Untitled38

July 5, 2020

1 ETS Models

In [72]:
```
## Load the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
## import the data
df = pd.read_csv("/Users/snigdhacheekoty/Desktop/prac/airline_passengers.csv", index_c
                 parse_dates = True)
```

In [3]:
```
df.head(10)
```

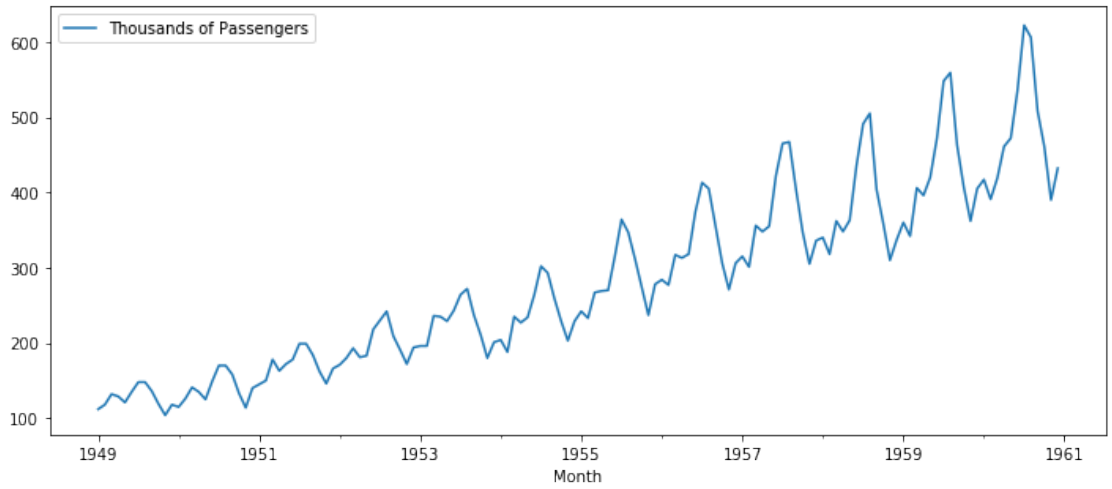
Out[3]:

Month	Thousands of Passengers
1949-01-01	112
1949-02-01	118
1949-03-01	132
1949-04-01	129
1949-05-01	121
1949-06-01	135
1949-07-01	148
1949-08-01	148
1949-09-01	136
1949-10-01	119

In [9]:
```
### there are no missing values
```

In [11]:
```
## plotting the observations
df.plot(figsize = (12,5))
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x116c3de48>
```




```
In [12]: ## The trend look exponential than linear
         ## SO we apply the multiplicative ETS Model
         from statsmodels.tsa.seasonal import seasonal_decompose
         result = seasonal_decompose(df["Thousands of Passengers"], model = "multiplicative")
         type(result)
```

```
/anaconda3/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The panc
  from pandas.core import datetools
```

```
Out[12]: statsmodels.tsa.seasonal.DecomposeResult
```

```
In [14]: result.trend.head(15) # observations of trend component
```

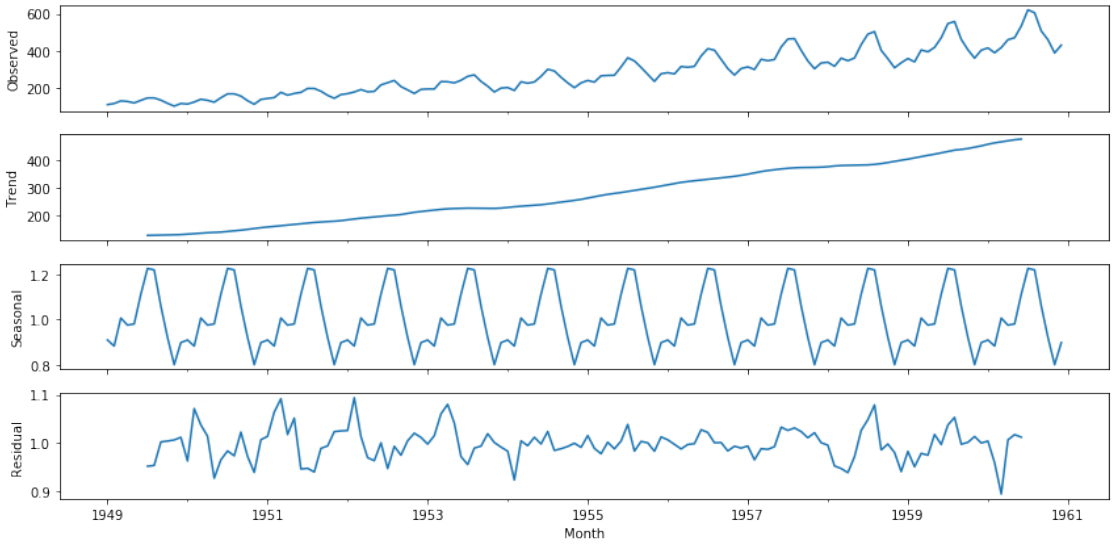
```
Out[14]: Month
         1949-01-01           NaN
         1949-02-01           NaN
         1949-03-01           NaN
         1949-04-01           NaN
         1949-05-01           NaN
         1949-06-01           NaN
         1949-07-01    126.791667
         1949-08-01    127.250000
         1949-09-01    127.958333
         1949-10-01    128.583333
         1949-11-01    129.000000
         1949-12-01    129.750000
         1950-01-01    131.250000
         1950-02-01    133.083333
         1950-03-01    134.916667
         Name: Thousands of Passengers, dtype: float64
```

