Example

March 5, 2018

1 Example

Import the necessary packages.

```
In [1]: import pandas as pd
        import numpy as np
        from fbprophet import Prophet

In [2]: df = pd.read_csv('./data/example_wp_peyton_manning.csv')
        df
```

Out[2]:

	ds	y
0	2007-12-10	14629
1	2007-12-11	5012
2	2007-12-12	3582
3	2007-12-13	3205
4	2007-12-14	2680
5	2007-12-15	2401
6	2007-12-16	4510
7	2007-12-17	6831
8	2007-12-18	4370
9	2007-12-19	3196
10	2007-12-20	2642
11	2007-12-21	2349
12	2007-12-22	1862
13	2007-12-23	4382
14	2007-12-24	5542
15	2007-12-25	2572
16	2007-12-26	2576
17	2007-12-27	3140
18	2007-12-28	2770
19	2007-12-29	2536
20	2007-12-30	16369
21	2007-12-31	11920
22	2008-01-01	3999
23	2008-01-02	4604
24	2008-01-03	3874

```
25    2008-01-04   3317
26    2008-01-05   2836
27    2008-01-06  13484
28    2008-01-07   6952
29    2008-01-08   4585
...          ...    ...
2875  2015-12-22   3329
2876  2015-12-23   2381
2877  2015-12-24   1552
2878  2015-12-25   1483
2879  2015-12-26   1237
2880  2015-12-27   7178
2881  2015-12-28  10411
2882  2015-12-29  10383
2883  2015-12-30   4409
2884  2015-12-31   2995
2885  2016-01-01   1977
2886  2016-01-02   2503
2887  2016-01-03   3825
2888  2016-01-04  10885
2889  2016-01-05   5262
2890  2016-01-06   4241
2891  2016-01-07   4007
2892  2016-01-08   3985
2893  2016-01-09   2406
2894  2016-01-10   3951
2895  2016-01-11   4773
2896  2016-01-12   3412
2897  2016-01-13   3188
2898  2016-01-14   3052
2899  2016-01-15   3047
2900  2016-01-16   2483
2901  2016-01-17  10656
2902  2016-01-18  30754
2903  2016-01-19   9190
2904  2016-01-20   7269

[2905 rows x 2 columns]
```

1.0.1 Applying log transform

why log transform

```
In [10]: df['y'] = np.log(df['y'])
```

Remember, the column names of the input data to prophet must be ds and y only. Otherwise, call to the fit method throws a KeyError.

1.0.2 Creating the model

The model is created using the prophet class available in the package. The class takes several arguments which will be discussed later. The arguments have default values, which means we can use default values for the arguments by passing nothing like below.

```
In [12]: model = Prophet()
```

1.0.3 Fitting the model to the data

This can be done using the fit function as shown below.

```
In [13]: model.fit(df)
```

```
INFO:fbprophet.forecaster:Disabling daily seasonality. Run prophet with daily_seasonality=True
```

```
Out[13]: <fbprophet.forecaster.Prophet at 0x27df6277278>
```

1.0.4 Preparing for forecasting

The prediction is done on a dataframe with column ds (and not anything else) that has the dates for which prediction is to be made. This means we first create a dataframe with date column that contains the future dates. This dataframe can be created by a helper function, implemented in the Prophet class, called make_future_dataframe.

The make_future_dataframe function by default includes the dates from the history which can be changed using the parameted include_history=False.

The periods argument to the make_future_dataframe function is the number of days in future to predict. It creates the passed number of dates after the last date in dataframe.

