start_p=2, start_q=2,
                            max_p=5, max_q=5, m=52,
                            seasonal=True,
                            start_P=0,
                            d=1, D=1, trace=True,
                            error_action='ignore',
                            suppress_warnings=True,
                            stepwise=True)
```

```
Fit ARIMA: order=(2, 1, 2) seasonal_order=(0, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 1, 0, 52); AIC=1813.367, BIC=182
0.032, Fit time=1.093 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(1, 1, 0, 52); AIC=1723.788, BIC=173
7.119, Fit time=36.176 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(0, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(0, 1, 0, 52); AIC=1777.768, BIC=178
7.766, Fit time=3.372 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(2, 1, 0, 52); AIC=1690.942, BIC=170
7.605, Fit time=95.446 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(2, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(2, 1, 0, 52); AIC=1734.619, BIC=174
7.950, Fit time=73.219 seconds
Fit ARIMA: order=(2, 1, 0) seasonal_order=(2, 1, 0, 52); AIC=1665.637, BIC=168
5.633, Fit time=154.932 seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(2, 1, 0, 52); AIC=1636.882, BIC=166
0.211, Fit time=136.295 seconds
Fit ARIMA: order=(3, 1, 2) seasonal_order=(2, 1, 0, 52); AIC=1640.059, BIC=167
0.053, Fit time=151.330 seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(1, 1, 0, 52); AIC=1660.795, BIC=168
0.792, Fit time=36.532 seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(2, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(2, 1, 0, 52); AIC=1636.521, BIC=165
6.517, Fit time=99.780 seconds
Fit ARIMA: order=(1, 1, 2) seasonal_order=(2, 1, 0, 52); AIC=1638.245, BIC=166
1.574, Fit time=121.729 seconds
Fit ARIMA: order=(2, 1, 2) seasonal_order=(2, 1, 0, 52); AIC=1638.725, BIC=166
5.387, Fit time=149.722 seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(1, 1, 0, 52); AIC=1662.282, BIC=167
8.946, Fit time=23.260 seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(2, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(2, 1, 0, 52); AIC=1634.582, BIC=165
