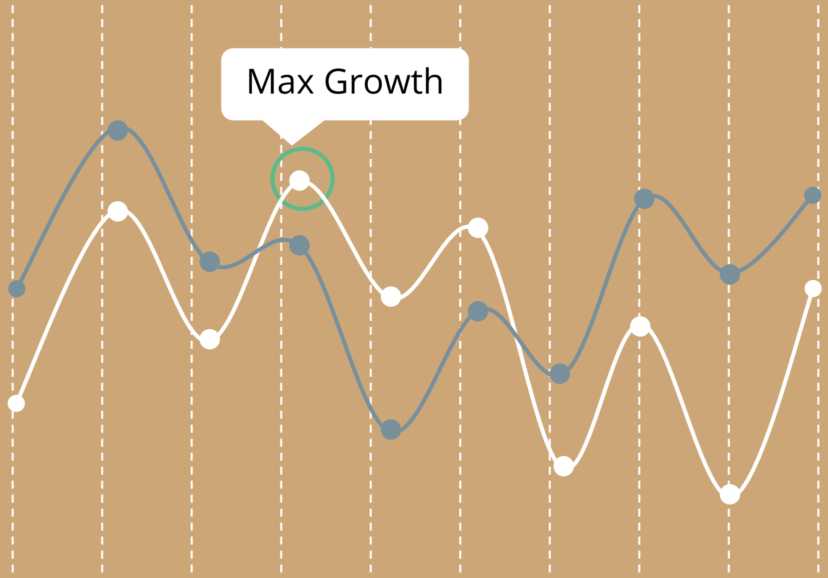
Time Series Analysis

Using ARIMA & Prophet

757 Python Meetup: 13th, July, 2017

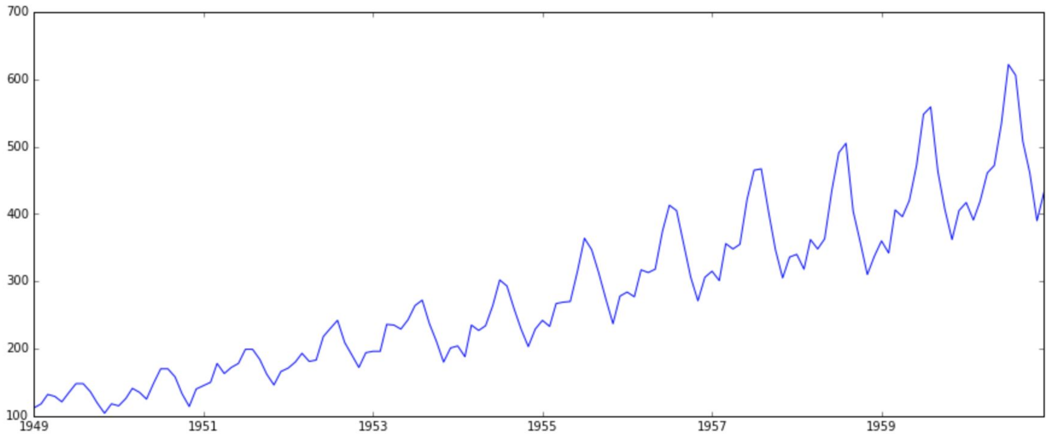
Raja Harsha Chinta (rchin001@odu.edu)



What makes a Time Series Special?

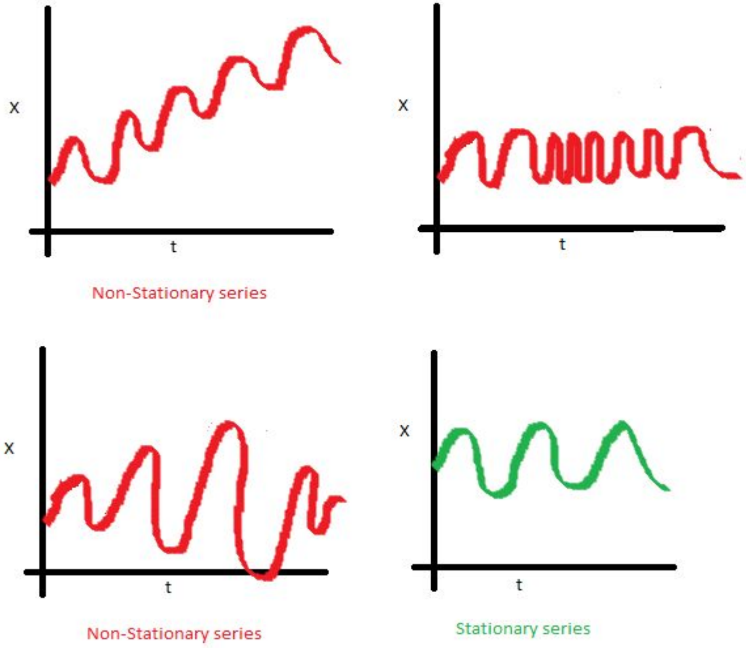
- TS is a collection of data points collected at constant time intervals
- Time Dependent
- Consists Trend and Seasonality
- Analyzed to determine the long term trend and forecast the future

