

6.5 Prophet 소개

prophet은 페이스북에서 개발한 시계열 예측 패키지다. ARIMA와 같은 확률론적이고 이론적인 모형이 아니라 몇 가지 경험적 규칙(heuristic rule)을 사용하는 단순 회귀모형이지만 단기적 예측에서는 큰 문제 없이 사용할 수 있다.

설치

prophet을 설치하려면 우선 pystan이라는 패키지를 설치해야 한다. 아나콘다를 이용하여 설치하는 것을 추천한다.

```
conda install pystan
conda install -c conda-forge fbprophet
```

사용법

prophet을 사용하여 다음 시계열 데이터를 예측하는 문제를 풀어보자.

In [1]:

```
import pandas as pd

url = "https://raw.githubusercontent.com/facebook/prophet/master/examples/example_wp_log_peyton_mann.csv"
df = pd.read_csv(url)
df.tail()
```

Out[1]:

	ds	y
2900	2016-01-16	7.817223
2901	2016-01-17	9.273878
2902	2016-01-18	10.333775
2903	2016-01-19	9.125871
2904	2016-01-20	8.891374

In [2]:

```
# prophet의 로그 기능 끄기
import logging
logging.getLogger('fbprophet').setLevel(logging.WARNING)
```

사용법은 간단하다. Prophet 클래스 객체를 만들고 시계열 데이터를 입력으로 fit 메서드를 호출한다.

In [3]:

```
from fbprophet import Prophet

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

ERROR:fbprophet:Importing plotly failed. Interactive plots will not work.

Out[3]:

<fbprophet.forecaster.Prophet at 0x7f3b30092f28>

예측을 하려면 다음 메서드를 사용한다.

- make_future_dataframe : 예측 날짜 구간 생성
- predict : 신뢰 구간을 포함한 예측 실행

In [4]:

```
future = m.make_future_dataframe(periods=365)
future.tail()
```

Out[4]:

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

In [5]:

```
forecast = m.predict(future)
forecast.tail()
```

Out[5]:

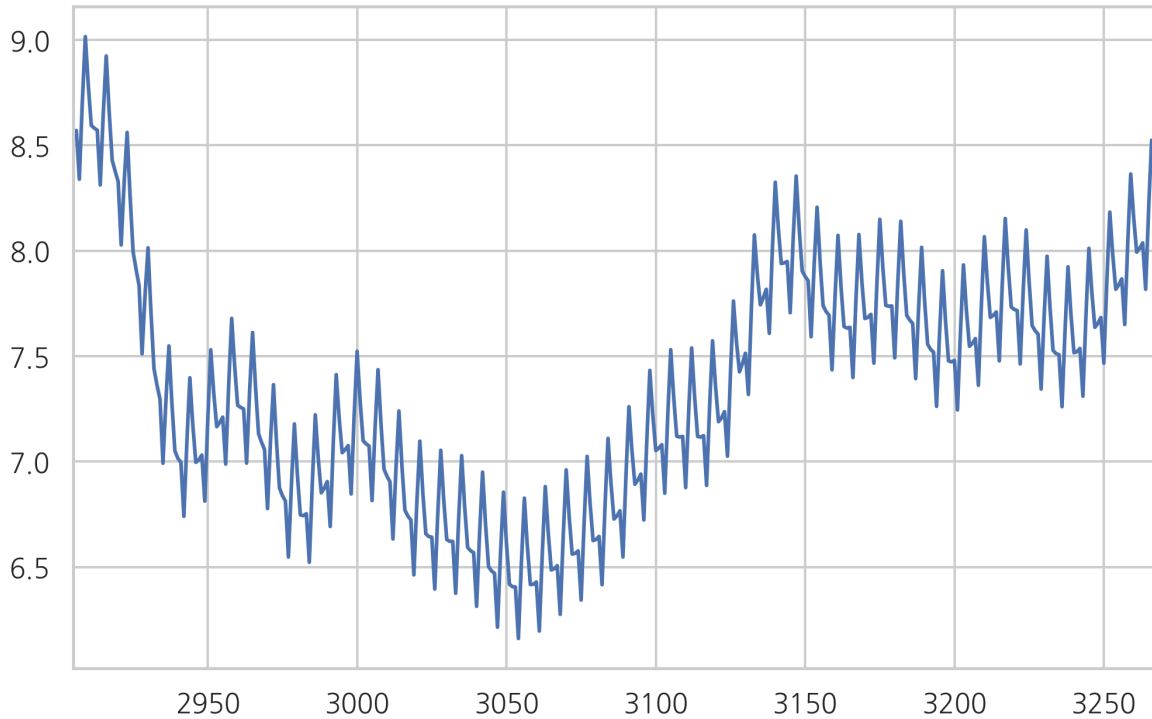
	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms
3265	2017-01-15	7.180847	7.435809	8.948534	6.827810	7.552677	1.018428	
3266	2017-01-16	7.179809	7.755114	9.291769	6.825133	7.552981	1.344435	
3267	2017-01-17	7.178771	7.530683	9.036235	6.822456	7.553219	1.132844	
3268	2017-01-18	7.177733	7.413488	8.845561	6.820041	7.553632	0.966499	
3269	2017-01-19	7.176695	7.425690	8.894338	6.818071	7.553324	0.979396	

In [6]:

