Chapter 18 - Exercise 1: Sales of shampoo over a three year

Cho dữ liệu bán shampoo 3 năm trong tập tin sales-of-shampoo-over-a-three-year.csv.

- Thực hiện việc dự báo bán sản phẩm shampoo sử dụng thuật toán ARIMA
- Cho biết trong 3 tháng sau 3 năm trên thì giá trị bán sản phẩm như thế nào?

In [ ]:

```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

```
Mounted at /content/gdrive
```

In [ ]:

```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter18_ARIMA/'
```

```
/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter18_ARIMA
```

In [ ]:

```
import pandas as pd
```

In [ ]:

```
data = pd.read_csv("sales-of-shampoo-over-a-three-year.csv", index_col=0)
data.head()
```

Out[4]:

	Sales of shampoo over a three year period
Month	
Friday, January 1, 2016	266.0
Monday, February 1, 2016	145.9
Tuesday, March 1, 2016	183.1
Friday, April 1, 2016	119.3
Sunday, May 1, 2016	180.3

In [ ]:

```
data.index = pd.to_datetime(data.index)
```

In [ ]:

```
data.index
```

Out[6]:

```
DatetimeIndex(['2016-01-01', '2016-02-01', '2016-03-01', '2016-04-01',
               '2016-05-01', '2016-06-01', '2016-07-01', '2016-08-01',
               '2016-09-01', '2016-10-01', '2016-11-01', '2016-12-01',
               '2017-01-01', '2017-02-01', '2017-03-01', '2017-04-01',
               '2017-05-01', '2017-06-01', '2017-07-01', '2017-08-01',
               '2017-09-01', '2017-10-01', '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',
               '2018-09-01', '2018-10-01', '2018-11-01', '2018-12-01'],
              dtype='datetime64[ns]', name='Month', freq=None)
```

In [ ]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 36 entries, 2016-01-01 to 2018-12-01
Data columns (total 1 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   Sales of shampoo over a three year period  36 non-null     float64
dtypes: float64(1)
memory usage: 576.0 bytes
```

In [ ]:

```
data.columns = ['Sales_of_shampoo']
```

In [ ]:

```
data.head()
```

Out[9]:

	Sales_of_shampoo
Month	
2016-01-01	266.0
2016-02-01	145.9
2016-03-01	183.1
2016-04-01	119.3
2016-05-01	180.3

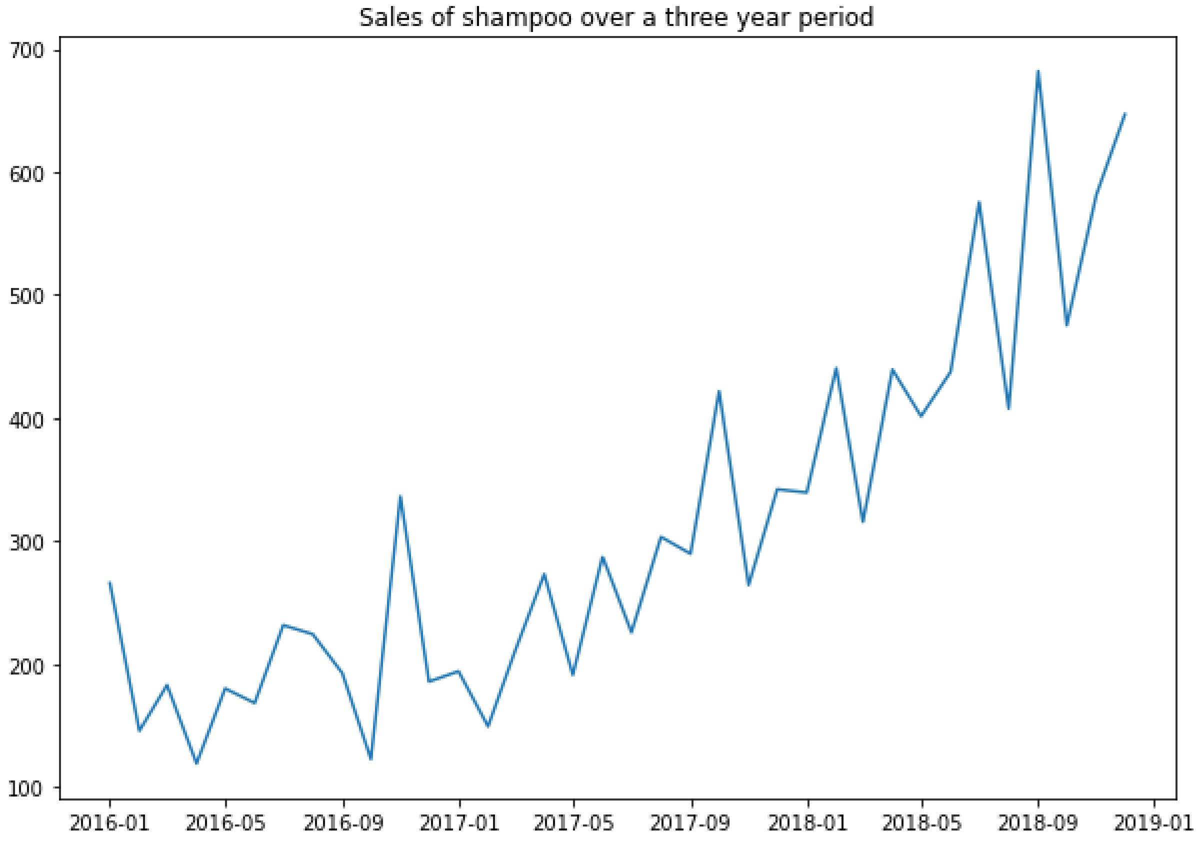
In [ ]:

```
from datetime import datetime
```