1.245, Fit time=67.594 seconds
Fit ARIMA: order=(0, 1, 2) seasonal_order=(2, 1, 0, 52); AIC=1636.505, BIC=165
6.501, Fit time=150.010 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(1, 1, 0, 52); AIC=1660.322, BIC=167
3.653, Fit time=17.888 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(2, 1, 1, 52); AIC=nan, BIC=nan, Fit
time=nan seconds
Total fit time: 1318.529 seconds
```

In [17]:
```
print(stepwise_model.aic())
```

```
1634.5815934160546
```

In [18]:
```
data.head()
```

Out[18]:

Week	Interest
2014-08-31	49
2014-09-07	63
2014-09-14	52
2014-09-21	60
2014-09-28	77

In [19]:
```
train = data.loc['2014-08-31':'2018-06-01']
test = data.loc['2018-06-01':]
```

In [20]:
```
train.shape
```

Out[20]:
```
(196, 1)
```

In [21]:
```
test.shape
```

Out[21]:
```
(64, 1)
```

In [22]:
```
stepwise_model.fit(train)
```

Out[22]:
```
ARIMA(callback=None, disp=0, maxiter=None, method=None, order=(0, 1, 1),
   out_of_sample_size=0, scoring='mse', scoring_args={},
   seasonal_order=(2, 1, 0, 52), solver='lbfgs', start_params=None,
   suppress_warnings=True, transparams=True, trend=None,
   with_intercept=True)
```

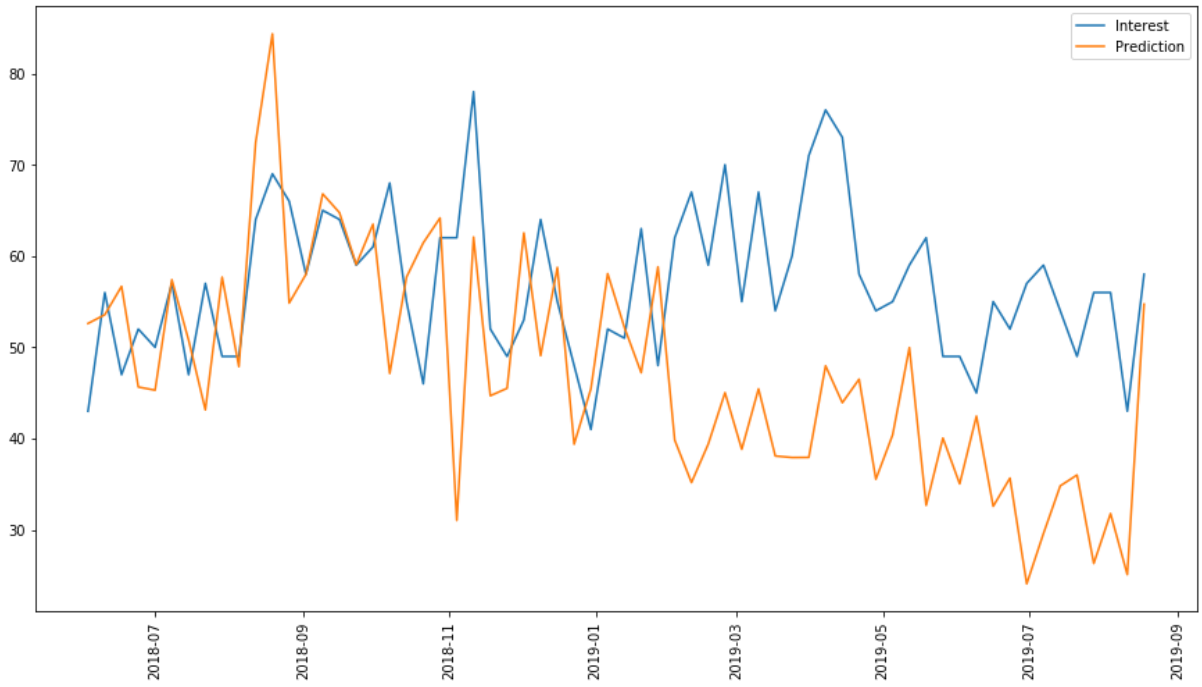
In [23]:
```
future_forecast = stepwise_model.predict(n_periods=len(test))
```

In [24]:
```
future_forecast
```

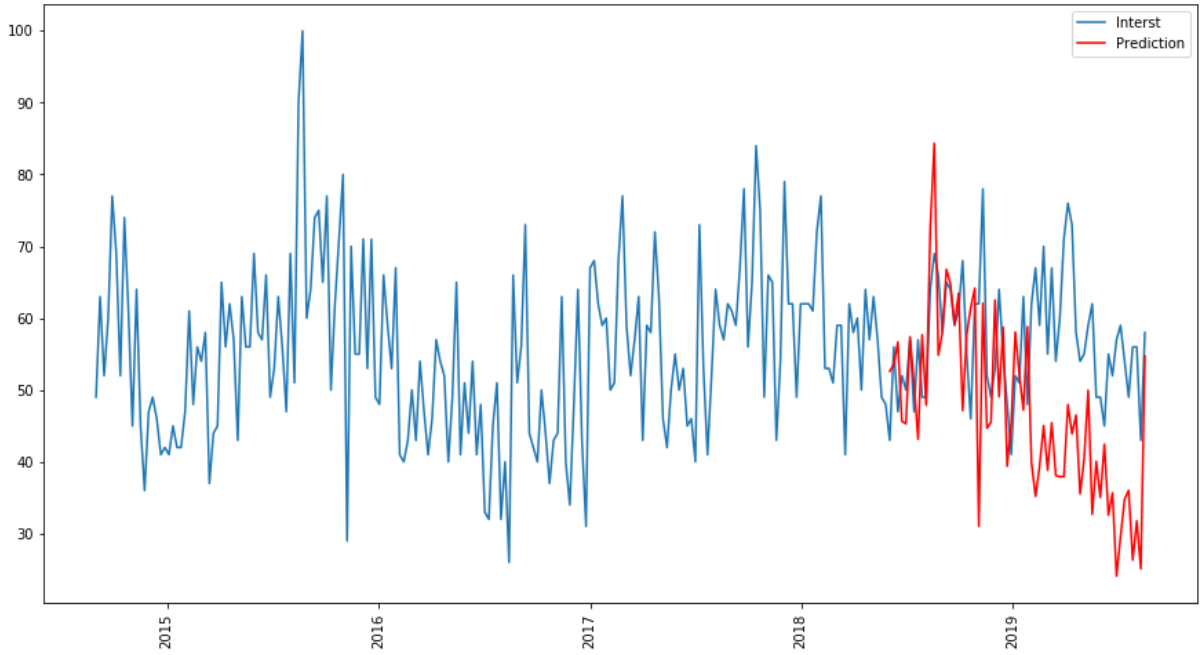
Out[24]:
```
array([52.61120811, 53.56503092, 56.6751102 , 45.65166489, 45.30394271,
       57.40314836, 50.79469654, 43.14465564, 57.68501617, 47.88404659,
       72.44128797, 84.33617947, 54.84413715, 57.9916966 , 66.80348799,
       64.74994477, 59.07032919, 63.48408602, 47.13483261, 57.68986335,
       61.44136461, 64.16678256, 31.0399268 , 62.07838094, 44.6937799 ,
       45.50884464, 62.53437068, 49.08269857, 58.75109973, 39.38519974,
       45.4529162 , 58.07036963, 52.17230915, 47.21661433, 58.81323416,
       39.83465129, 35.18468846, 39.40059972, 45.03074698, 38.82703259,
       45.45049323, 38.09538878, 37.92915456, 37.94001039, 47.97230114,
       43.92817842, 46.50662045, 35.55092562, 40.40492366, 49.96877358,