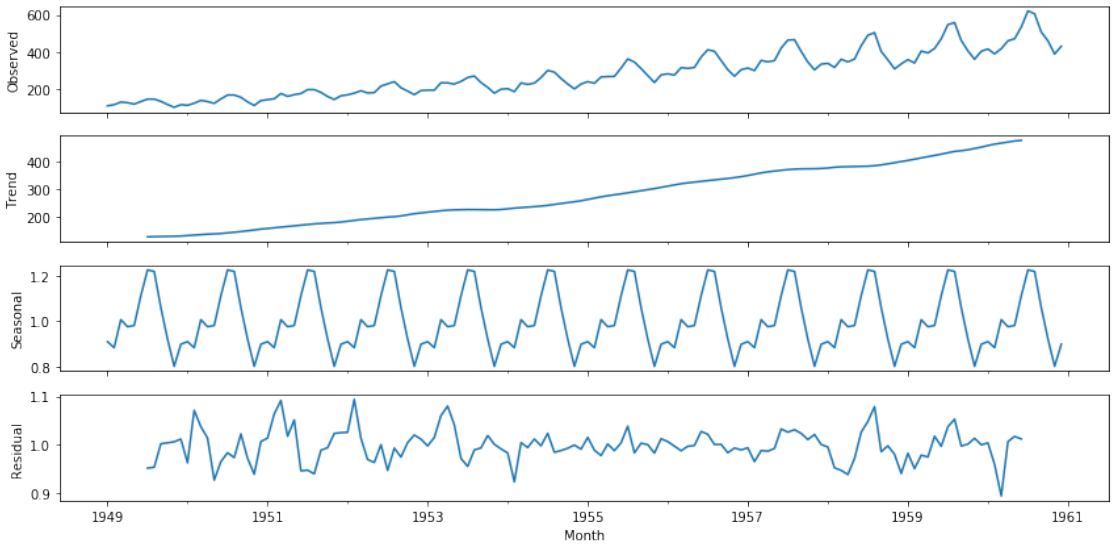
```
In [15]: result.seasonal.head(15) # observations of seasonal component

Out[15]: Month
         1949-01-01    0.910230
         1949-02-01    0.883625
         1949-03-01    1.007366
         1949-04-01    0.975906
         1949-05-01    0.981378
         1949-06-01    1.112776
         1949-07-01    1.226556
         1949-08-01    1.219911
         1949-09-01    1.060492
         1949-10-01    0.921757
         1949-11-01    0.801178
         1949-12-01    0.898824
         1950-01-01    0.910230
         1950-02-01    0.883625
         1950-03-01    1.007366
         Name: Thousands of Passengers, dtype: float64

In [17]: result.resid.head(15)  # observations of residual component

Out[17]: Month
         1949-01-01         NaN
         1949-02-01         NaN
         1949-03-01         NaN
         1949-04-01         NaN
         1949-05-01         NaN
         1949-06-01         NaN
         1949-07-01    0.951664
         1949-08-01    0.953401
         1949-09-01    1.002220
         1949-10-01    1.004028
         1949-11-01    1.006270
         1949-12-01    1.011812
         1950-01-01    0.962603
         1950-02-01    1.071467
         1950-03-01    1.037447
         Name: Thousands of Passengers, dtype: float64

In [19]: ## Plotting the trend, seasonality and residual components
         result.plot(figsize = (12,5))  ## This doesn't work because there are MULTIPLE PLOTS


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-19-b6e5abd2a7a5> in <module>()
```

```
      1 ## Plotting the trend, seasonality and residual components
----> 2 result.plot(figsize = (12,5))


TypeError: plot() got an unexpected keyword argument 'figsize'
```

