Time Series Decomposition

Contents

1. Components of Time Series

2. Understanding Moving Averages

3. Time Series Decomposition

Components of Time Series

- As we know, Trend and Seasonality are main components of Time Series.
- If we assume an additive model, we can write

where,

$$Y_t = S_t + T_t + R_t$$

Y_t : Time series value at period t

S_t : Seasonal component at period t

T_t : Trend-cycle component at period t

R_t : Remainder (or irregular or error)

component at period t

- Alternatively, a multiplicative model would be written as

$$Y_t = S_t * T_t * R_t$$

Additive model is most appropriate if the magnitude of the seasonal fluctuations or the variation around the trend-cycle does not vary with the level of the time series.

Understanding Moving Averages

- Moving Averages are **averages calculated for consecutive data from overlapping subgroups of fixed length**
- Moving averages smoothen a time series by filtering out random fluctuations

Day	End of Day Sales
1	1500
2	2100
3	1750
4	1900
5	1650

Moving Average of Period 3
NA
NA
(1500+2100+1750)/3=1783.33
(2100+1750+1900)/3=1916.67
(1750+1900+1650)/3=1766.67

The first 2 MA values for length 3 are not calculated

- Period of the moving average depends on type of data
- **Non-seasonal data**: Shorter length (Typically 3 period or 5 period MA is considered)
- **Seasonal data**: Typical period is 12 for monthly data and 4 for quarterly data

Time Series Decomposition – Simple Method

Decomposition is a statistical method that deconstructs a time series.

Steps to follow :

Find Trend

Obtain moving averages covering one season – This provides trend component of the time series

Eliminate Trend

Eliminate trend component from original time series. Calculate $Y_t - T_t$

Estimate Seasonality

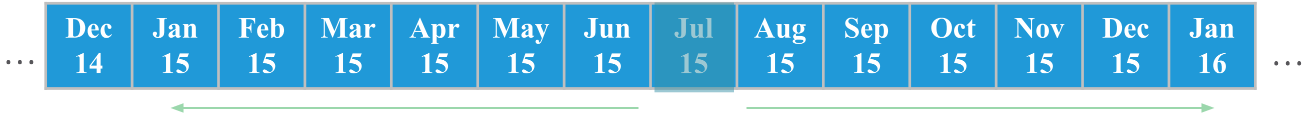
To estimate the seasonal component for a given time period, simply average the de-trended values for that time period. These seasonal indexes are then adjusted to ensure that they add to zero

The remainder component is calculated by subtracting the estimated seasonal and trend-cycle components

Time Series Decomposition – Simple Method

Suppose we have monthly time series data, for three years 2014, 2015 and 2016:

Step 1 □ Calculate Moving Averages



…	Dec 14	Jan 15	Feb 15	Mar 15	Apr 15	May 15	Jun 15	Jul 15	Aug 15	Sep 15	Oct 15	Nov 15	Dec 15	Jan 16	…

(Consider moving average period of 13 - previous 6 months, next 6 months and current month to calculate moving average of current month)

This gives the trend component T_t

Step 2 □ Eliminating Trend

Remove T_t from the original time series Y_t

Step 3 □ Estimate Seasonal Component

The seasonal index for July is the average of all the de-trended July values in the data i.e. Average of De-trended July 2014, July 2015 and July 2016

Case Study

Background

- Monthly Sales Data for 3 Years (2013, 2014, 2015)