	ds	y
0	2007-12-10	9.590761
1	2007-12-11	8.519590
2	2007-12-12	8.183677
3	2007-12-13	8.072467
4	2007-12-14	7.893572



Stationarity of a Time Series

- TS with a particular behaviour over time → Very high probability to repeat
- The mean of the series should not be a function of time rather should be a constant.
- The variance of the series should not a be a function of time.
- The covariance of the i th term and the (i + m) th term should not be a function of time.



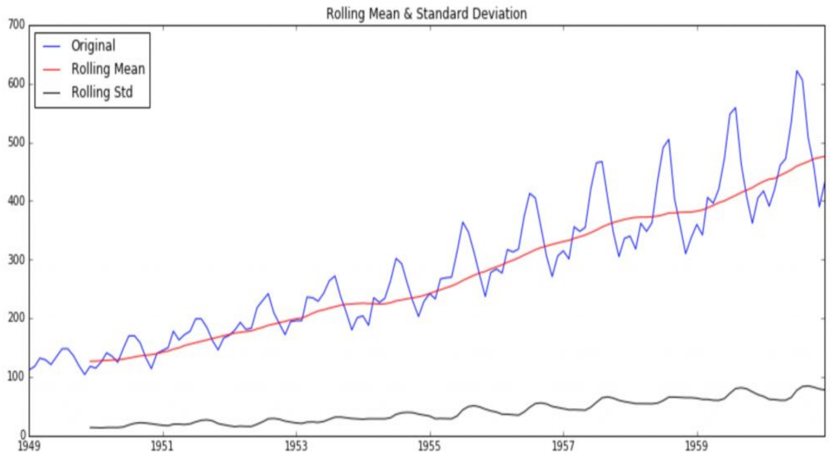
Check for Stationarity

Plotting Rolling Statistics:

- Plot the moving average or moving variance and see if it varies with time.
- More of a Visual technique.

Dickey-Fuller Test:

- Statistical test where NULL hypothesis is referred as Non-Stationary.
- Comprise of a **Test Statistic, Critical Values** for different confidence levels.
- If the **'Test Statistic' < 'Critical Value'**, we can reject the null hypothesis and say that the series is stationary.



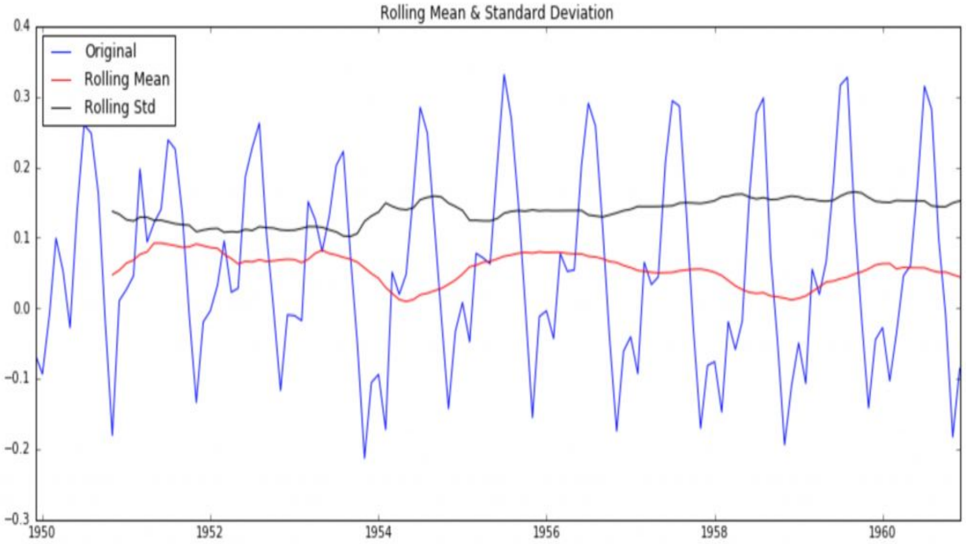

```
Results of Dickey-Fuller Test:
Test Statistic                   0.815369
p-value                          0.991880
#Lags Used                      13.000000
Number of Observations Used    130.000000
Critical Value (5%)             -2.884042
Critical Value (1%)             -3.481682
Critical Value (10%)            -2.578770
dtype: float64
```

