In [2]:

```
import warnings
warnings.filterwarnings('ignore')

import os
from datetime import date
import dateutil.relativedelta

import pandas as pd                                    # panda's nickname is

import numpy as np                                     # numpy as np

from pandas import DataFrame, Series                   # for convenience

import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd
import fbprophet
from fbprophet import Prophet

import statsmodels
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.tsa.seasonal import seasonal_decompose
from datetime import datetime

import tensorflow as tf
from tensorflow.contrib.timeseries.python.timeseries import  NumpyReader
import tensorflow.python.util.deprecation as deprecation
deprecation._PRINT_DEPRECATION_WARNINGS = False
import time
```

In [3]:
```
#load data.
df = pd.read_csv('issues.csv')
df
```

Out[3]:

	issue_number	OriginationPhase	DetectionPhase	Category	Priority	Status
0	1	Requirements	Coding	Bug	Critical	Approved
1	2	Design	Testing	Enhancement	High	Approved
2	3	Requirements	Design	Inquiry	Low	Rejected
3	4	Testing	Field	Bug	High	Completed
4	5	Documentation	Field	Enhancement	Major	pendingReview
...	...	...	...	...	...	...
1995	1996	Design	Coding	Inquiry	Low	Rejected
1996	1997	Testing	Field	Bug	Medium	Completed
1997	1998	Documentation	Field	Enhancement	Major	Rejected
1998	1999	Design	Coding	Inquiry	High	pendingReview
1999	2000	Design	Coding	Inquiry	High	Completed

2000 rows × 9 columns

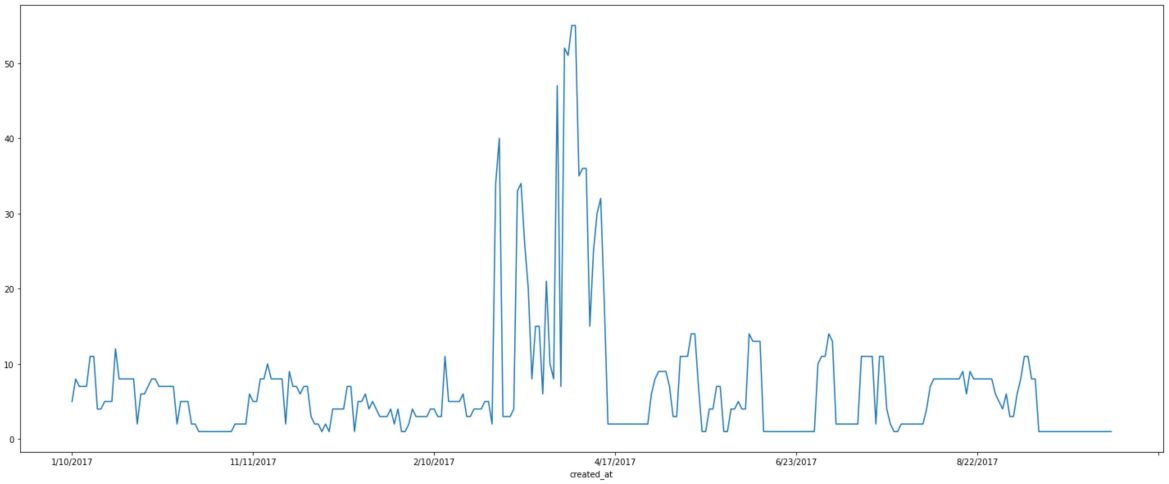
In [4]:
```
df.tail()
```

Out[4]:

	issue_number	OriginationPhase	DetectionPhase	Category	Priority	Status
1995	1996	Design	Coding	Inquiry	Low	Rejected
1996	1997	Testing	Field	Bug	Medium	Completed
1997	1998	Documentation	Field	Enhancement	Major	Rejected
1998	1999	Design	Coding	Inquiry	High	pendingReview
1999	2000	Design	Coding	Inquiry	High	Completed

In [5]:
```
DailyIssue = df.groupby(['created_at']).created_at.count()
DailyIssue.plot(figsize= (25, 10))
#Data = DataFrame(DailyIssue)
```

Out[5]: <AxesSubplot:xlabel='created_at'>



In [6]:
```
df1 = df.groupby(['created_at'], as_index = False).count()
dataFrame = df1[['created_at','issue_number']]
dataFrame.columns = ['ds', 'y']
dataFrame
dataFrame.to_csv (r'github_data.csv', index = None, header=True)
```