       32.6919809 , 40.05806333, 35.04505512, 42.46125126, 32.58309522,
       35.67125636, 24.0918668 , 29.6107319 , 34.82145952, 36.01332123,
       26.32514382, 31.79001677, 25.10390924, 54.72439835])
```

In [25]:
```
future_forecast = pd.DataFrame(future_forecast,
                               index = test.index,
                               columns=['Prediction'])
```

In [26]:
```
plt.figure(figsize=(15,8))
plt.plot(test, label='Interest')
plt.plot(future_forecast, label='Prediction')
plt.xticks(rotation='vertical')
plt.legend()
plt.show()
```



In [27]:
```
plt.figure(figsize=(15,8))
plt.plot(data, label='Interst')
plt.plot(future_forecast, label='Prediction', color='red')
plt.xticks(rotation='vertical')
plt.legend()
plt.show()
```

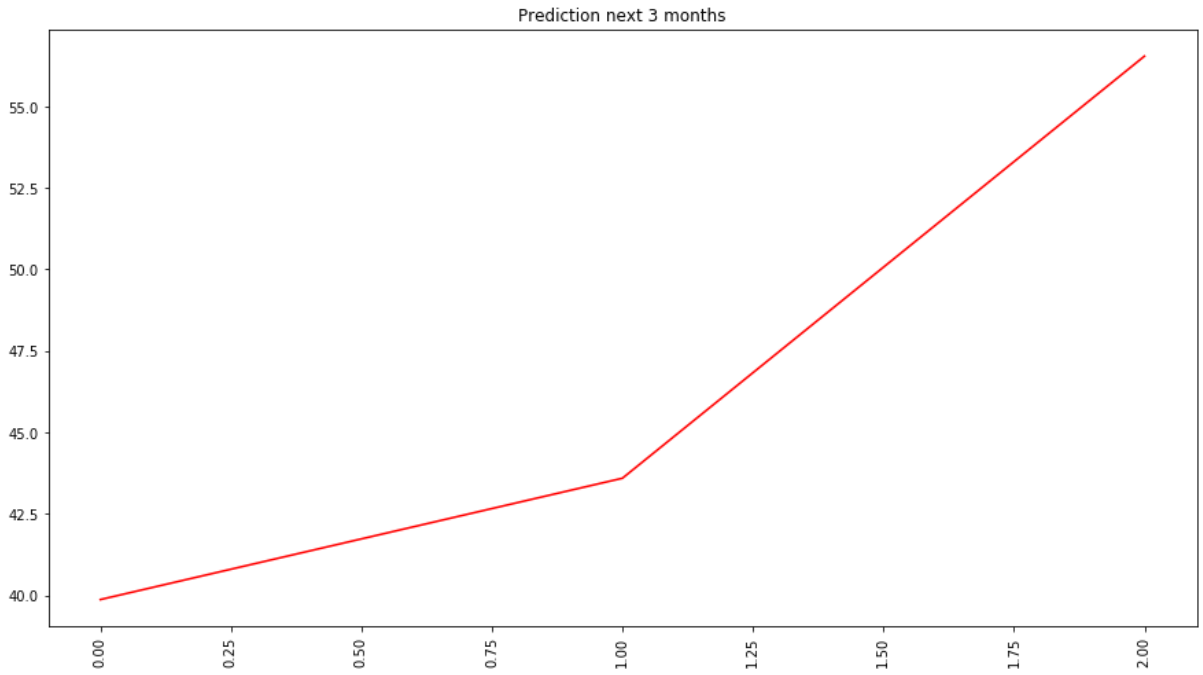


In [28]:
```
# Dự đoán 3 tháng sau
```

In [29]:
```
future_forecast = stepwise_model.predict(n_periods=len(test)+3)
future_forecast
```

Out[29]:
```
array([52.61120811, 53.56503092, 56.6751102 , 45.65166489, 45.30394271,
       57.40314836, 50.79469654, 43.14465564, 57.68501617, 47.88404659,
       72.44128797, 84.33617947, 54.84413715, 57.9916966 , 66.80348799,
       64.74994477, 59.07032919, 63.48408602, 47.13483261, 57.68986335,
       61.44136461, 64.16678256, 31.0399268 , 62.07838094, 44.6937799 ,
       45.50884464, 62.53437068, 49.08269857, 58.75109973, 39.38519974,
       45.4529162 , 58.07036963, 52.17230915, 47.21661433, 58.81323416,
       39.83465129, 35.18468846, 39.40059972, 45.03074698, 38.82703259,
       45.45049323, 38.09538878, 37.92915456, 37.94001039, 47.97230114,
       43.92817842, 46.50662045, 35.55092562, 40.40492366, 49.96877358,
       32.6919809 , 40.05806333, 35.04505512, 42.46125126, 32.58309522,
       35.67125636, 24.0918668 , 29.6107319 , 34.82145952, 36.01332123,
       26.32514382, 31.79001677, 25.10390924, 54.72439835, 39.87294506,
       43.59479699, 56.54215221])
```

In [30]:
```
plt.figure(figsize=(15,8))
plt.plot(future_forecast[len(test):], color='red')
plt.xticks(rotation='vertical')
plt.title("Prediction next 3 months")
plt.show()
```



In [ ]:
```
```