Objective

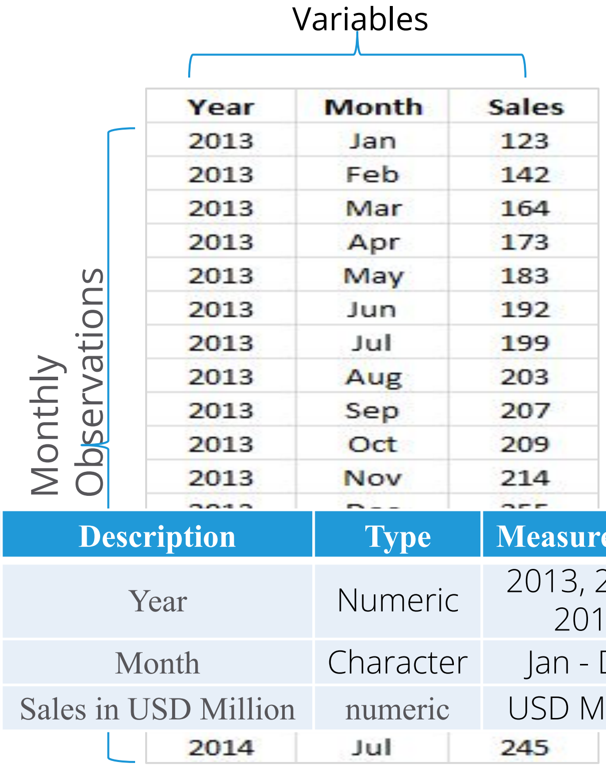
- To decompose time series into its components and study each component separately.

Available Information

- Sample size is 36
- Variables: Year, Month, Sales

Data Snapshot

Sales Data for 3 Years



Variables

Monthly Observations

Year	Month	Sales
2013	Jan	123
2013	Feb	142
2013	Mar	164
2013	Apr	173
2013	May	183
2013	Jun	192
2013	Jul	199
2013	Aug	203
2013	Sep	207
2013	Oct	209
2013	Nov	214
2013	Dec	255
2014	Jul	245

Columns	Description	Type	Measurement	Possible values
Year	Year	Numeric	2013, 2014, 2015	3
Month	Month	Character	Jan - Dec	12
Sales	Sales in USD Million	numeric	USD Million	Positive values

Time Series Decomposition in Python

```
# Simple Decomposition
import pandas as pd
salesdata = pd.read_csv("Sales Data for 3 Years.csv")
rng = pd.date_range('01-01-2013','31-12-2015',freq='M')
s = salesdata.Sales.values
salesseries = pd.Series(s, rng)
```

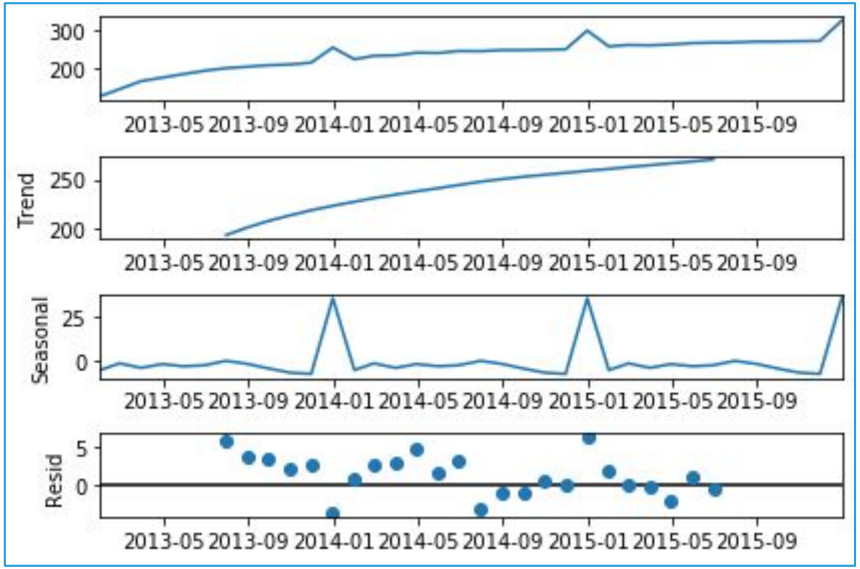
- **freq =** “M” indicates monthly date data.
- **pd.Series()** creates time series object using date object as index

```
import statsmodels.api as sm
decomp = sm.tsa.seasonal_decompose(salesseries.interpolate())
decomp.plot()
```

- **tsa.seasonal_decompose()** performs a classical seasonal decomposition through moving averages.
- **plot()** of decompose object gives a 4-level visual representation.