Stationarize a Time Series

- **Trend** & **Seasonality** are two main reasons for Non - Stationarity.

- Apply transformations like **Log**, **Square, Cube Root**, etc. on TS to Stationarize.

- In real-time scenarios it is inevitable to see noise in data. In these cases we estimate trend by calculating:
  - Monthly, Weekly Averages
  - Rolling Averages (Smoothing)
  - Exponentially Weighted Moving Average (ewma)
  - Fit a regression model

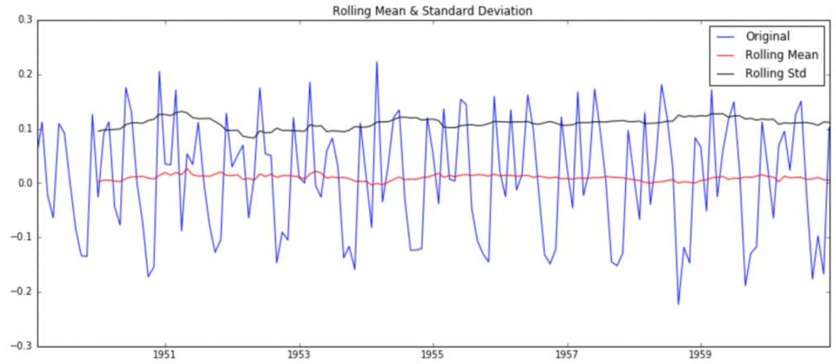



```
Results of Dickey-Fuller Test:
Test Statistic                 -3.162908
p-value                         0.022235
#Lags Used                     13.000000
Number of Observations Used   119.000000
Critical Value (5%)            -2.886151
Critical Value (1%)            -3.486535
Critical Value (10%)           -2.579896
dtype: float64
```

Eliminating Trend & Seasonality

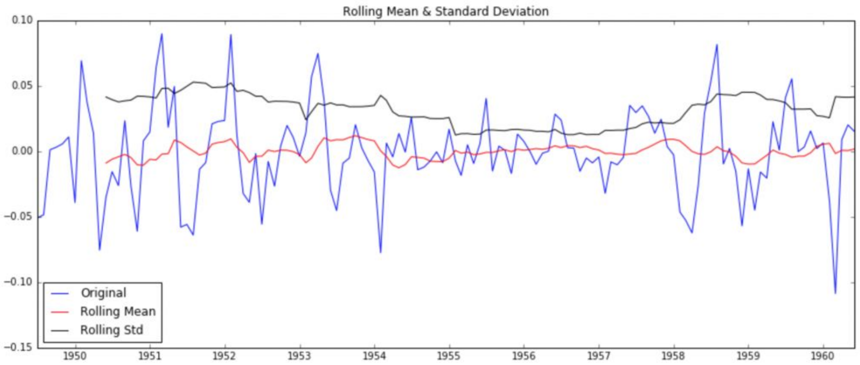
Simple Trend reduction techniques don’t work in all cases. Particularly, with high seasonality.