In [22]:
```
## Plotting the trend, seasonality and residual components
from pylab import rcParams
rcParams["figure.figsize"] = 12,6
result.plot()
```

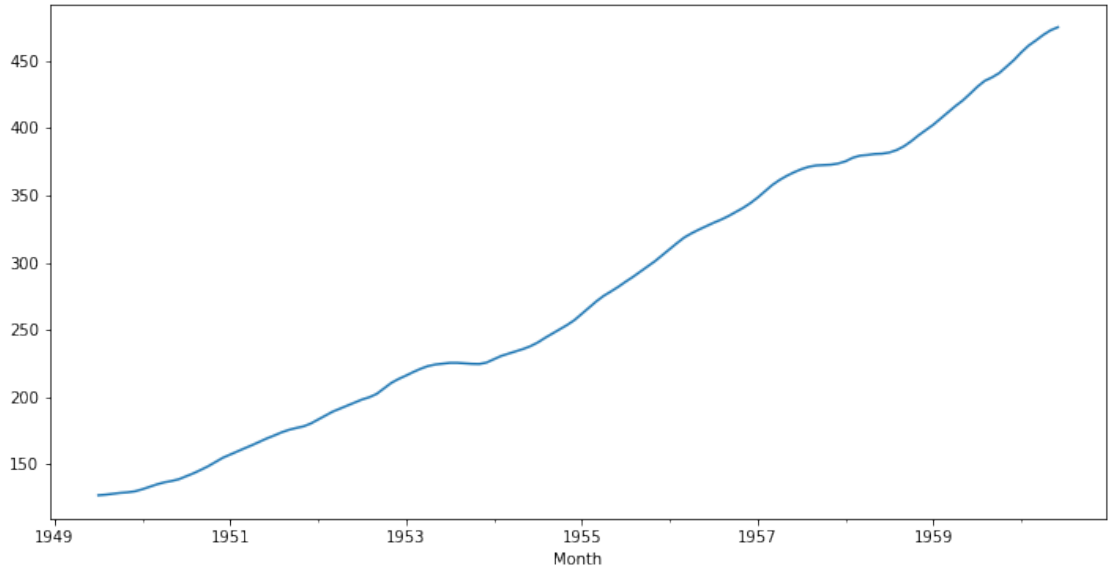
Out[22]:





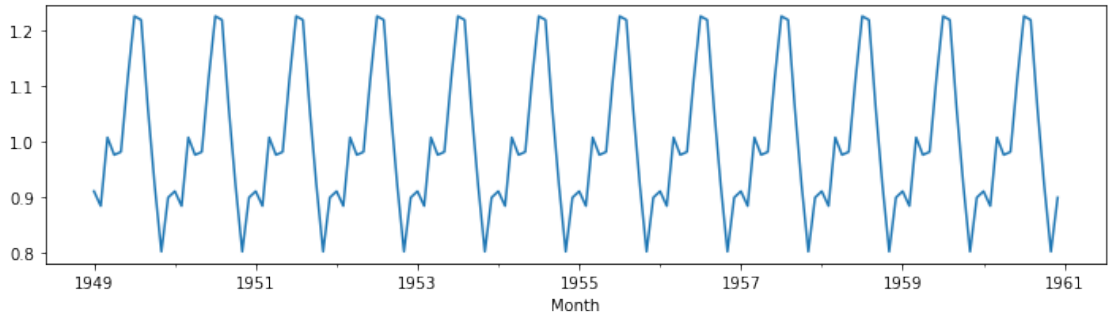
```
In [23]: ## Plotting the components individually
         result.trend.plot()
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1c19c6b390>
```