In [7]:
```
#facebook prophet
dataFrame = pd.read_csv('github_data.csv')
m = Prophet()
m.fit(dataFrame)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_season
ality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonal
ity=True to override this.
```

Out[7]: <fbprophet.forecaster.Prophet at 0x293c9419e88>

In [8]:
```
#facebook
#make predict
future = m.make_future_dataframe(periods=365)
future.tail()
```

Out[8]:

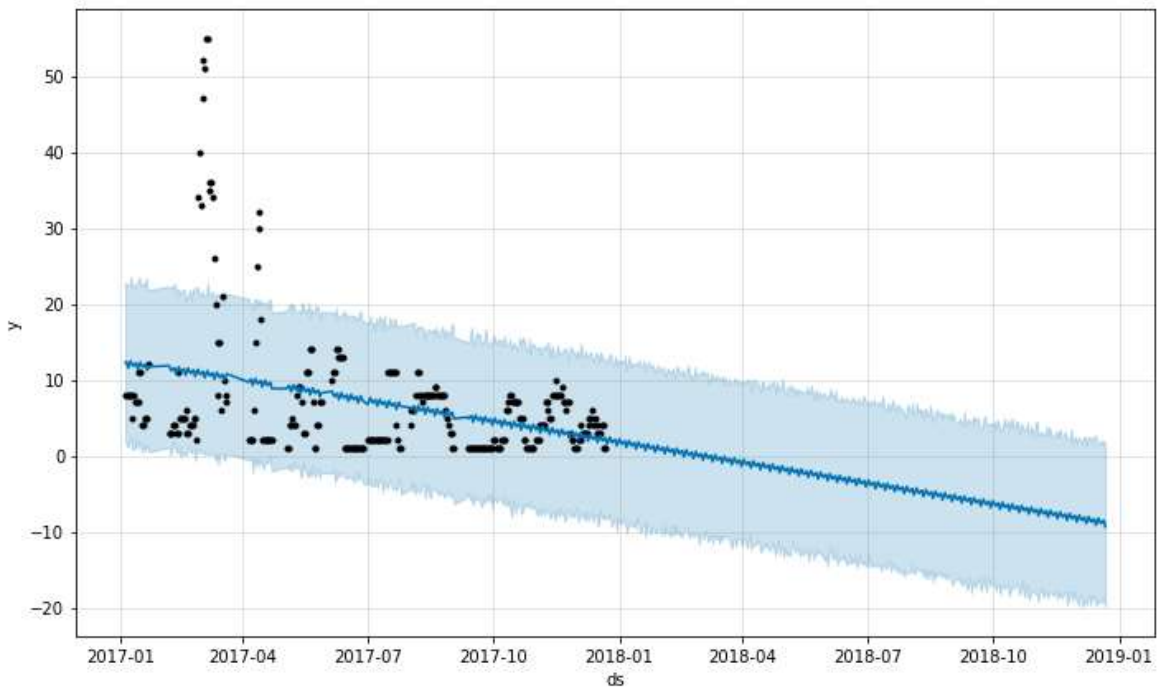
	ds
648	2018-12-18
649	2018-12-19
650	2018-12-20
651	2018-12-21
652	2018-12-22