The make_future_dataframe has another argument called freq which specifies the frequency at which we intend to predict. It is important to make the same frequency forecasts as the frequency of the data that is being fit to the prophet, otherwise the seasonality components kick in and often result in overfitting or unexpected trends.

```
In [14]: future = model.make_future_dataframe(periods=365)
```

1.0.5 Forecasting

The forecast results can be obtained using the predict method availble in the Prophet class. The predict method will assign each row (date) in the created dataframe a predicted value which it names yhat. Along with the predicted value it assigns It assigns trends, seasonalities, daily, weekly, yearly trends and their confidence intervals.

```
In [15]: forecast = model.predict(future)
         forecast
```

```
Out[15]:            ds     trend  trend_lower  trend_upper  yhat_lower  yhat_upper  \
         0  2007-12-10  8.043514     8.043514     8.043514    8.207783    9.443553
         1  2007-12-11  8.041958     8.041958     8.041958    8.002048    9.204919
         2  2007-12-12  8.040401     8.040401     8.040401    7.773076    9.036145
```

3	2007-12-13	8.038845	8.038845	8.038845	7.730721	8.964983
4	2007-12-14	8.037288	8.037288	8.037288	7.784848	8.927975
5	2007-12-15	8.035732	8.035732	8.035732	7.491713	8.689742
6	2007-12-16	8.034175	8.034175	8.034175	7.807520	9.047194
7	2007-12-17	8.032619	8.032619	8.032619	8.117969	9.394405
8	2007-12-18	8.031062	8.031062	8.031062	7.864373	9.105911
9	2007-12-19	8.029506	8.029506	8.029506	7.672630	8.910727
10	2007-12-20	8.027949	8.027949	8.027949	7.661534	8.930092
11	2007-12-21	8.026393	8.026393	8.026393	7.718720	8.950912
12	2007-12-22	8.024836	8.024836	8.024836	7.485406	8.726272
13	2007-12-23	8.023280	8.023280	8.023280	7.842569	9.060560
14	2007-12-24	8.021723	8.021723	8.021723	8.120958	9.417184
15	2007-12-25	8.020167	8.020167	8.020167	7.924362	9.170436
16	2007-12-26	8.018610	8.018610	8.018610	7.778855	9.026917
17	2007-12-27	8.017054	8.017054	8.017054	7.799826	9.069458
18	2007-12-28	8.015497	8.015497	8.015497	7.822797	9.017535
19	2007-12-29	8.013941	8.013941	8.013941	7.558477	8.791942
20	2007-12-30	8.012384	8.012384	8.012384	7.984691	9.195876
21	2007-12-31	8.010828	8.010828	8.010828	8.305328	9.494093
22	2008-01-01	8.009271	8.009271	8.009271	8.095527	9.321810
23	2008-01-02	8.007715	8.007715	8.007715	7.971941	9.235852
24	2008-01-03	8.006158	8.006158	8.006158	7.923533	9.220777
25	2008-01-04	8.004602	8.004602	8.004602	7.962121	9.262927
26	2008-01-05	8.003045	8.003045	8.003045	7.716725	9.013984
27	2008-01-06	8.001489	8.001489	8.001489	8.124599	9.397655
28	2008-01-07	7.999932	7.999932	7.999932	8.484165	9.736069
29	2008-01-08	7.998376	7.998376	7.998376	8.298025	9.552042
...	...	...	...	...	...	...
3240	2016-12-21	7.214088	6.911252	7.554922	6.774701	8.291942
3241	2016-12-22	7.213060	6.909230	7.556200	6.812654	8.309879
3242	2016-12-23	7.212032	6.907207	7.556352	6.827891	8.252538
3243	2016-12-24	7.211004	6.905185	7.556427	6.619991	8.010000
3244	2016-12-25	7.209976	6.901616	7.556503	6.945829	8.424191
3245	2016-12-26	7.208948	6.898836	7.556579	7.316433	8.726073
3246	2016-12-27	7.207920	6.897101	7.556655	7.095411	8.470008
3247	2016-12-28	7.206893	6.895461	7.556730	6.924862	8.389463
3248	2016-12-29	7.205865	6.893821	7.556806	6.983011	8.398453
3249	2016-12-30	7.204837	6.892185	7.556870	6.989189	8.409699
3250	2016-12-31	7.203809	6.890601	7.556915	6.721940	8.195197
3251	2017-01-01	7.202781	6.889018	7.556959	7.105776	8.563659
3252	2017-01-02	7.201753	6.887434	7.557008	7.487681	8.938459
3253	2017-01-03	7.200725	6.884858	7.557078	7.256288	8.703774
3254	2017-01-04	7.199697	6.882199	7.557148	7.076200	8.554813
3255	2017-01-05	7.198669	6.879540	7.557218	7.127395	8.596500
3256	2017-01-06	7.197641	6.876881	7.557759	7.186440	8.616338
3257	2017-01-07	7.196613	6.874222	7.558639	6.962783	8.355592
3258	2017-01-08	7.195585	6.870921	7.559693	7.284578	8.757045
3259	2017-01-09	7.194558	6.865842	7.561015	7.662835	9.103429