1. **Differencing** – Taking the difference with a particular time lag
2. **Decomposition** – Modeling both trend and seasonality and removing them from the model.




```
Results of Dickey-Fuller Test:
Test Statistic                  -2.717131
p-value                          0.071121
#Lags Used                      14.000000
Number of Observations Used    128.000000
Critical Value (5%)             -2.884398
Critical Value (1%)             -3.482501
Critical Value (10%)            -2.578960
dtype: float64
```

Differencing




```
Results of Dickey-Fuller Test:
Test Statistic                -6.332387e+00
p-value                        2.885059e-08
#Lags Used                     9.000000e+00
Number of Observations Used    1.220000e+02
Critical Value (5%)           -2.885538e+00
Critical Value (1%)           -3.485122e+00
Critical Value (10%)          -2.579569e+00
dtype: float64
```

Decomposition

ARIMA - Time Series Forecasting

A series with significant **dependence among values** need to use some statistical models like ARIMA to forecast the data. ARIMA stands for **Auto-Regressive Integrated Moving Averages**.

```
model = ARIMA(ts_log, order=(p, d, q))
```

The predictors depend on the parameters (p,d,q) of the ARIMA model:

- **Number of AR (Auto-Regressive) terms (p):**

  AR terms are just lags of dependent variable.

  For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5).

- **Number of MA (Moving Average) terms (q):**

  MA terms are lagged forecast errors in prediction equation.

  For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.

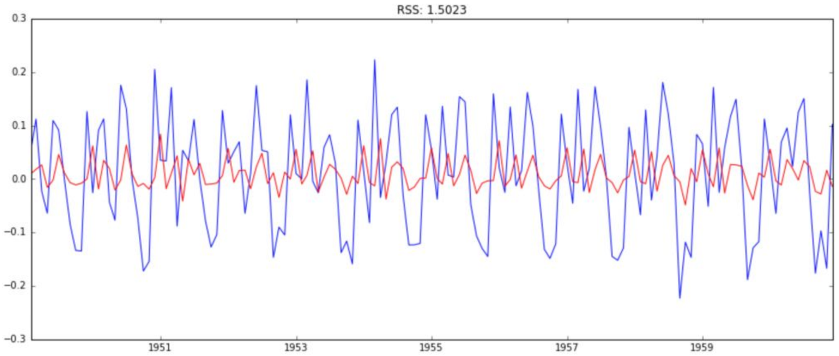
- **Number of Differences (d):**

  These are the number of non-seasonal differences.

AR & MA Models

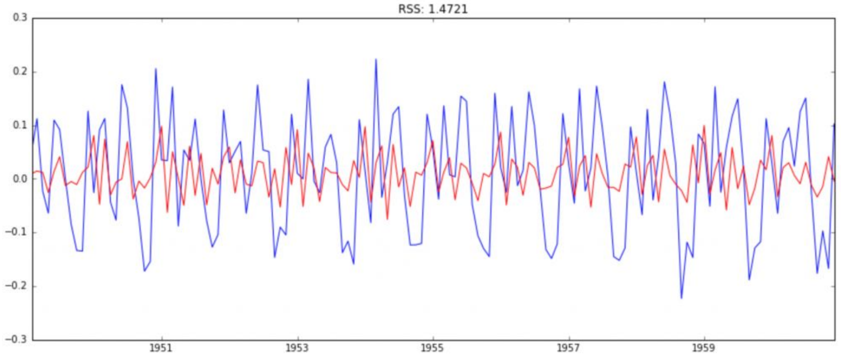
- AR Model : p != 0 and q = 0

```
model = ARIMA(ts_log, order=(2, 1, 0))
```



- MA Model : p = 0 and q != 0

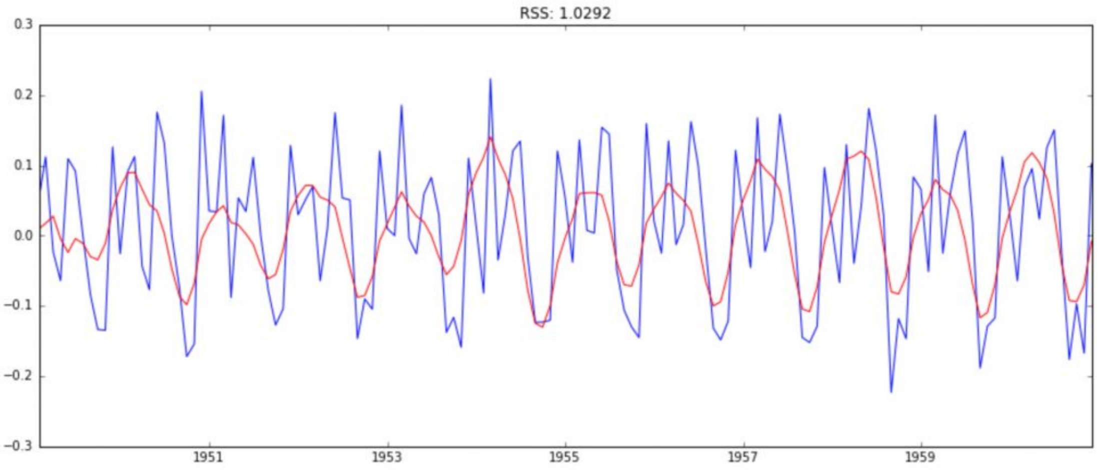
```
model = ARIMA(ts_log, order=(0, 1, 2))
```



