# Chapter 19 - Exercise 2: Monthly Candy production

- Cho dữ liệu candy_production.csv. Áp dụng mô hình HoltWinters để dự báo lượng sản phẩm kẹo cho 12 tháng tiếp theo

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice_2020/Chapter19_Holt
```

```
Mounted at /content/gdrive
/content/gdrive/My Drive/LDS6_MachineLearning/practice_2020/Chapter19_HoltWinte
r
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: Future
Warning: pandas.util.testing is deprecated. Use the functions in the public API
at pandas.testing instead.
  import pandas.util.testing as tm
```

## Đọc dữ liệu, kiểm tra/định dạng thời gian

In [3]:
```
df = pd.read_csv('candy_production.csv',
                 parse_dates=['observation_date'],
                 index_col='observation_date')
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 548 entries, 1972-01-01 to 2017-08-01
Data columns (total 1 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   IPG3113N  548 non-null    float64
dtypes: float64(1)
memory usage: 8.6 KB
```

In [5]:
```
# freq='H', 'D', 'W', 'M', 'MS': Hour, Day, Week, Month, Calendar month begin
df.index.freq = 'MS'
```

In [6]:
```
df.columns = ['candy_production']
```

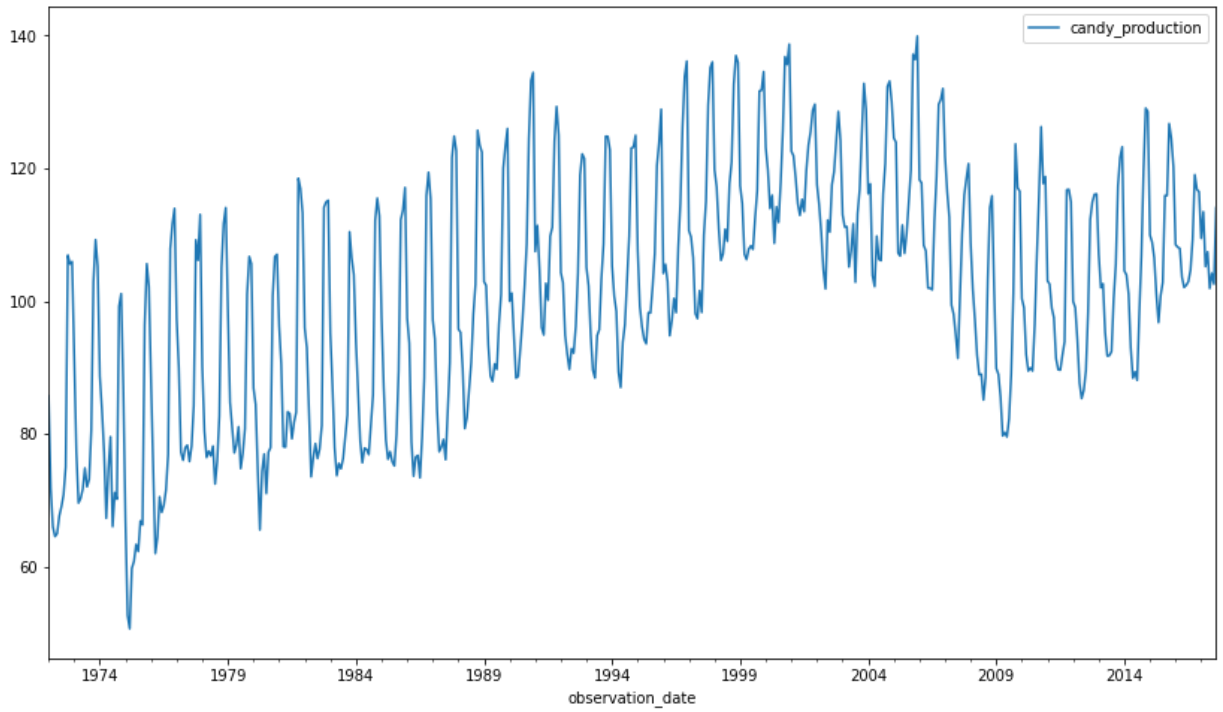
In [7]:
```
df.head()
```

Out[7]:

| observation_date | candy_production |
|---|---|
| **1972-01-01** | 85.6945 |
| **1972-02-01** | 71.8200 |
| **1972-03-01** | 66.0229 |
| **1972-04-01** | 64.5645 |
| **1972-05-01** | 65.0100 |

In [8]:
```
df.plot(figsize=(14,8))
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0880ebee48>
```

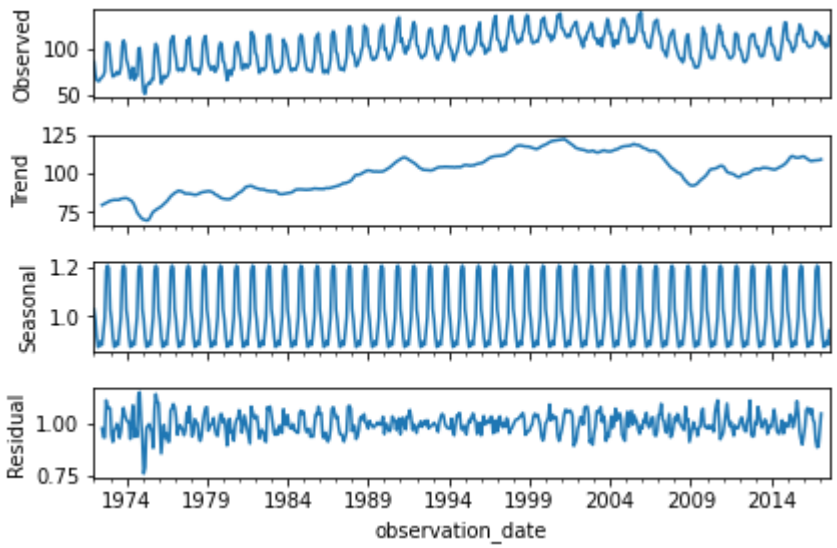



# Decomposition

In [9]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(df, model='multiplicative')
result
```