3260	2017-01-10	7.193530	6.862432	7.562337	7.506854	8.855329
3261	2017-01-11	7.192502	6.860649	7.562928	7.287543	8.691112
3262	2017-01-12	7.191474	6.858950	7.563314	7.296851	8.730155
3263	2017-01-13	7.190446	6.857223	7.564086	7.306958	8.814020
3264	2017-01-14	7.189418	6.855095	7.564972	7.130686	8.592075
3265	2017-01-15	7.188390	6.852109	7.565859	7.487177	8.917324
3266	2017-01-16	7.187362	6.849283	7.566819	7.769156	9.246764
3267	2017-01-17	7.186334	6.847031	7.567997	7.587141	9.080322
3268	2017-01-18	7.185306	6.844768	7.569012	7.439610	8.921231
3269	2017-01-19	7.184278	6.842017	7.571704	7.445648	8.884512

	seasonal	seasonal_lower	seasonal_upper	seasonalities	\
0	0.802754	0.802754	0.802754	0.802754	
1	0.552779	0.552779	0.552779	0.552779	
2	0.350112	0.350112	0.350112	0.350112	
3	0.329551	0.329551	0.329551	0.329551	
4	0.319020	0.319020	0.319020	0.319020	
5	0.065846	0.065846	0.065846	0.065846	
6	0.417234	0.417234	0.417234	0.417234	
7	0.715358	0.715358	0.715358	0.715358	
8	0.479668	0.479668	0.479668	0.479668	
9	0.293350	0.293350	0.293350	0.293350	
10	0.290807	0.290807	0.290807	0.290807	
11	0.299546	0.299546	0.299546	0.299546	
12	0.066466	0.066466	0.066466	0.066466	
13	0.438342	0.438342	0.438342	0.438342	
14	0.756932	0.756932	0.756932	0.756932	
15	0.541294	0.541294	0.541294	0.541294	
16	0.374253	0.374253	0.374253	0.374253	
17	0.389899	0.389899	0.389899	0.389899	
18	0.415473	0.415473	0.415473	0.415473	
19	0.197660	0.197660	0.197660	0.197660	
20	0.583083	0.583083	0.583083	0.583083	
21	0.913405	0.913405	0.913405	0.913405	
22	0.707647	0.707647	0.707647	0.707647	
23	0.548652	0.548652	0.548652	0.548652	
24	0.570576	0.570576	0.570576	0.570576	
25	0.600771	0.600771	0.600771	0.600771	
26	0.386066	0.386066	0.386066	0.386066	
27	0.773256	0.773256	0.773256	0.773256	
28	1.104190	1.104190	1.104190	1.104190	
29	0.898080	0.898080	0.898080	0.898080	
...	...	...	...	...	
3240	0.308966	0.308966	0.308966	0.308966	
3241	0.314584	0.314584	0.314584	0.314584	
3242	0.331168	0.331168	0.331168	0.331168	
3243	0.105483	0.105483	0.105483	0.105483	
3244	0.484195	0.484195	0.484195	0.484195	