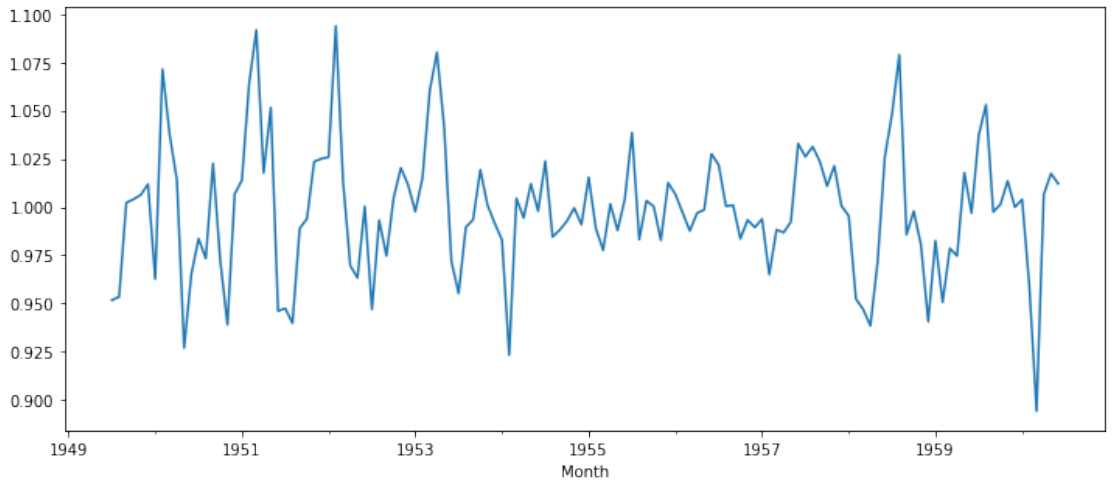

```
In [26]: result.seasonal.plot(figsize = (12,3))
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1c19e7e550>
```




```
In [28]: result.resid.plot(figsize = (12,5))
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1a256c50>
```



2 EWMA Model

```
In [29]: ### "EWMA" - Exponentially Weighted Moving Average Model
         ### More sophisticated model for analysing the time series trends
         ## It's a simple exponential smoothing method
         ## libraries required: numpy, pandas, matplotlib

In [31]: df.head(10)

Out[31]:             Thousands of Passengers
         Month
         1949-01-01                      112
         1949-02-01                      118
         1949-03-01                      132
         1949-04-01                      129
         1949-05-01                      121
         1949-06-01                      135
         1949-07-01                      148
         1949-08-01                      148
         1949-09-01                      136
         1949-10-01                      119

In [32]: df.index

Out[32]: DatetimeIndex(['1949-01-01', '1949-02-01', '1949-03-01', '1949-04-01',
                        '1949-05-01', '1949-06-01', '1949-07-01', '1949-08-01',
                        '1949-09-01', '1949-10-01',
                        ...
                        '1960-03-01', '1960-04-01', '1960-05-01', '1960-06-01',
                        '1960-07-01', '1960-08-01', '1960-09-01', '1960-10-01',
```

```
                    '1960-11-01', '1960-12-01'],
                   dtype='datetime64[ns]', name='Month', length=144, freq=None)
```

```
In [34]: ## Applying SMA - Simple Moving Average
         df["6-month avg"] = df["Thousands of Passengers"].rolling(window = 6).mean()
```

```
In [35]: df["12-month avg"] = df["Thousands of Passengers"].rolling(window = 12).mean()
```