Out[9]:
```
<statsmodels.tsa.seasonal.DecomposeResult at 0x7f0880d78748>
```

In [10]:
```
result.plot()
plt.show()
```




# Chia dữ liệu train/test => Áp dụng mô hình

In [11]:
```
train, test = df.iloc[:500, 0], df.iloc[500:, 0]
```

In [12]:
```
train[0:5]
```

Out[12]:
```
observation_date
1972-01-01    85.6945
1972-02-01    71.8200
1972-03-01    66.0229
1972-04-01    64.5645
1972-05-01    65.0100
Freq: MS, Name: candy_production, dtype: float64
```

In [13]:
```
test[0:5]
```

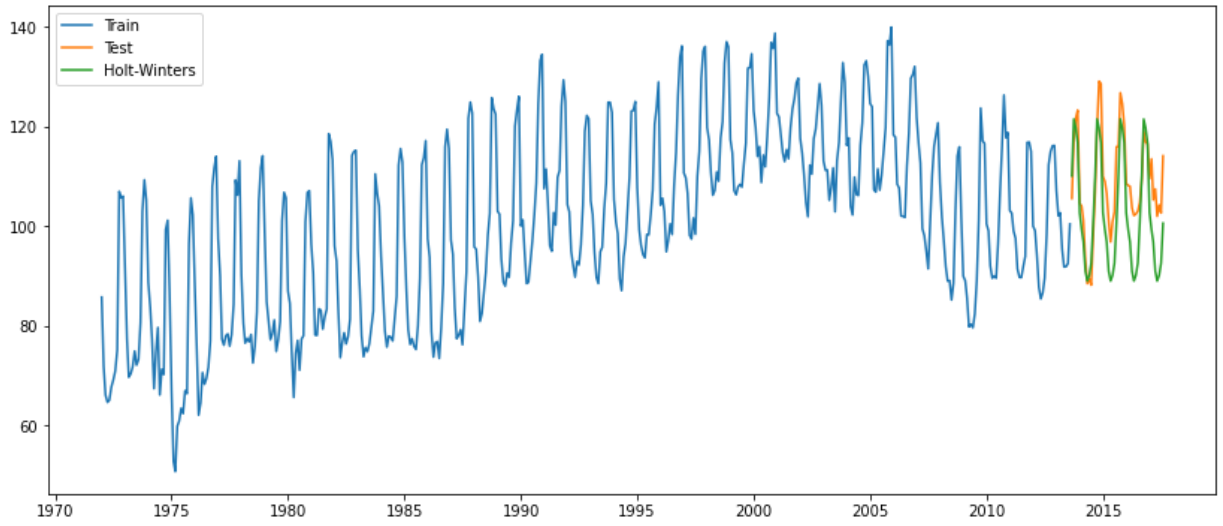
Out[13]:
```
observation_date
2013-09-01    105.5167
2013-10-01    117.3458
2013-11-01    121.6179
2013-12-01    123.2412
2014-01-01    104.5665
Freq: MS, Name: candy_production, dtype: float64
```

- tsa (https://www.statsmodels.org/stable/generated/statsmodels.tsa.holtwinters.ExponentialSmoothir
- tsa es (https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.

