

Prophet 기본 이해

- 관련 코드 : https://facebook.github.io/prophet/docs/quick_start.html#python-api
(https://facebook.github.io/prophet/docs/quick_start.html#python-api)
- 데이터 출처 :
https://github.com/facebook/prophet/blob/main/examples/example_wp_log_peyton_manning.csv
(https://github.com/facebook/prophet/blob/main/examples/example_wp_log_peyton_manning.csv)
- 데이터 분석 코드
 - [github 코드](https://github.com/LDJWJ/dataAnalysis/blob/main/01_08B_ivospa_stock_Prophet.ipynb) (https://github.com/LDJWJ/dataAnalysis/blob/main/01_08B_ivospa_stock_Prophet.ipynb)
 - [HTML코드](https://ldjwj.github.io/dataAnalysis/01_08B_ivospa_stock_Prophet.html) (https://ldjwj.github.io/dataAnalysis/01_08B_ivospa_stock_Prophet.html)

목차

- [01. 라이브러리 불러오기 및 설치](#)
- [02. Prophet를 활용한 모델 만들기](#)
- [03. 예측 내용 그래프로 표시](#)

01. 라이브러리 불러오기 및 설치

[목차로 이동하기](#)

fbprophet 설치가 안되어 있을 경우,

- pip uninstall pystan
- pip install pystan==2.17.1.0
- pip install fbprophet

In [17]:

```
# 라이브러리 불러오기
import pandas as pd
from fbprophet import Prophet
```

In [18]:

```
# 예제 데이터 셋을 활용
df = pd.read_csv('./data/Prophet/example_wp_log_peyton_manning.csv')
print(df.shape)
df.head()
```

(2905, 2)

Out[18]:

	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

02. Prophet를 활용한 모델 만들기

목차로 이동하기

- Prophet는 Facebook에서 만든 시계열 데이터 모델링을 위한 파이썬 모듈
- URL : <https://facebook.github.io/prophet/> (<https://facebook.github.io/prophet/>)
- Prophet은 머신러닝 기법은 아니며, ARIMA라고 하는 알고리즘 방법론을 응용.
- Prophet는 한층 더 진보적인 방법으로 트렌드와 주기적 특성 뿐 아니라 예외적이고 이벤트와 같은 휴가철 상황까지도 모델링하도록 ARIMA알고리즘을 확장한 것.
- Prophet 객체 생성
- fit() 메소드를 이용한 최적화(학습)을 수행

In [22]:

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

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Out[22]:

<fbprophet.forecaster.Prophet at 0x7feed3fc0280>

In [27]:

```
future = m.make_future_dataframe(periods=365)
print(df.shape)
print(future.shape)
future.tail()
```

```
(2905, 2)
(3270, 1)
```

Out[27]:

	ds
3265	2017-01-15
3266	2017-01-16
3267	2017-01-17
3268	2017-01-18
3269	2017-01-19

- ds의 열이 있는 데이터 프레임에서 예측이 이루어집니다. make_future_dataframe를 사용하여 지정된 일수만큼 미래로 확장되는 적절한 데이터 프레임을 얻을 수 있음.

예측하기

- predict() 메소드는 yhat이라는 이름의 예측 값을 각 행에 할당.
- 여기서 예측 개체는 예측이 포함된 열과 구성 요소 및 불확실성 간격에 대한 열을 포함하는 새 데이터 프레임입니다.

In [28]:

```
forecast = m.predict(future)
print(forecast.shape)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