```
forecast.iloc[-365:, :].yhat.plot()
```

Out[6]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f3b28f11f28>

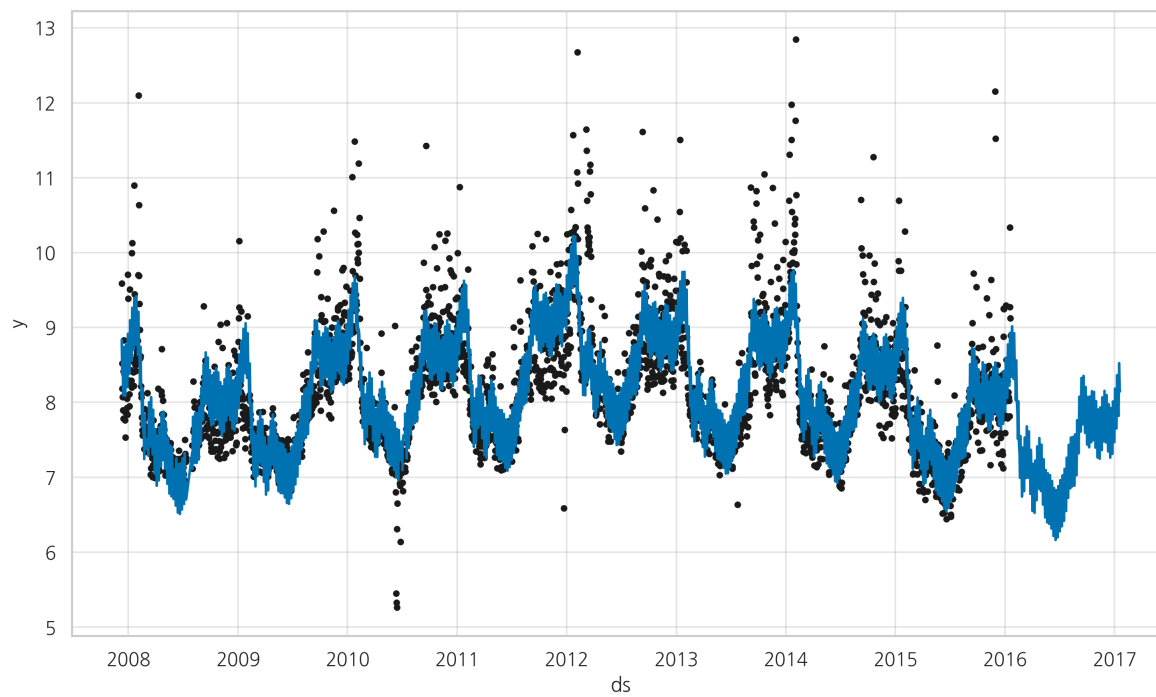