ARIMA Model

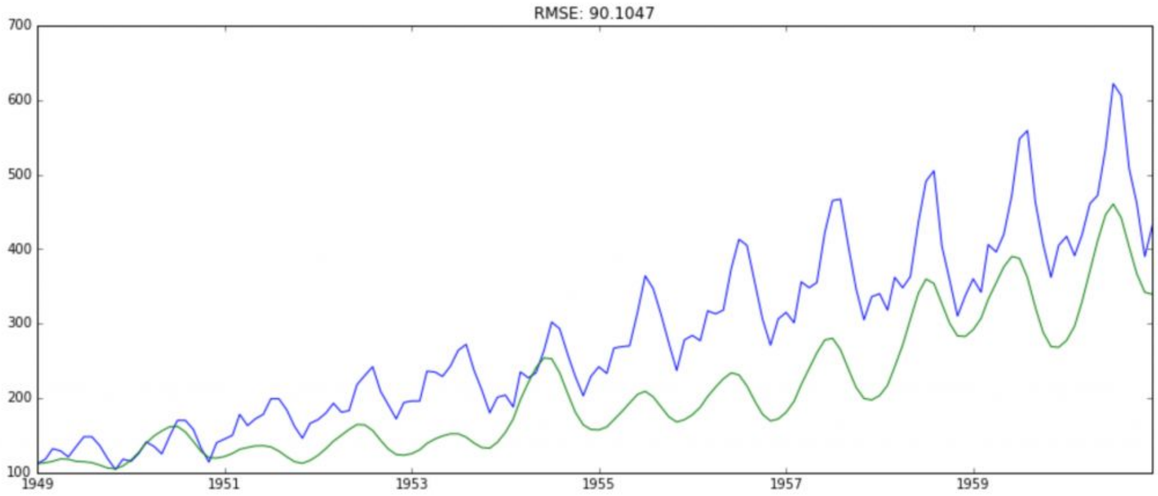
Both AR & MA models have RSS values **1.5023**, **1.4721** respectively and are further reduced to **1.0292** after combining both models into ARIMA with order **(2, 1, 2)**.

```
Model = ARIMA(ts_log, order=(2, 1, 2))
```



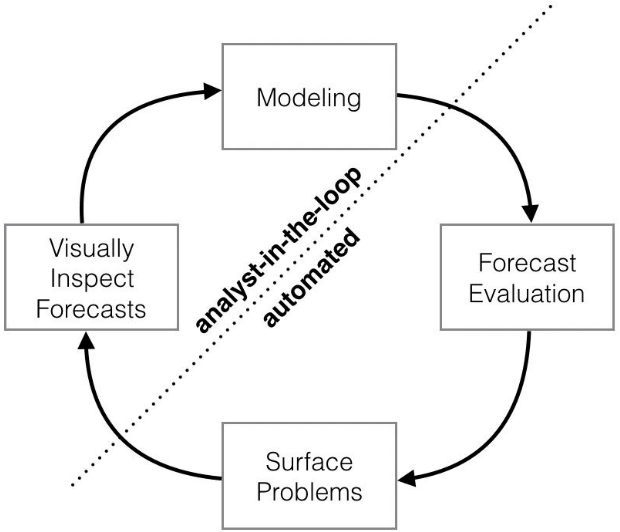
Forecasting back in Original Scale

```
model = ARIMA(ts_log, order=(2, 1, 2))
results_ARIMA = model.fit(disp=-1)

predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()

predictions_ARIMA_log = pd.Series(ts_log.ix[0], index=ts_log.index)
predictions_ARIMA_log = predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum,fill_value=0)

predictions_ARIMA = np.exp(predictions_ARIMA_log)
```



Facebook’s Prophet

- Prophet is a procedure for forecasting time series data.
- It is based on an additive model where non-linear trends are fit with yearly and weekly seasonality, plus holidays.
- It works best with daily periodicity data with at least one year of historical data.
- Prophet is robust to missing data, shifts in the trend, and large outliers.
- Accurate, Fast, Fully Automatic and Tunable forecasts.