```
(3270, 19)
```

Out[28]:

	ds	yhat	yhat_lower	yhat_upper
3265	2017-01-15	8.214006	7.512405	8.975372
3266	2017-01-16	8.539078	7.746396	9.279901
3267	2017-01-17	8.326518	7.587072	9.078349
3268	2017-01-18	8.159179	7.465441	8.866700
3269	2017-01-19	8.171119	7.448082	8.900605

03. 예측 내용 그래프로 표시

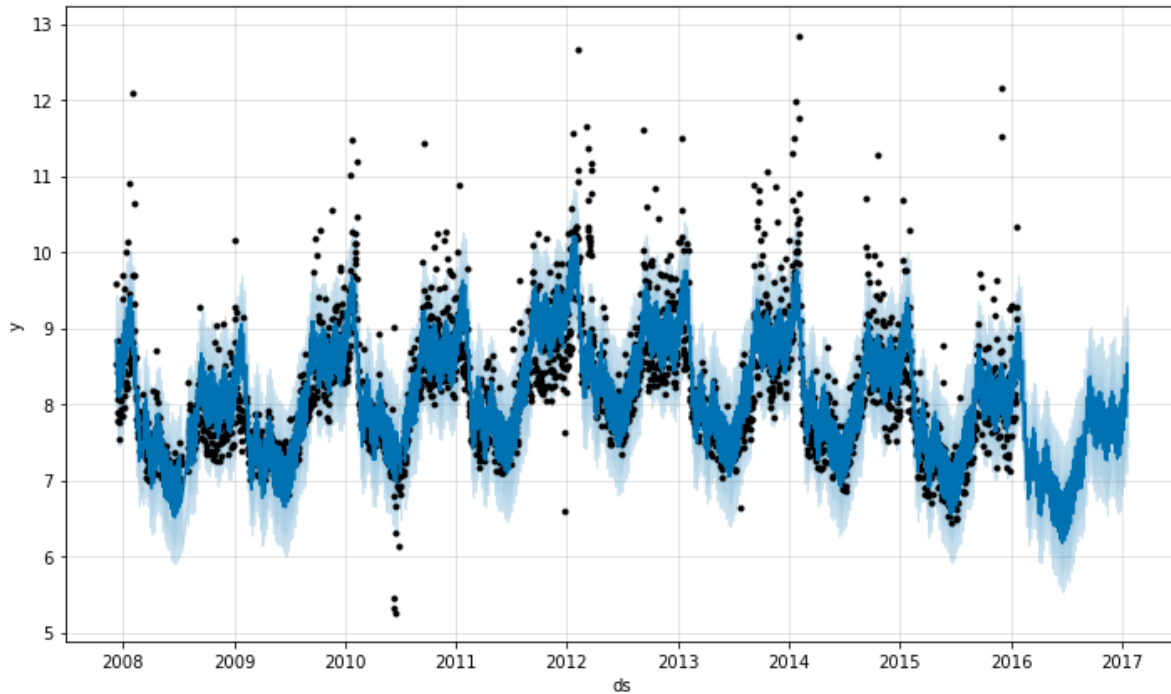
[목차로 이동하기](#)

Prophet.plot

- Prophet.plot이라고 불리는 것으로 예측에 대한 plot를 그릴 수 있다.

In [34]:

```
fig1 = m.plot(forecast)
```

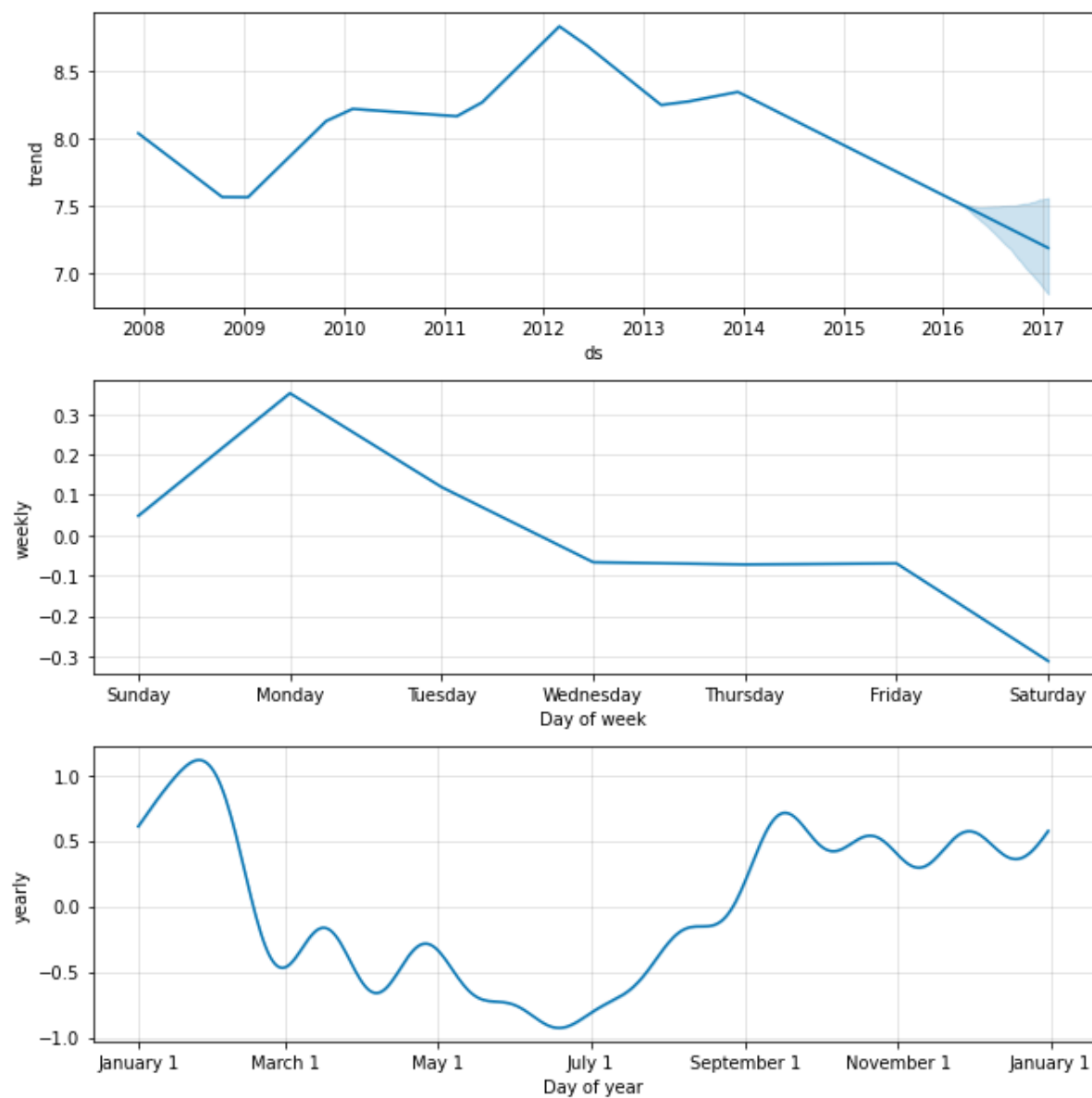


Prophet.plot_components 메소드로 세부 예측 요소 확인

- 미래 예측 요소 확인
 - trend, yearly seasonality에 대한 시계열 데이터
 - weekly seasonality에 대한 시계열 데이터

In [35]:

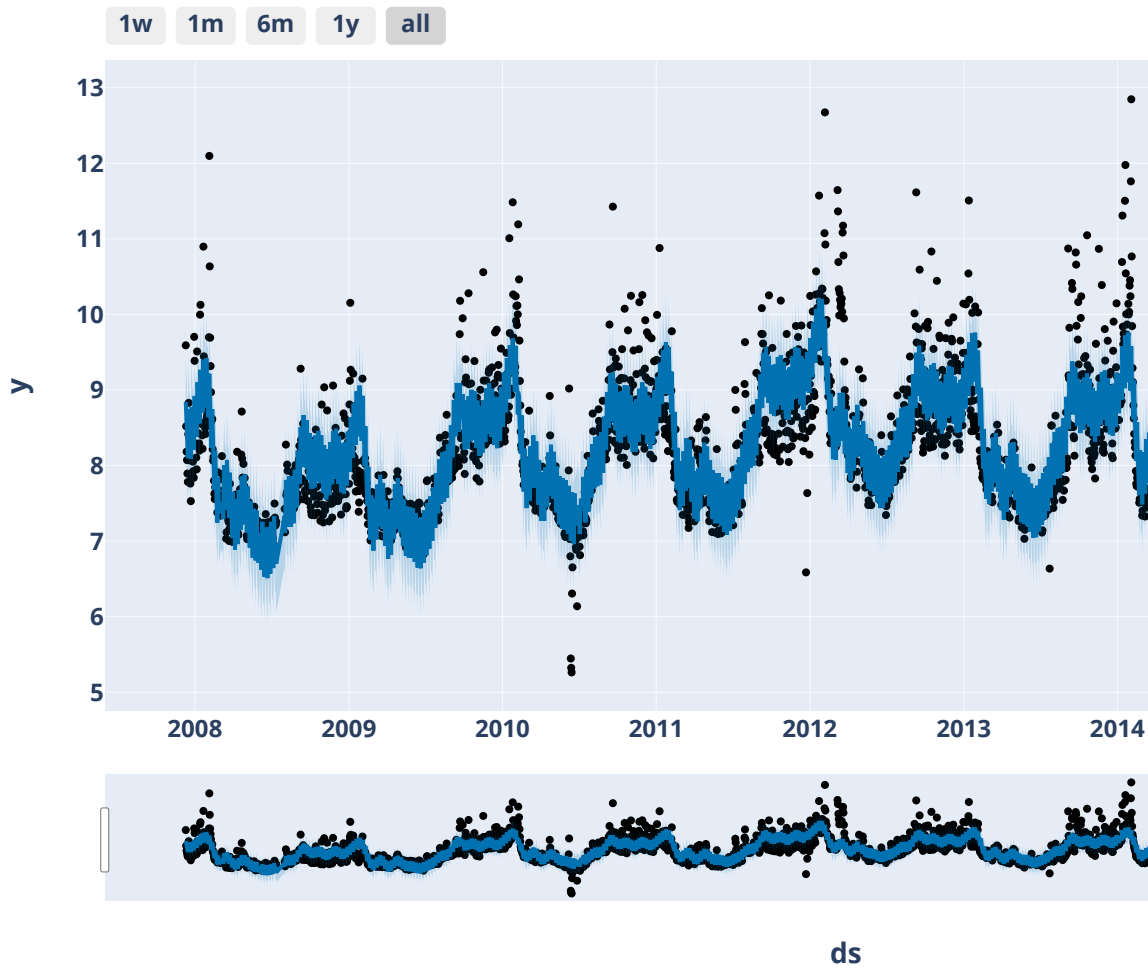
```
fig2 = m.plot_components(forecast)
```



plotly를 활용한 미래 예측 내용 만들기

In [36]:

```
from fbprophet.plot import plot_plotly, plot_components_plotly  
  
plot_plotly(m, forecast)
```



In [37]:

```
plot_components_plotly(m, forecast)
```