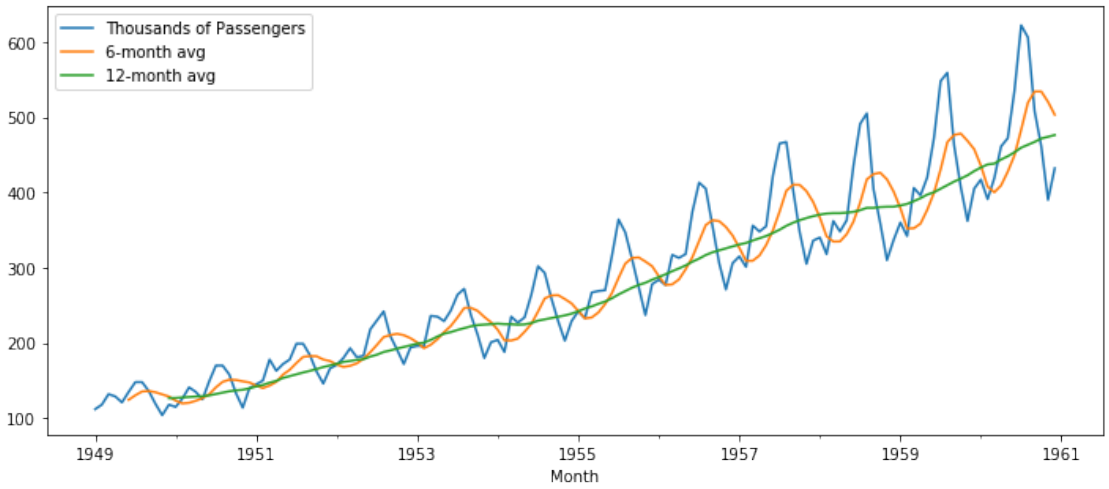
```
In [37]: df.head(20)
```

Out[37]:

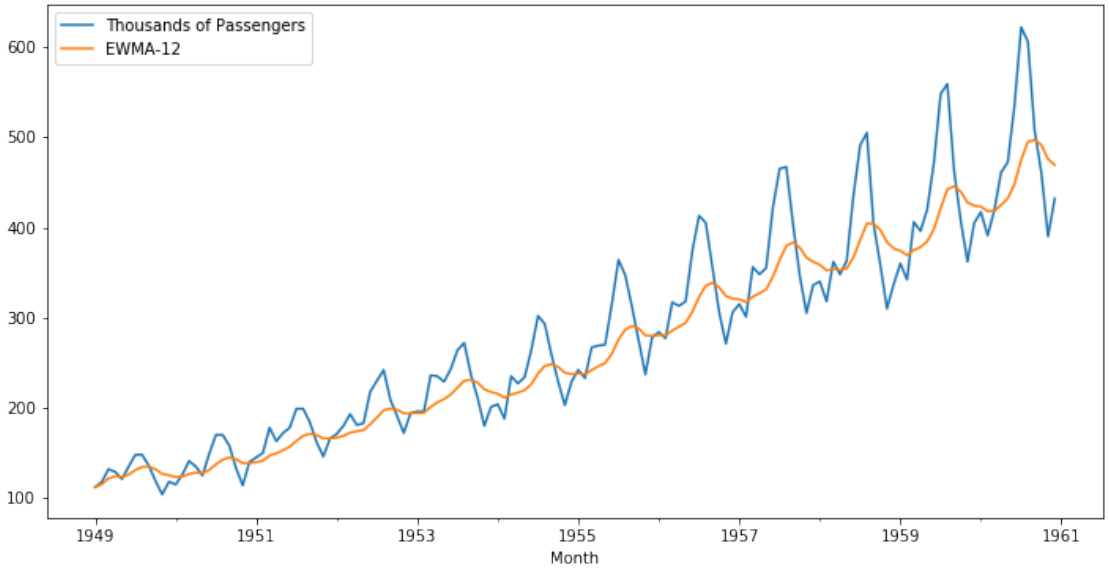
Month	Thousands of Passengers	6-month avg	12-month avg
1949-01-01	112	NaN	NaN
1949-02-01	118	NaN	NaN
1949-03-01	132	NaN	NaN
1949-04-01	129	NaN	NaN
1949-05-01	121	NaN	NaN
1949-06-01	135	124.500000	NaN
1949-07-01	148	130.500000	NaN
1949-08-01	148	135.500000	NaN
1949-09-01	136	136.166667	NaN
1949-10-01	119	134.500000	NaN
1949-11-01	104	131.666667	NaN
1949-12-01	118	128.833333	126.666667
1950-01-01	115	123.333333	126.916667
1950-02-01	126	119.666667	127.583333
1950-03-01	141	120.500000	128.333333
1950-04-01	135	123.166667	128.833333
1950-05-01	125	126.666667	129.166667
1950-06-01	149	131.833333	130.333333
1950-07-01	170	141.000000	132.166667
1950-08-01	170	148.333333	134.000000

```
In [38]: ## Plotting SMA
         df.plot(figsize = (12,5))
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1a5829b0>
```




```
In [39]: ## Applying EWMA
         df["EWMA-12"] = df["Thousands of Passengers"].ewm(span = 12).mean()

In [40]: df[["Thousands of Passengers","EWMA-12"]].plot()

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1a85c710>
```



3 HOLT WINTERS METHOD

```
In [42]: # EWMA - SIMPLE Exponential Smoothing
         # Holt Winters- DOUBLE Exponential Smoothing and TRIPLE Exponential Smoothing
         # Parameters - 'alpha' for Level, 'beta' for trend, 'gamma' for seasonality
```