3245	0.808974	0.808974	0.808974	0.808974
3246	0.598813	0.598813	0.598813	0.598813
3247	0.436499	0.436499	0.436499	0.436499
3248	0.456105	0.456105	0.456105	0.456105
3249	0.484881	0.484881	0.484881	0.484881
3250	0.269539	0.269539	0.269539	0.269539
3251	0.656752	0.656752	0.656752	0.656752
3252	0.988242	0.988242	0.988242	0.988242
3253	0.783104	0.783104	0.783104	0.783104
3254	0.624261	0.624261	0.624261	0.624261
3255	0.645953	0.645953	0.645953	0.645953
3256	0.675610	0.675610	0.675610	0.675610
3257	0.460140	0.460140	0.460140	0.460140
3258	0.846402	0.846402	0.846402	0.846402
3259	1.176297	1.176297	1.176297	1.176297
3260	0.969074	0.969074	0.969074	0.969074
3261	0.807782	0.807782	0.807782	0.807782
3262	0.826749	0.826749	0.826749	0.826749
3263	0.853451	0.853451	0.853451	0.853451
3264	0.634792	0.634792	0.634792	0.634792
3265	1.017585	1.017585	1.017585	1.017585
3266	1.343631	1.343631	1.343631	1.343631
3267	1.132037	1.132037	1.132037	1.132037
3268	0.965667	0.965667	0.965667	0.965667
3269	0.978629	0.978629	0.978629	0.978629

	seasonalities_lower	seasonalities_upper	weekly	weekly_lower	\
0	0.802754	0.802754	0.352301	0.352301	
1	0.552779	0.552779	0.119628	0.119628	
2	0.350112	0.350112	-0.066672	-0.066672	
3	0.329551	0.329551	-0.072248	-0.072248	
4	0.319020	0.319020	-0.069582	-0.069582	
5	0.065846	0.065846	-0.311706	-0.311706	
6	0.417234	0.417234	0.048279	0.048279	
7	0.715358	0.715358	0.352301	0.352301	
8	0.479668	0.479668	0.119628	0.119628	
9	0.293350	0.293350	-0.066672	-0.066672	
10	0.290807	0.290807	-0.072248	-0.072248	
11	0.299546	0.299546	-0.069582	-0.069582	
12	0.066466	0.066466	-0.311706	-0.311706	
13	0.438342	0.438342	0.048279	0.048279	
14	0.756932	0.756932	0.352301	0.352301	
15	0.541294	0.541294	0.119628	0.119628	
16	0.374253	0.374253	-0.066672	-0.066672	
17	0.389899	0.389899	-0.072248	-0.072248	
18	0.415473	0.415473	-0.069582	-0.069582	
19	0.197660	0.197660	-0.311706	-0.311706	
20	0.583083	0.583083	0.048279	0.048279	

21	0.913405	0.913405	0.352301	0.352301	
22	0.707647	0.707647	0.119628	0.119628	
23	0.548652	0.548652	-0.066672	-0.066672	
24	0.570576	0.570576	-0.072248	-0.072248	
25	0.600771	0.600771	-0.069582	-0.069582	
26	0.386066	0.386066	-0.311706	-0.311706	
27	0.773256	0.773256	0.048279	0.048279	
28	1.104190	1.104190	0.352301	0.352301	
29	0.898080	0.898080	0.119628	0.119628	
...	...	...	...	...	
3240	0.308966	0.308966	-0.066672	-0.066672	
3241	0.314584	0.314584	-0.072248	-0.072248	
3242	0.331168	0.331168	-0.069582	-0.069582	
3243	0.105483	0.105483	-0.311706	-0.311706	
3244	0.484195	0.484195	0.048279	0.048279	
3245	0.808974	0.808974	0.352301	0.352301	
3246	0.598813	0.598813	0.119628	0.119628	
3247	0.436499	0.436499	-0.066672	-0.066672	
3248	0.456105	0.456105	-0.072248	-0.072248	
3249	0.484881	0.484881	-0.069582	-0.069582	
3250	0.269539	0.269539	-0.311706	-0.311706	
3251	0.656752	0.656752	0.048279	0.048279	
3252	0.988242	0.988242	0.352301	0.352301	
3253	0.783104	0.783104	0.119628	0.119628	
3254	0.624261	0.624261	-0.066672	-0.066672	
3255	0.645953	0.645953	-0.072248	-0.072248	
3256	0.675610	0.675610	-0.069582	-0.069582	
3257	0.460140	0.460140	-0.311706	-0.311706	
3258	0.846402	0.846402	0.048279	0.048279	
3259	1.176297	1.176297	0.352301	0.352301	
3260	0.969074	0.969074	0.119628	0.119628	
3261	0.807782	0.807782	-0.066672	-0.066672	
3262	0.826749	0.826749	-0.072248	-0.072248	
3263	0.853451	0.853451	-0.069582	-0.069582	
3264	0.634792	0.634792	-0.311706	-0.311706	
3265	1.017585	1.017585	0.048279	0.048279	
3266	1.343631	1.343631	0.352301	0.352301	
3267	1.132037	1.132037	0.119628	0.119628	
3268	0.965667	0.965667	-0.066672	-0.066672	
3269	0.978629	0.978629	-0.072248	-0.072248	