다음 메서드를 사용하면 시계열을 시각화할 수 있다.

- plot : 원래의 시계열 데이터와 예측 데이터
- plot_components : 선형회귀 및 계절성 성분별로 분리

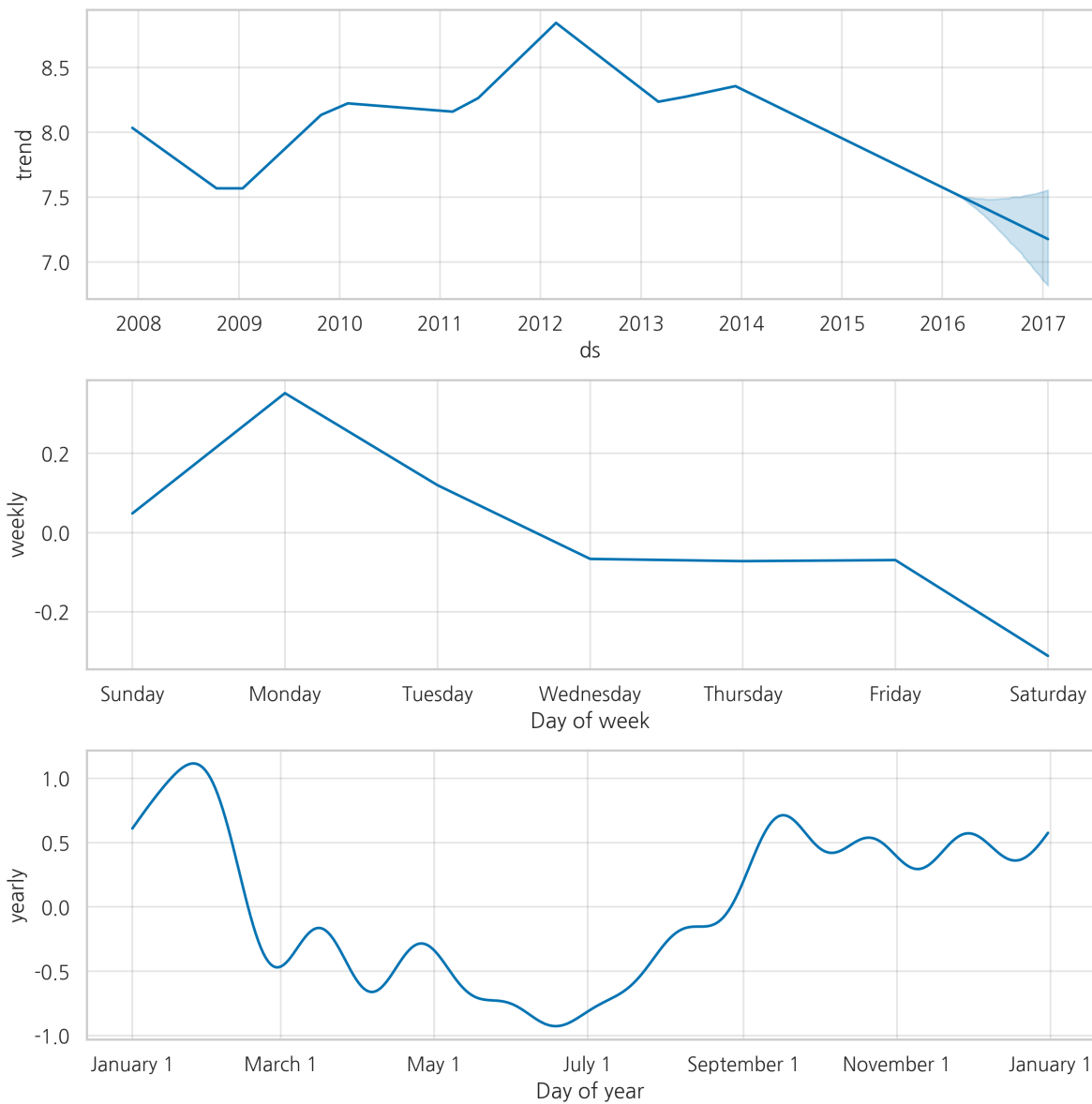
In [7]:

```
fig1 = m.plot(forecast, uncertainty=False)  
plt.show()
```



In [8]:

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



기본 원리

Prophet은 다음 순서로 시계열에 대한 회귀분석 모형을 만든다.

- 시간 데이터의 각종 특징을 임베딩해서 계절성 추정을 한다.
- 나머지 데이터는 구간별 선형회귀(piecewise linear regression) 분석을 한다.

선형 회귀분석은 전체 시계열의 앞 80%부분을 25개의 구간으로 나누어 실시한다. 구간 구분점(change point)는 changepoints 속성에 있다.

In [9]:

```
m.changepoints
```

Out[9]:

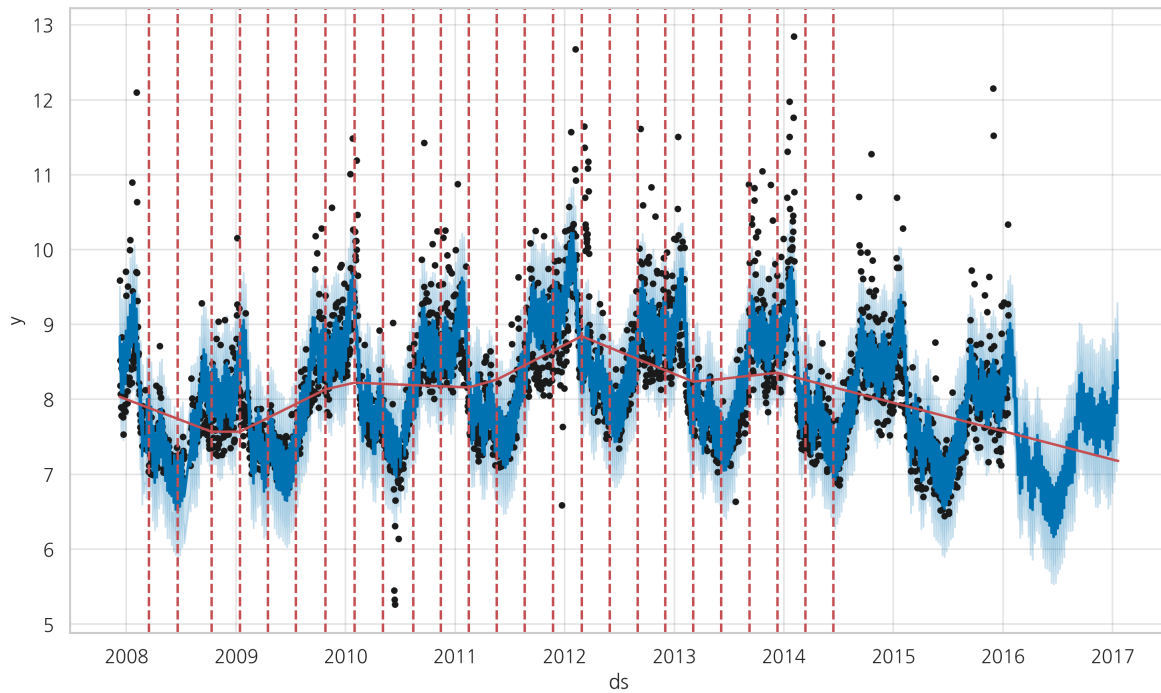
```
93      2008-03-17
186     2008-06-20
279     2008-10-11
372     2009-01-14
465     2009-04-17
558     2009-07-19
650     2009-10-26
743     2010-01-31
836     2010-05-06
929     2010-08-14
1022    2010-11-15
1115    2011-02-16
1208    2011-05-20
1301    2011-08-21
1394    2011-11-24
1487    2012-02-27
1580    2012-05-31
1673    2012-09-01
1765    2012-12-02
1858    2013-03-05
1951    2013-06-06
2044    2013-09-08
2137    2013-12-10
2230    2014-03-14
2323    2014-06-15
Name: ds, dtype: datetime64[ns]
```

선형회귀모형의 기울기와 y-절편은 param 속성의 delta 와 beta 값에 있다.

In [10]:

```
from fbprophet.plot import add_changepoints_to_plot

fig = m.plot(forecast)
a = add_changepoints_to_plot(fig.gca(), m, forecast, threshold=0)
plt.show()
```



신뢰구간

prophet은 마지막 20% 데이터에 대해 선형회귀분석을 하고 신뢰구간을 계산할 수 있다. 신뢰구간은 구간별 기울기의 분포를 라플라스 확률분포로 추정한 후 기울기 변화를 가정하는 방법으로 신뢰구간을 계산한다.

In [11]:

```
sns.distplot(m.params["delta"], kde=False, fit=sp.stats.laplace)  
plt.show()
```