In [44]:
```
## libraries - numpy , pandas , matplotlib

df. index
```

Out[44]:
```
DatetimeIndex(['1949-01-01', '1949-02-01', '1949-03-01', '1949-04-01',
               '1949-05-01', '1949-06-01', '1949-07-01', '1949-08-01',
               '1949-09-01', '1949-10-01',
               ...
               '1960-03-01', '1960-04-01', '1960-05-01', '1960-06-01',
               '1960-07-01', '1960-08-01', '1960-09-01', '1960-10-01',
               '1960-11-01', '1960-12-01'],
              dtype='datetime64[ns]', name='Month', length=144, freq=None)
```

In [53]:
```
## As you see above, freq = None. This needs to be changed to fit the model using Sta
## Because: Data has frequency and Statsmodels needs to know it


## Hence, Setting the frequency to index
df.index.freq = 'MS'
## MS - Because the observations start at the starting of each month
```

In [54]:
```
df.head()
```

Out[54]:

Month	Thousands of Passengers	6-month avg	12-month avg	EWMA-12
1949-01-01	112	NaN	NaN	112.000000
1949-02-01	118	NaN	NaN	115.250000
1949-03-01	132	NaN	NaN	121.787529
1949-04-01	129	NaN	NaN	124.064224
1949-05-01	121	NaN	NaN	123.231685

In [60]:
```
## Simple Exponential Smoothing using EWM
import pandas as pd
span = 12  # 12 months in a year
alpha = 2/(span+1) # alpha - formula

df["EWMA12"] = df["Thousands of Passengers"].ewm(alpha = alpha, adjust = False).mean()
```

In [61]:
```
df.head(10)
```

Out[61]:

Month	Thousands of Passengers	6-month avg	12-month avg	EWMA-12
1949-01-01	112	NaN	NaN	112.000000
1949-02-01	118	NaN	NaN	115.250000
1949-03-01	132	NaN	NaN	121.787529
1949-04-01	129	NaN	NaN	124.064224
1949-05-01	121	NaN	NaN	123.231685
1949-06-01	135	124.500000	NaN	126.092005
1949-07-01	148	130.500000	NaN	130.980697

```
1949-08-01                      148   135.500000          NaN  134.532364
1949-09-01                      136   136.166667          NaN  134.822714
1949-10-01                      119   134.500000          NaN  131.824316

                 EWMA12
Month
1949-01-01  112.000000
1949-02-01  112.923077
1949-03-01  115.857988
1949-04-01  117.879836
1949-05-01  118.359861
1949-06-01  120.919883
1949-07-01  125.086055
1949-08-01  128.611277
1949-09-01  129.748004
1949-10-01  128.094465
```

```
In [98]: ## Simple Exponential Smoothing using HoltWinters model
         from statsmodels.tsa.holtwinters import SimpleExpSmoothing
```

```
In [110]: fit = SimpleExpSmoothing(df["Thousands of Passengers"]).fit(smoothing_level= alpha, o
```

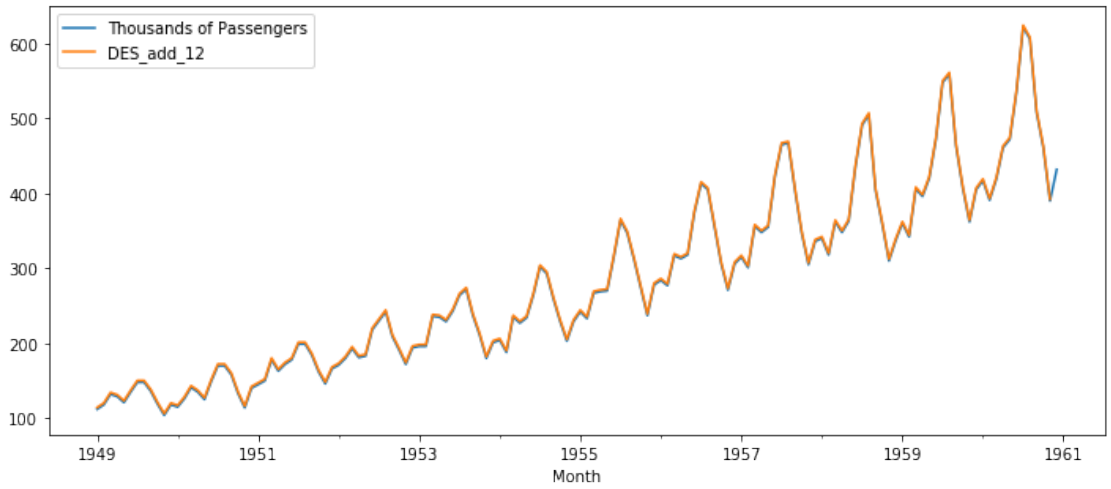
```
In [116]: ## Double Exponential Smoothing using HoltWinters model
          from statsmodels.tsa.holtwinters import ExponentialSmoothing
          df["DES_add_12"] = ExponentialSmoothing(df["Thousands of Passengers"],trend="add").fi
```

```
In [117]: df.head()
```

```
Out[117]:             Thousands of Passengers  6-month avg  12-month avg     EWMA-12  \
          Month
          1949-01-01                      112          NaN           NaN  112.000000
          1949-02-01                      118          NaN           NaN  115.250000
          1949-03-01                      132          NaN           NaN  121.787529
          1949-04-01                      129          NaN           NaN  124.064224
          1949-05-01                      121          NaN           NaN  123.231685

                         EWMA12  DES_add_12
          Month
          1949-01-01  112.000000  114.336734
          1949-02-01  112.923077  120.336734
          1949-03-01  115.857988  134.336734
          1949-04-01  117.879836  131.336734
          1949-05-01  118.359861  123.336734
```

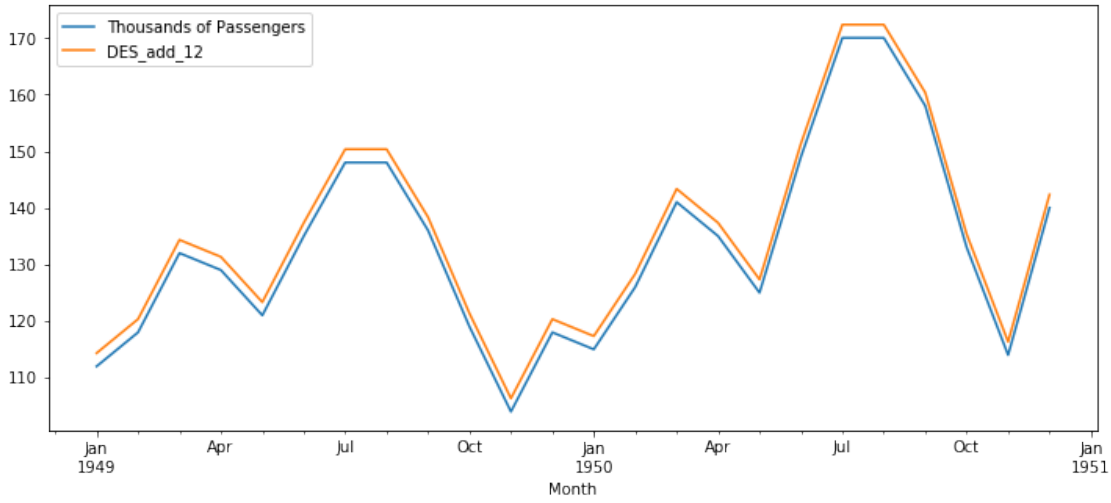
```
In [119]: ## Plotting the double exponential moving average
          df[["Thousands of Passengers", "DES_add_12"]].plot(figsize = (12,5))
```

```
Out[119]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1c5f57f0>
```




```
In [120]: df[["Thousands of Passengers", "DES_add_12"]].iloc[:24].plot(figsize = (12,5))
```

```
Out[120]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1c5235c0>
```




```
In [125]: ## Triple Exponential Smoothing
          from statsmodels.tsa.holtwinters import ExponentialSmoothing
          df["DES_mul_12"] = ExponentialSmoothing(df["Thousands of Passengers"], trend = "mul"
```

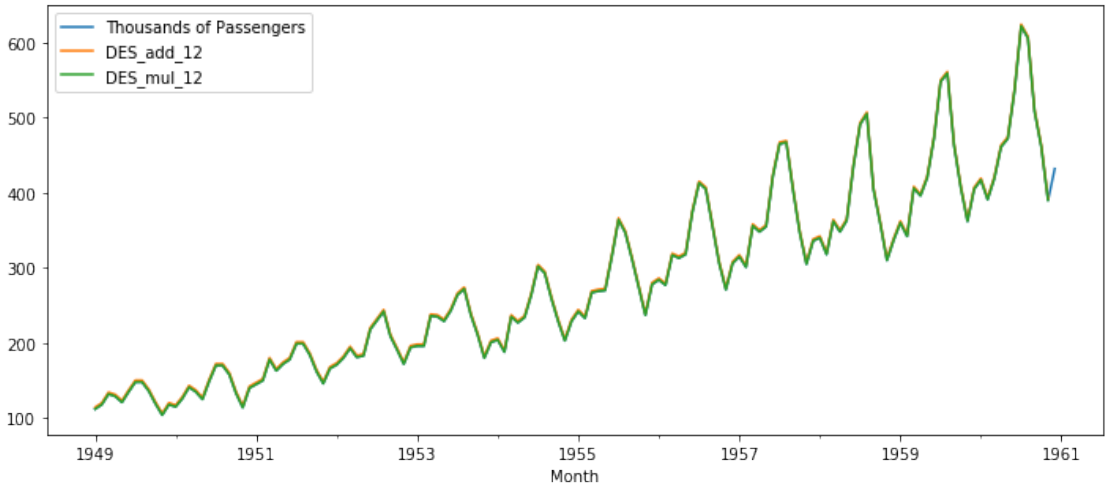
```
In [126]: df.head()
```

```
Out[126]:             Thousands of Passengers  6-month avg  12-month avg     EWMA-12  \
          Month
```

```
1949-01-01                    112          NaN          NaN  112.000000
1949-02-01                    118          NaN          NaN  115.250000
1949-03-01                    132          NaN          NaN  121.787529
1949-04-01                    129          NaN          NaN  124.064224
1949-05-01                    121          NaN          NaN  123.231685

                 EWMA12  DES_add_12  DES_mul_12
Month
1949-01-01  112.000000  114.336734  112.049247
1949-02-01  112.923077  120.336734  118.051885
1949-03-01  115.857988  134.336734  132.058041
1949-04-01  117.879836  131.336734  129.056722
1949-05-01  118.359861  123.336734  121.053204
```

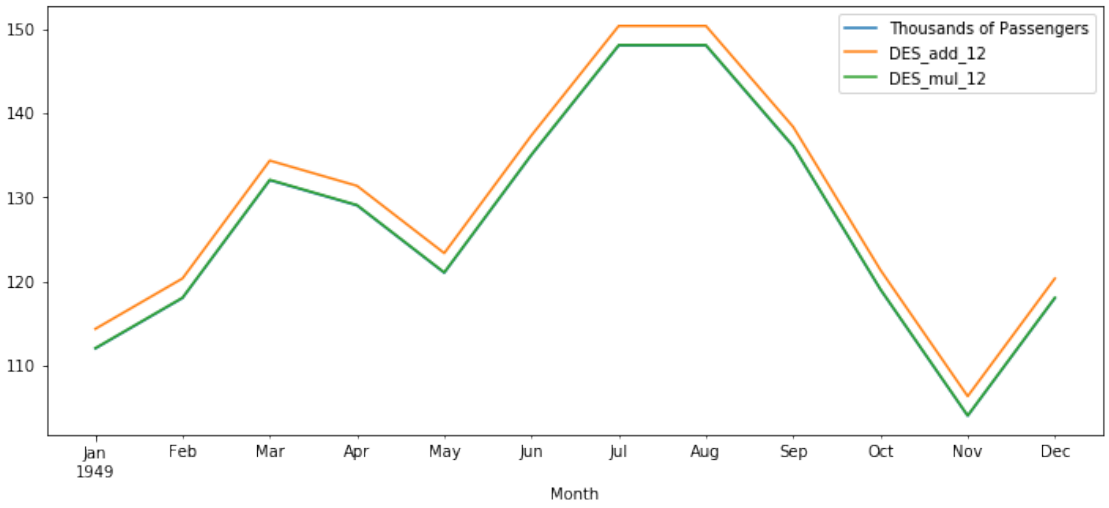
In [131]:
```
## Plotting
df[["Thousands of Passengers","DES_add_12", "DES_mul_12"]].plot(figsize = (12,5))
```

Out[131]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1c570c50>



In [134]:
```
df[["Thousands of Passengers","DES_add_12", "DES_mul_12"]].iloc[:12].plot(figsize =
```

Out[134]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1ca91668>



In [ ]: *## We can see that multiplicative model models the data better*