	weekly_upper	yearly	yearly_lower	yearly_upper	yhat
0	0.352301	0.450453	0.450453	0.450453	8.846269
1	0.119628	0.433151	0.433151	0.433151	8.594737
2	-0.066672	0.416784	0.416784	0.416784	8.390513
3	-0.072248	0.401799	0.401799	0.401799	8.368396
4	-0.069582	0.388602	0.388602	0.388602	8.356309
5	-0.311706	0.377552	0.377552	0.377552	8.101578

6	0.048279	0.368955	0.368955	0.368955	8.451410
7	0.352301	0.363057	0.363057	0.363057	8.747977
8	0.119628	0.360040	0.360040	0.360040	8.510731
9	-0.066672	0.360022	0.360022	0.360022	8.322856
10	-0.072248	0.363055	0.363055	0.363055	8.318756
11	-0.069582	0.369128	0.369128	0.369128	8.325939
12	-0.311706	0.378172	0.378172	0.378172	8.091302
13	0.048279	0.390063	0.390063	0.390063	8.461622
14	0.352301	0.404631	0.404631	0.404631	8.778656
15	0.119628	0.421666	0.421666	0.421666	8.561461
16	-0.066672	0.440926	0.440926	0.440926	8.392864
17	-0.072248	0.462148	0.462148	0.462148	8.406953
18	-0.069582	0.485055	0.485055	0.485055	8.430970
19	-0.311706	0.509366	0.509366	0.509366	8.211600
20	0.048279	0.534804	0.534804	0.534804	8.595467
21	0.352301	0.561104	0.561104	0.561104	8.924233
22	0.119628	0.588019	0.588019	0.588019	8.716918
23	-0.066672	0.615325	0.615325	0.615325	8.556367
24	-0.072248	0.642825	0.642825	0.642825	8.576735
25	-0.069582	0.670353	0.670353	0.670353	8.605373
26	-0.311706	0.697772	0.697772	0.697772	8.389111
27	0.048279	0.724977	0.724977	0.724977	8.774745
28	0.352301	0.751889	0.751889	0.751889	9.104122
29	0.119628	0.778452	0.778452	0.778452	8.896456
...	...	...	...	...	...
3240	-0.066672	0.375638	0.375638	0.375638	7.523054
3241	-0.072248	0.386832	0.386832	0.386832	7.527644
3242	-0.069582	0.400749	0.400749	0.400749	7.543200
3243	-0.311706	0.417189	0.417189	0.417189	7.316487
3244	0.048279	0.435916	0.435916	0.435916	7.694172
3245	0.352301	0.456673	0.456673	0.456673	8.017923
3246	0.119628	0.479185	0.479185	0.479185	7.806734
3247	-0.066672	0.503172	0.503172	0.503172	7.643392
3248	-0.072248	0.528354	0.528354	0.528354	7.661970
3249	-0.069582	0.554462	0.554462	0.554462	7.689717
3250	-0.311706	0.581245	0.581245	0.581245	7.473348
3251	0.048279	0.608473	0.608473	0.608473	7.859533
3252	0.352301	0.635941	0.635941	0.635941	8.189995
3253	0.119628	0.663476	0.663476	0.663476	7.983829
3254	-0.066672	0.690934	0.690934	0.690934	7.823959
3255	-0.072248	0.718201	0.718201	0.718201	7.844622
3256	-0.069582	0.745192	0.745192	0.745192	7.873252
3257	-0.311706	0.771846	0.771846	0.771846	7.656753
3258	0.048279	0.798123	0.798123	0.798123	8.041987
3259	0.352301	0.823996	0.823996	0.823996	8.370854
3260	0.119628	0.849446	0.849446	0.849446	8.162603
3261	-0.066672	0.874455	0.874455	0.874455	8.000284
3262	-0.072248	0.898997	0.898997	0.898997	8.018222

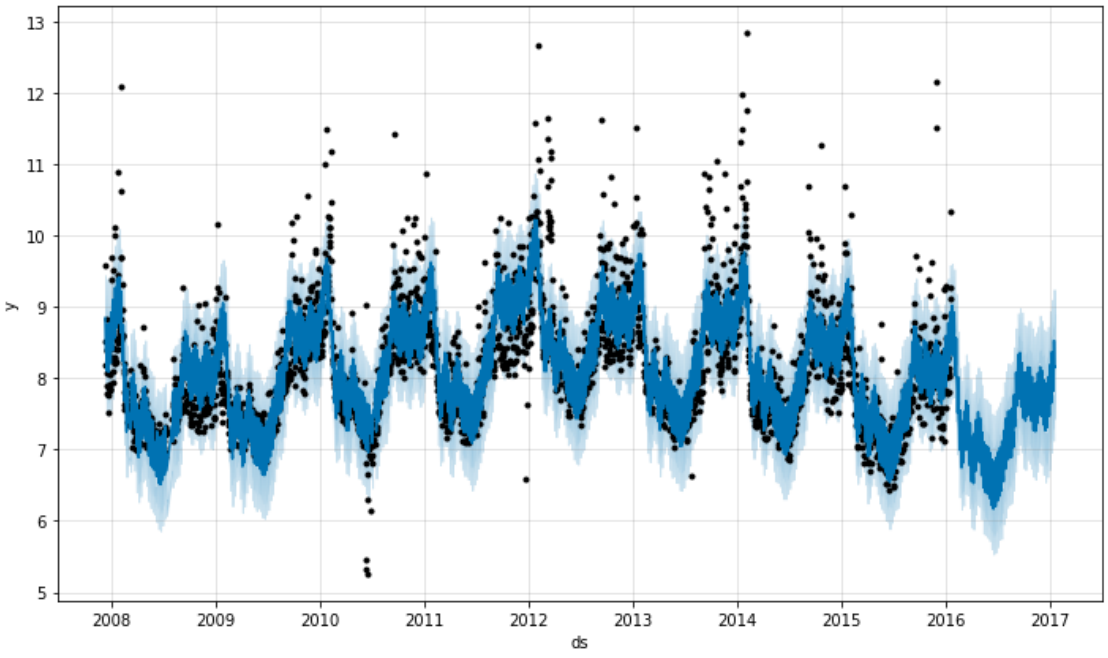
```
3263     -0.069582  0.923032       0.923032       0.923032  8.043897
3264     -0.311706  0.946499       0.946499       0.946499  7.824211
3265      0.048279  0.969306       0.969306       0.969306  8.205975
3266      0.352301  0.991330       0.991330       0.991330  8.530993
3267      0.119628  1.012409       1.012409       1.012409  8.318371
3268     -0.066672  1.032340       1.032340       1.032340  8.150974
3269     -0.072248  1.050878       1.050878       1.050878  8.162908

[3270 rows x 19 columns]
```

1.0.6 Plotting

There is a plot function implemented in the Prophet class which takes input the predicted dataframe(in this case the forecast dataframe).

```
In [16]: model.plot(forecast)
```