In [14]:
```
model = ExponentialSmoothing(train, seasonal='mul',
                             seasonal_periods=12).fit()
pred = model.predict(start=test.index[0], end=test.index[-1])
```

In [15]:
```
plt.figure(figsize=(14,6))
plt.plot(train.index, train, label='Train')
plt.plot(test.index, test, label='Test')
plt.plot(pred.index, pred, label='Holt-Winters')
plt.legend(loc='best')
```

Out[15]:
```
<matplotlib.legend.Legend at 0x7f087b3f8278>
```




# Dự đoán

In [16]:
```
df.tail()
```

Out[16]:

| observation_date | candy_production |
|---|---|
| **2017-04-01** | 107.4288 |
| **2017-05-01** | 101.9209 |
| **2017-06-01** | 104.2022 |
| **2017-07-01** | 102.5861 |
| **2017-08-01** | 114.0613 |

In [17]:
```
import datetime
s = datetime.datetime(2017, 9, 1)
e = datetime.datetime(2018, 8, 1)
pred_next_12_month = model.predict(start= s, end=e)
pred_next_12_month
```

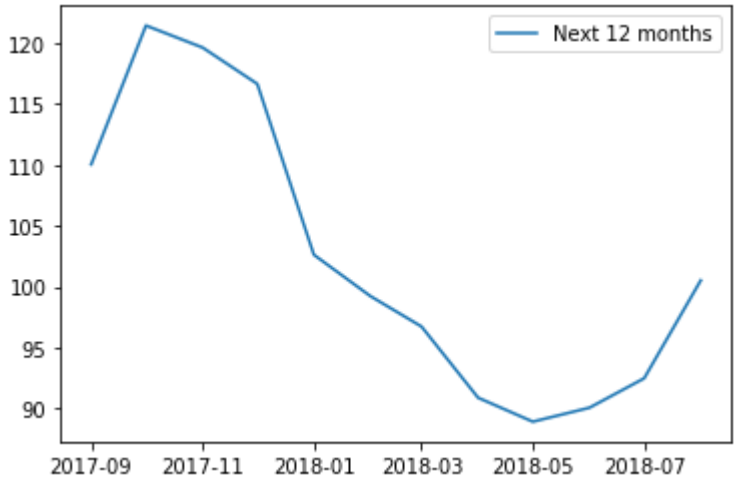
Out[17]:
```
2017-09-01    110.049089
2017-10-01    121.441542
2017-11-01    119.639014
2017-12-01    116.654449
2018-01-01    102.637514
2018-02-01     99.240327
2018-03-01     96.744683
2018-04-01     90.905232
2018-05-01     88.940178
2018-06-01     90.084152
2018-07-01     92.498123
2018-08-01    100.534045
Freq: MS, dtype: float64
```

In [18]:
```
x = pd.Series(pred_next_12_month)
type(x)
```

Out[18]:
```
pandas.core.series.Series
```

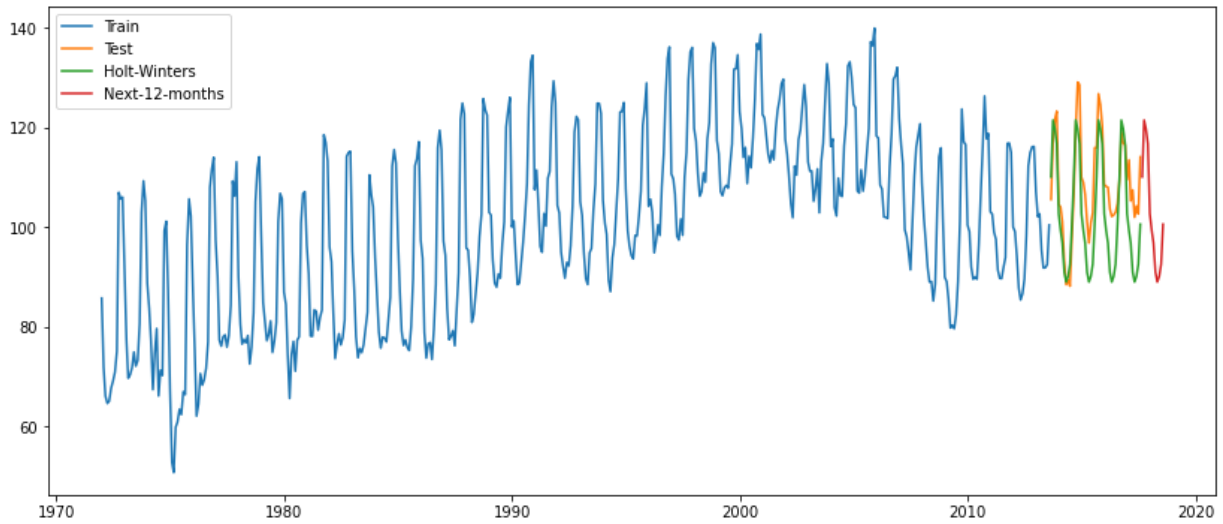
In [19]:
```
plt.plot(x.index, x.values, label='Next 12 months')
plt.legend(loc='best')
```

Out[19]:
```
<matplotlib.legend.Legend at 0x7f087b3d3080>
```




# Trực quan hóa dữ liệu

In [20]:
```
plt.figure(figsize=(14,6))
plt.plot(train.index, train, label='Train')
plt.plot(test.index, test, label='Test')
plt.plot(pred.index, pred, label='Holt-Winters')
plt.plot(x.index, x.values, label='Next-12-months')
plt.legend(loc='best')
plt.show()
```




In [20]: