01-ETS-Decomposition

October 19, 2022

For more information, visit us at www.pieriandata.com

0.1 ETS

0.2 Error/Trend/Seasonality Models

As we begin working with endogenous data ("endog" for short) and start to develop forecasting models, it helps to identify and isolate factors working within the system that influence behavior. Here the name "endogenous" considers internal factors, while "exogenous" would relate to external forces. These fall under the category of state space models, and include decomposition (described below), and exponential smoothing (described in an upcoming section).

The decomposition of a time series attempts to isolate individual components such as error, trend, and seasonality (ETS). We've already seen a simplistic example of this in the Introduction to Statsmodels section with the Hodrick-Prescott filter. There we separated data into a trendline and a cyclical feature that mapped observed data back to the trend.

Related Function:

statsmodels.tsa.seasonal.seasonal_decompose(x, model) Seasonal decomposition using moving averages

For Further Reading:

Forecasting: Principles and Practice Innovations state space models for exponential smoothing Wikipedia Decomposition of time series

0.3 Seasonal Decomposition

Statsmodels provides a seasonal decomposition tool we can use to separate out the different components. This lets us see quickly and visually what each component contributes to the overall behavior.

We apply an additive model when it seems that the trend is more linear and the seasonality and trend components seem to be constant over time (e.g. every year we add 10,000 passengers). A multiplicative model is more appropriate when we are increasing (or decreasing) at a non-linear rate (e.g. each year we double the amount of passengers).

For these examples we'll use the International Airline Passengers dataset, which gives monthly totals in thousands from January 1949 to December 1960.

[1]:
```
import pandas as pd
import numpy as np
%matplotlib inline
```

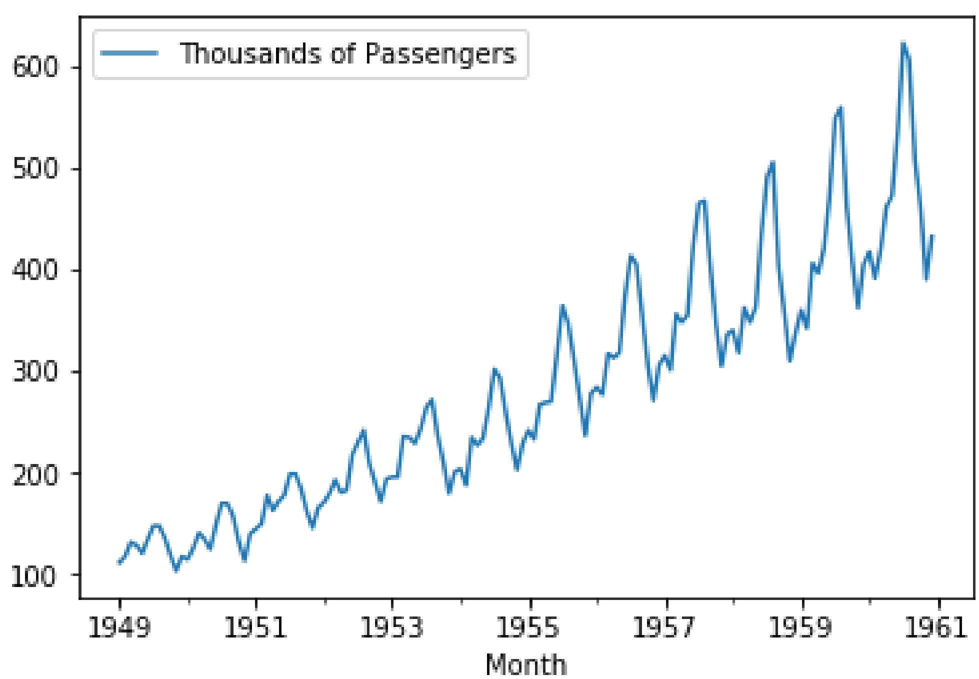
[2]:
```
airline = pd.read_csv('../Data/airline_passengers.
↪csv',index_col='Month',parse_dates=True)
```

[3]:
```
airline.dropna(inplace=True)
```

[4]:
```
airline.head()
```

[4]:
```
            Thousands of Passengers
Month
1949-01-01                      112
1949-02-01                      118
1949-03-01                      132
1949-04-01                      129
1949-05-01                      121
```

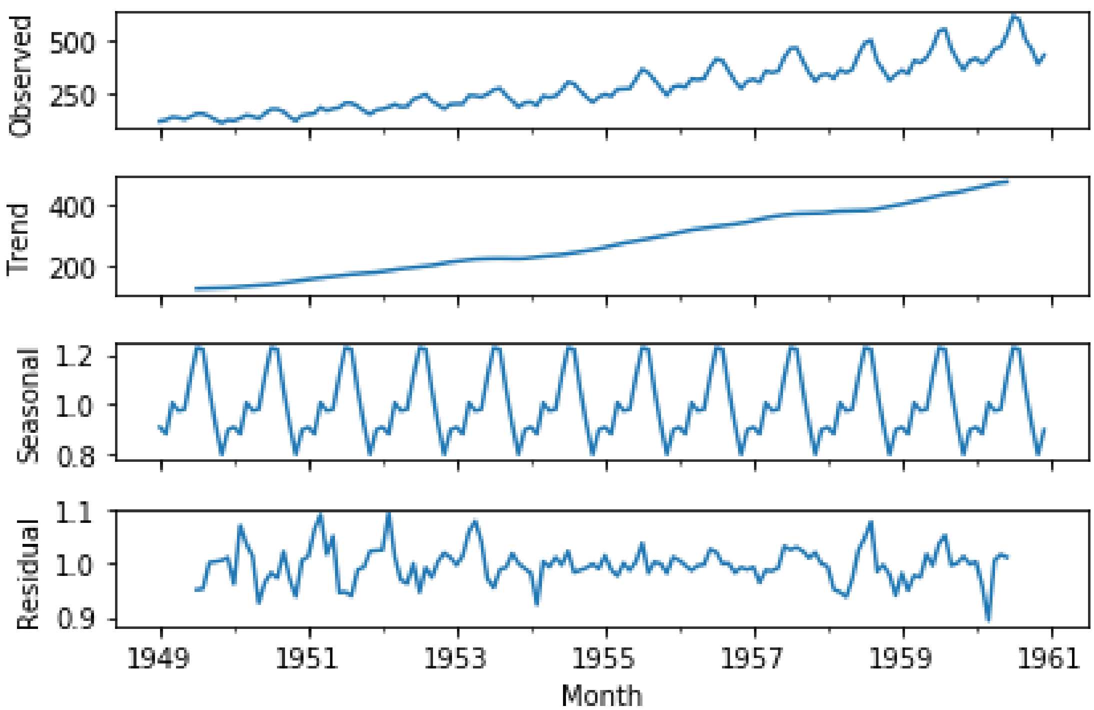
[5]:
```
airline.plot();
```



Based on this chart, it looks like the trend in the earlier days is increasing at a higher rate than just linear (although it is a bit hard to tell from this one plot).

[6]:
```
from statsmodels.tsa.seasonal import seasonal_decompose

result = seasonal_decompose(airline['Thousands of Passengers'],␣
 ↪model='multiplicative')  # model='mul' also works
result.plot();
```



Great! In the next section we'll see how to apply exponential smoothing models to each of these contributing factors.