Time Series Decomposition in Python

```
# Output
```



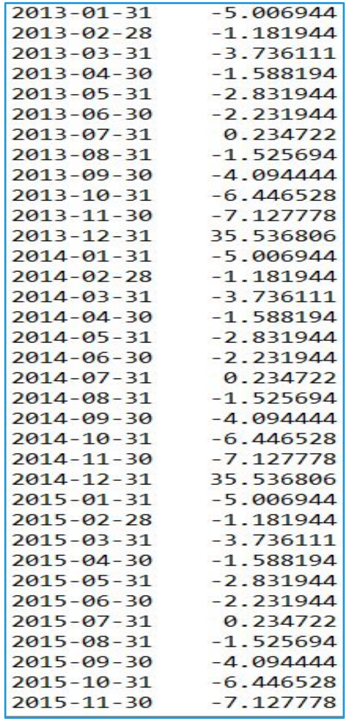
Trend is not estimated for first/last few values
Seasonal Component repeats from year to year

* The **`tsa.seasonal_decompose()`** function takes time series as additive by default. If you wish to create a multiplicative time series, simply add **`model=“multiplicative”`** in the function.

Decomposition in Python – Seasonal Component

```
#Analyzing the decomp object. Each component can be separately viewed
```

```
decomp.seasonal
```



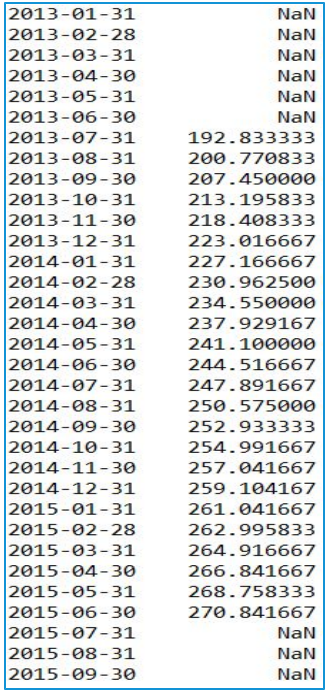
```
2013-01-31     -5.006944
2013-02-28     -1.181944
2013-03-31     -3.736111
2013-04-30     -1.588194
2013-05-31     -2.831944
2013-06-30     -2.231944
2013-07-31      0.234722
2013-08-31     -1.525694
2013-09-30     -4.094444
2013-10-31     -6.446528
2013-11-30     -7.127778
2013-12-31     35.536806
2014-01-31     -5.006944
2014-02-28     -1.181944
2014-03-31     -3.736111
2014-04-30     -1.588194
2014-05-31     -2.831944
2014-06-30     -2.231944
2014-07-31      0.234722
2014-08-31     -1.525694
2014-09-30     -4.094444
2014-10-31     -6.446528
2014-11-30     -7.127778
2014-12-31     35.536806
2015-01-31     -5.006944
2015-02-28     -1.181944
2015-03-31     -3.736111
2015-04-30     -1.588194
2015-05-31     -2.831944
2015-06-30     -2.231944
2015-07-31      0.234722
2015-08-31     -1.525694
2015-09-30     -4.094444
2015-10-31     -6.446528
2015-11-30     -7.127778
```

Interpretation :

- This table shows seasonal component of time series

Decomposition in Python – Trend Component

```
decomp.trend
```



```
2013-01-31           NaN
2013-02-28           NaN
2013-03-31           NaN
2013-04-30           NaN
2013-05-31           NaN
2013-06-30           NaN
2013-07-31    192.833333
2013-08-31    200.770833
2013-09-30    207.450000
2013-10-31    213.195833
2013-11-30    218.408333
2013-12-31    223.016667
2014-01-31    227.166667
2014-02-28    230.962500
2014-03-31    234.550000
2014-04-30    237.929167
2014-05-31    241.100000
2014-06-30    244.516667
2014-07-31    247.891667
2014-08-31    250.575000
2014-09-30    252.933333
2014-10-31    254.991667
2014-11-30    257.041667
2014-12-31    259.104167
2015-01-31    261.041667
2015-02-28    262.995833
2015-03-31    264.916667
2015-04-30    266.841667
2015-05-31    268.758333
2015-06-30    270.841667
2015-07-31           NaN
2015-08-31           NaN
2015-09-30           NaN
```

Interpretation :