In [ ]:

```
import matplotlib.pyplot as plt
```

```
In [ ]: plt.figure(figsize=(10,7))
        plt.plot(data)
        plt.title("Sales of shampoo over a three year period")
        plt.show()
```



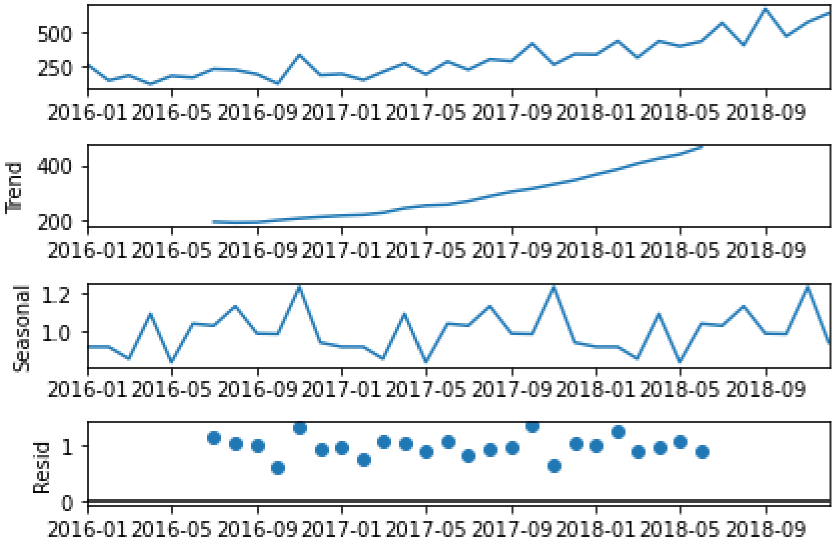
In [ ]:
```
type(data)
```

Out[13]:
```
pandas.core.frame.DataFrame
```

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

Out[14]:
```
<statsmodels.tsa.seasonal.DecomposeResult at 0x7f17f5b72eb8>
```

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



In [ ]:
```
! pip install pmdarima
```

In [ ]:
```
from pmdarima import auto_arima
```

In [ ]:

```
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)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(1,1,1)[12]             : AIC=inf, Time=1.32 sec
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=305.954, Time=0.02 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=287.610, Time=0.16 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=289.951, Time=0.28 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=287.696, Time=0.04 sec
 ARIMA(1,1,0)(2,1,0)[12]             : AIC=inf, Time=1.40 sec
 ARIMA(1,1,0)(1,1,1)[12]             : AIC=inf, Time=0.67 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=287.722, Time=0.25 sec
 ARIMA(1,1,0)(2,1,1)[12]             : AIC=291.042, Time=1.73 sec
 ARIMA(0,1,0)(1,1,0)[12]             : AIC=303.591, Time=0.11 sec
 ARIMA(2,1,0)(1,1,0)[12]             : AIC=286.460, Time=0.27 sec
 ARIMA(2,1,0)(0,1,0)[12]             : AIC=288.501, Time=0.07 sec
 ARIMA(2,1,0)(2,1,0)[12]             : AIC=inf, Time=1.84 sec
 ARIMA(2,1,0)(1,1,1)[12]             : AIC=inf, Time=0.89 sec
 ARIMA(2,1,0)(0,1,1)[12]             : AIC=287.268, Time=0.36 sec
 ARIMA(2,1,0)(2,1,1)[12]             : AIC=288.971, Time=2.39 sec
 ARIMA(3,1,0)(1,1,0)[12]             : AIC=287.153, Time=0.33 sec
 ARIMA(2,1,1)(1,1,0)[12]             : AIC=287.237, Time=0.39 sec
 ARIMA(1,1,1)(1,1,0)[12]             : AIC=285.464, Time=0.24 sec
 ARIMA(1,1,1)(0,1,0)[12]             : AIC=286.063, Time=0.08 sec
 ARIMA(1,1,1)(2,1,0)[12]             : AIC=286.473, Time=1.61 sec
 ARIMA(1,1,1)(1,1,1)[12]             : AIC=inf, Time=0.88 sec
 ARIMA(1,1,1)(0,1,1)[12]             : AIC=285.779, Time=0.35 sec
 ARIMA(1,1,1)(2,1,1)[12]             : AIC=288.479, Time=1.90 sec
 ARIMA(0,1,1)(1,1,0)[12]             : AIC=289.967, Time=0.19 sec
 ARIMA(1,1,2)(1,1,0)[12]             : AIC=inf, Time=0.48 sec
 ARIMA(0,1,2)(1,1,0)[12]             : AIC=inf, Time=0.48 sec
 ARIMA(2,1,2)(1,1,0)[12]             : AIC=inf, Time=0.72 sec
 ARIMA(1,1,1)(1,1,0)[12] intercept   : AIC=inf, Time=0.40 sec

Best model:  ARIMA(1,1,1)(1,1,0)[12]
Total fit time: 19.888 seconds
```

In [ ]:

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

```
285.4635906667674
```

In [ ]:

```
train = data.loc['2016-01-01':'2018-02-01']
test = data.loc['2018-02-01':]
```

```
In [ ]: test
```

Out[21]:

	Sales_of_shampoo
Month	
2018-02-01	440.4
2018-03-01	315.9
2018-04-01	439.3
2018-05-01	401.3
2018-06-01	437.4
2018-07-01	575.5
2018-08-01	407.6
2018-09-01	682.0
2018-10-01	475.3
2018-11-01	581.3
2018-12-01	646.9

```
In [ ]: len(test)

Out[22]: 11
```

```
In [ ]: len(train)

Out[23]: 26
```

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

Out[24]: ARIMA(maxiter=50, method='lbfgs', order=(1, 1, 1), out_of_sample_size=0,
               scoring='mse', scoring_args={}, seasonal_order=(1, 1, 0, 12),
               start_params=None, suppress_warnings=True, trend=None,
               with_intercept=False)
```

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

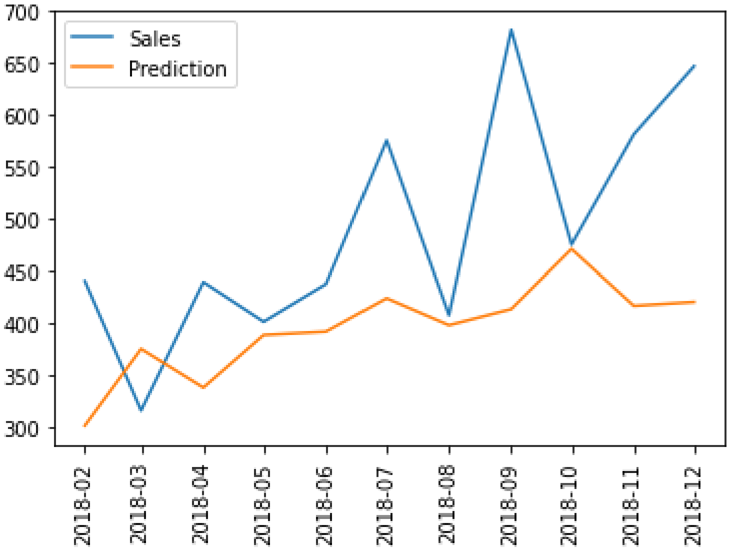
```
In [ ]: future_forecast

Out[26]: array([301.37822619, 375.19264882, 338.01219278, 388.5995918 ,
                391.89687524, 423.77554979, 398.01232103, 413.41907471,
                471.17193988, 416.54027228, 420.25738927])
```

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

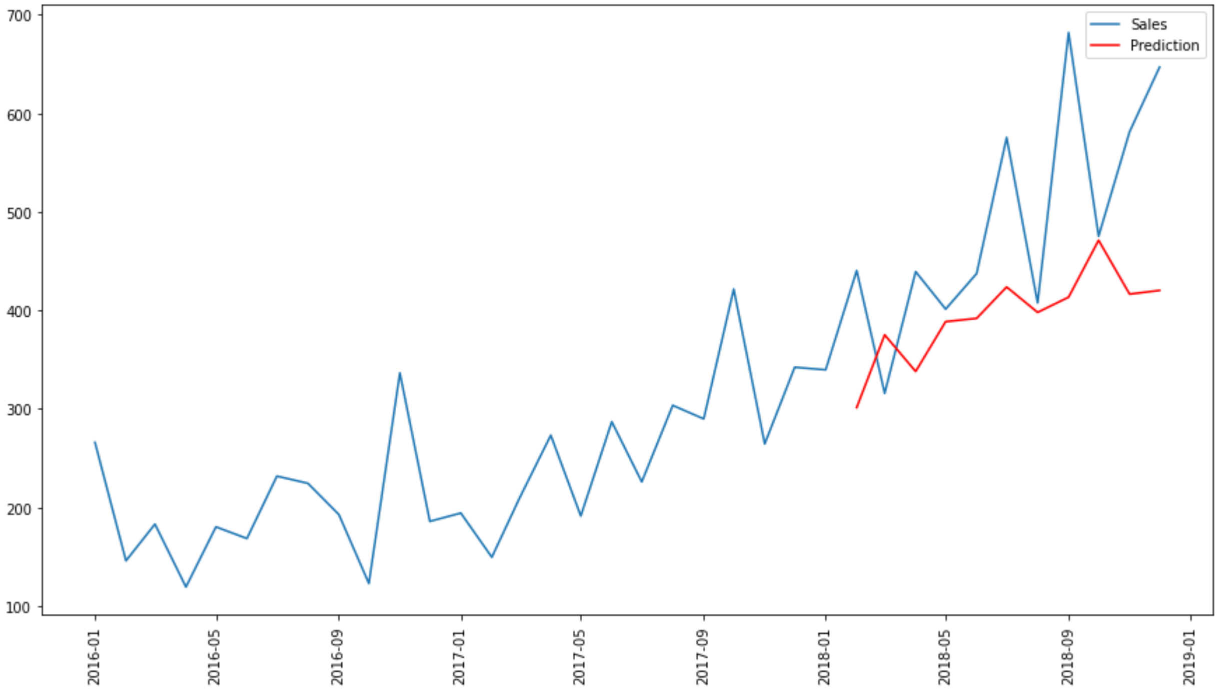
In [ ]:

```
plt.plot(test, label='Sales')
plt.plot(future_forecast, label='Prediction')
plt.xticks(rotation='vertical')
plt.legend()
plt.show()
```



In [ ]:

```
plt.figure(figsize=(15,8))
plt.plot(data, label='Sales')
plt.plot(future_forecast, label='Prediction', color='red')
plt.xticks(rotation='vertical')
plt.legend()
plt.show()
```



In [ ]:

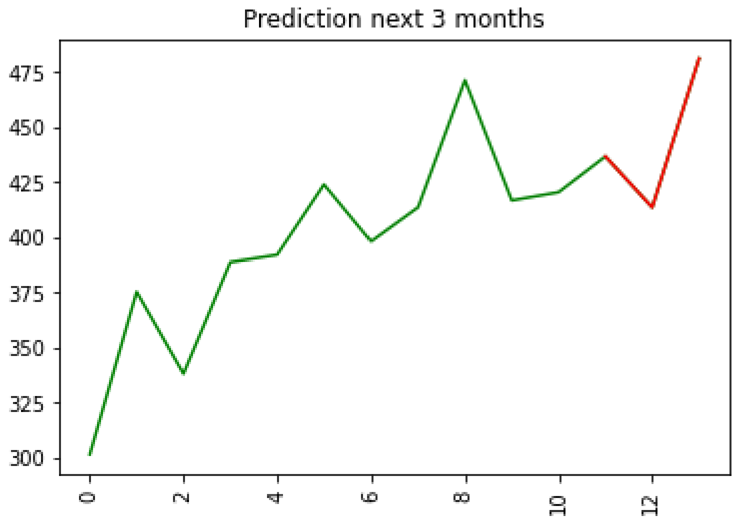
```
# Dự đoán 3 tháng sau
```

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

```
Out[31]: array([301.37822619, 375.19264882, 338.01219278, 388.5995918 ,
                391.89687524, 423.77554979, 398.01232103, 413.41907471,
                471.17193988, 416.54027228, 420.25738927, 436.52817491,
                413.36922593, 481.02851979])
```

```
In [ ]: import numpy as np
```

```
In [ ]: plt.plot(np.arange(14), future_forecast, color='green')
        plt.plot(np.array([11, 12, 13]),future_forecast[len(test):], color='red')
        plt.xticks(rotation='vertical')
        plt.title("Prediction next 3 months")
        plt.show()
```



```
In [ ]:
```