Forecasting using Prophet

- The input to Prophet is always a dataframe with two columns: ds and y.
- **ds** & **y** - Represents the date stamp and measurement we wish to forecast.
- We fit the model by instantiating a new Prophet object and use Fit method.

```
Model = Prophet.fit(df)
```

- Create future dataframe by using default Prophet method - make_future_dataframe

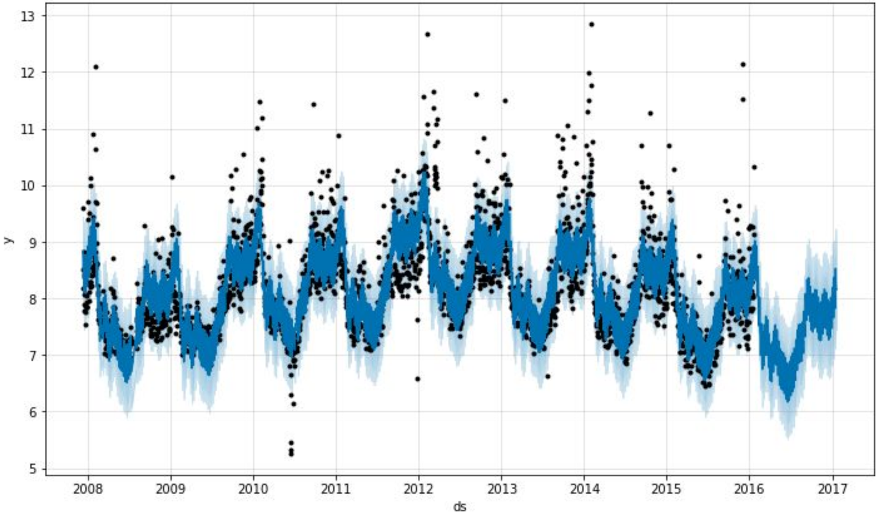
```
Future = Model.make_future_dataframe(periods=365)
```

- The predict method will assign each row in future a predicted value which it names yhat.

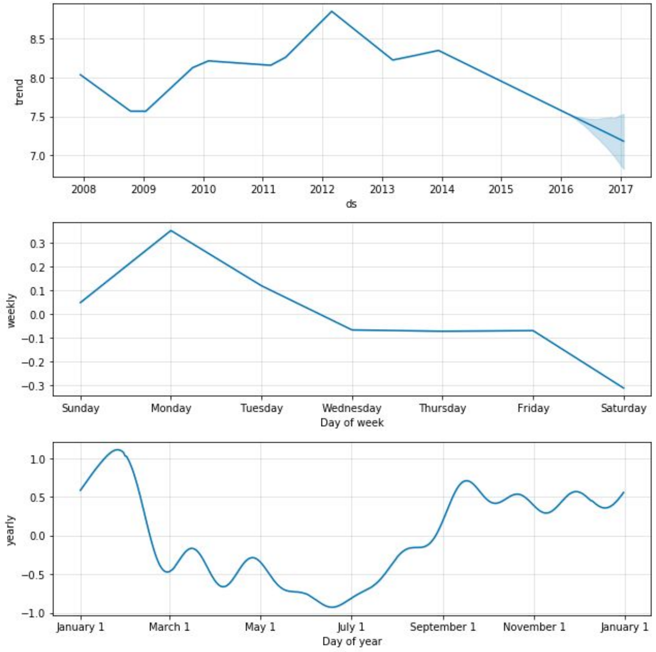
```
Forecast = Model.predict(Future)
```

Visualizing Model Results

- Pass forecast dataframe and Plot by calling the Prophet.plot method.
- Plot forecast components like trend, yearly, weekly seasonality and holidays can be Plot using Prophet.plot_components method.




```
m.plot(forecast);
```




```
m.plot_components(forecast);
```

Thank you

Happy Forecasting!