- This table shows trend component of time series

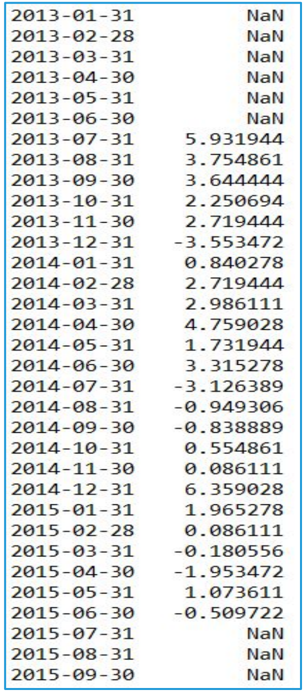
?

Why are NAs getting generated?

This is because trend not estimated for first/last few values. Consequently, the same will be reflected in random component as well.

Decomposition in Python – Random Component

```
decomp.resid
```



```
2013-01-31          NaN
2013-02-28          NaN
2013-03-31          NaN
2013-04-30          NaN
2013-05-31          NaN
2013-06-30          NaN
2013-07-31     5.931944
2013-08-31     3.754861
2013-09-30     3.644444
2013-10-31     2.250694
2013-11-30     2.719444
2013-12-31    -3.553472
2014-01-31     0.840278
2014-02-28     2.719444
2014-03-31     2.986111
2014-04-30     4.759028
2014-05-31     1.731944
2014-06-30     3.315278
2014-07-31    -3.126389
2014-08-31    -0.949306
2014-09-30    -0.838889
2014-10-31     0.554861
2014-11-30     0.086111
2014-12-31     6.359028
2015-01-31     1.965278
2015-02-28     0.086111
2015-03-31    -0.180556
2015-04-30    -1.953472
2015-05-31     1.073611
2015-06-30    -0.509722
2015-07-31          NaN
2015-08-31          NaN
2015-09-30          NaN
```

Interpretation :

- This table shows random component of time series

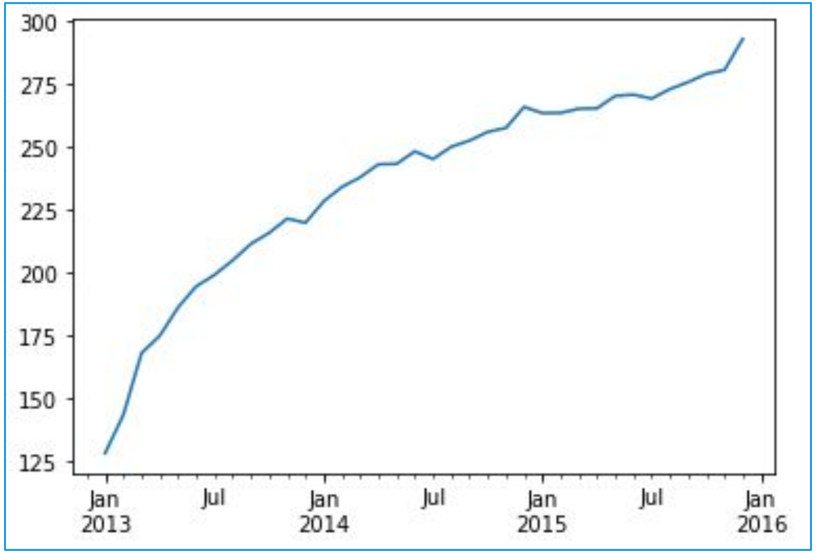
?

Why are NAs getting generated?

This is because trend not estimated for first/last few values. Consequently, the same will be reflected in random component as well.

Seasonally Adjusted Time Series

```
# Doing Seasonal Adjustment
salesadj = salesseries - decomp.seasonal
salesadj.plot()
# Output
```



Interpretation :

- This plot shows seasonally adjusted time series

Quick Recap

In this session, we learnt about **time series decomposition and exponential smoothing**:

What is Decomposition

- A time series is made up of multiple components such as seasonality, trend, randomness
- Sometimes, studying these components separately provides a more comprehensive insight about the series

Decomposition in Python

- **`sm.tsa.seasonal_decompose()`** carries out simple seasonal decomposition