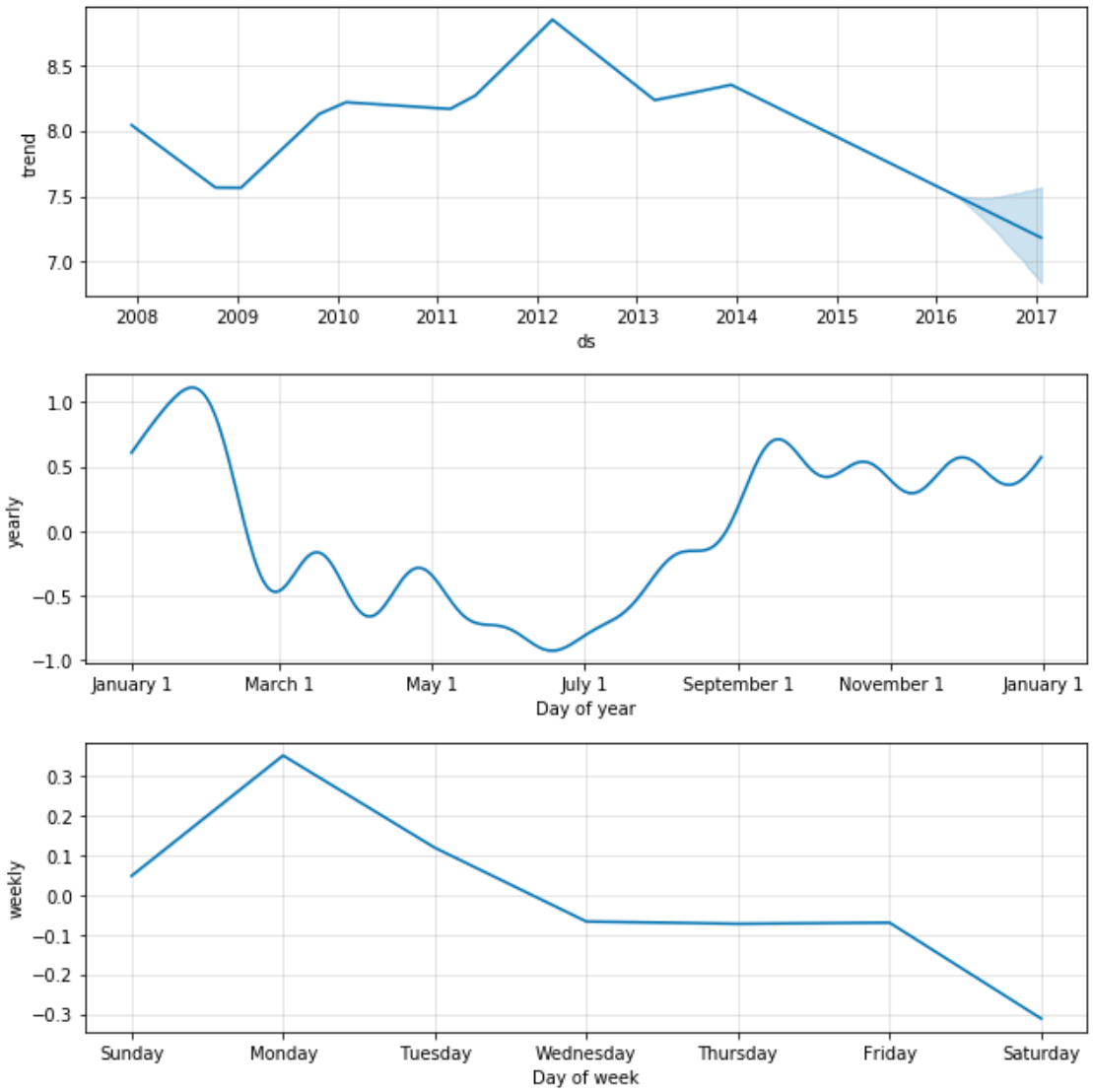
Out[16]:



Prophet provides another provision to plot the individual components using the method plot_components().

```
In [17]: model.plot_components(forecast)
```

Out[17]:


trend
8.5
8.0
7.5
7.0
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
ds
yearly
1.0
0.5
0.0
−0.5
−1.0
January 1
March 1
May 1
July 1
September 1
November 1
January 1
Day of year
weekly
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week