In [9]:
```
#facebook prophet
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

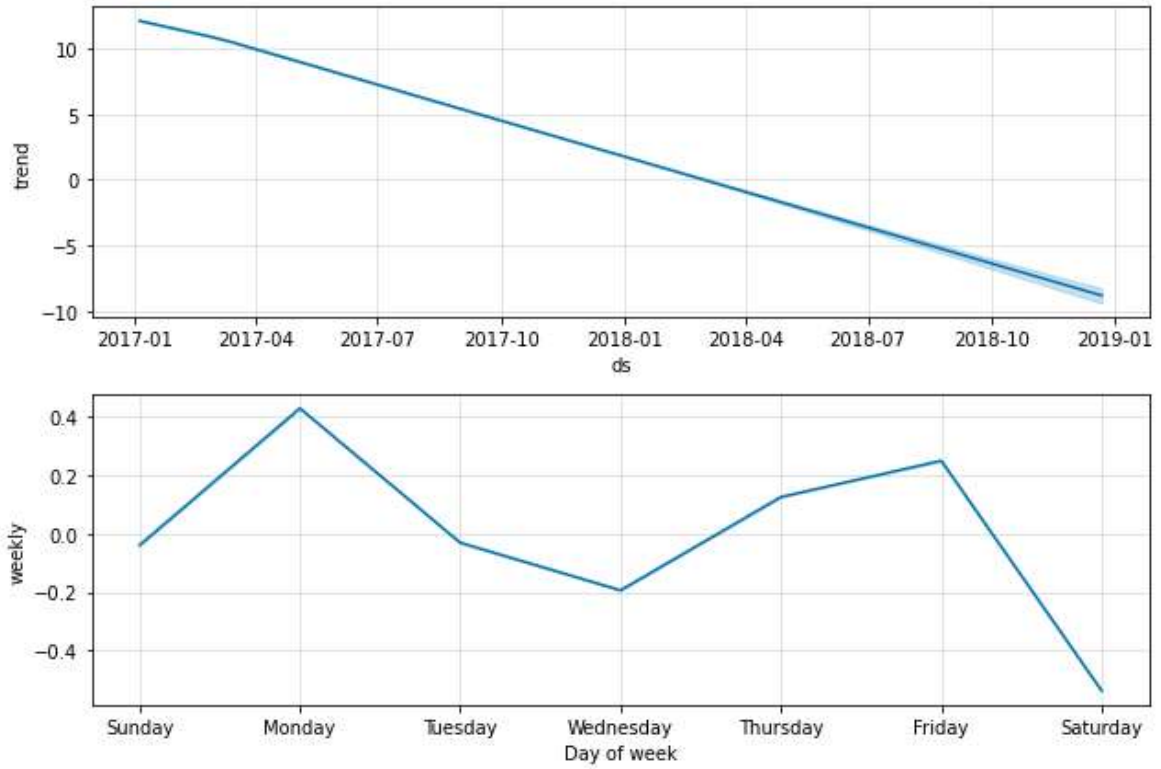
Out[9]:

	ds	yhat	yhat_lower	yhat_upper
648	2018-12-18	-8.736755	-19.272354	1.602213
649	2018-12-19	-8.929065	-19.320022	2.073831
650	2018-12-20	-8.640544	-18.768626	1.641611
651	2018-12-21	-8.546741	-19.332849	2.062546
652	2018-12-22	-9.360859	-19.960743	1.213743

In [10]:
```
#facebook prophet
forcast_fig1 = m.plot(forecast)
```

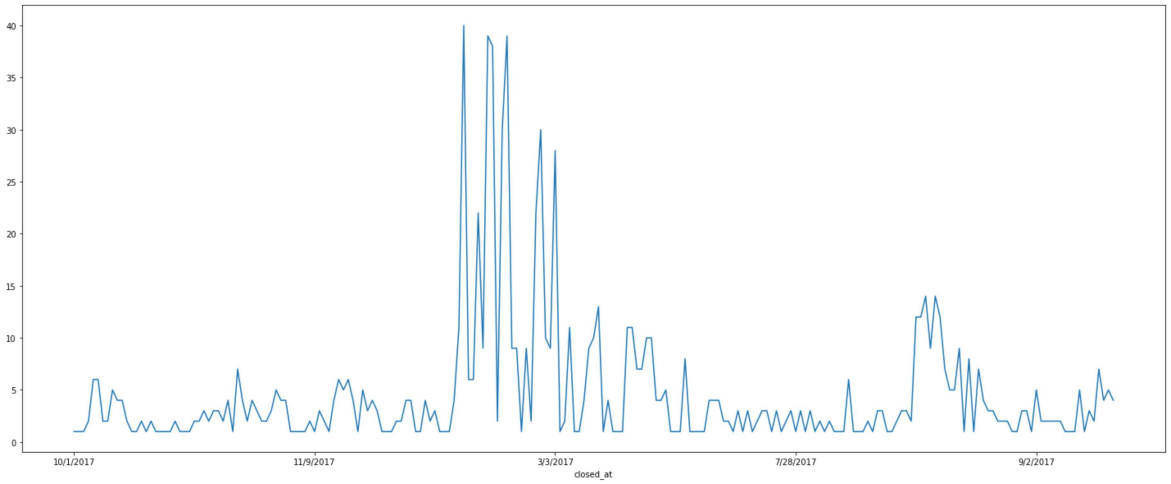


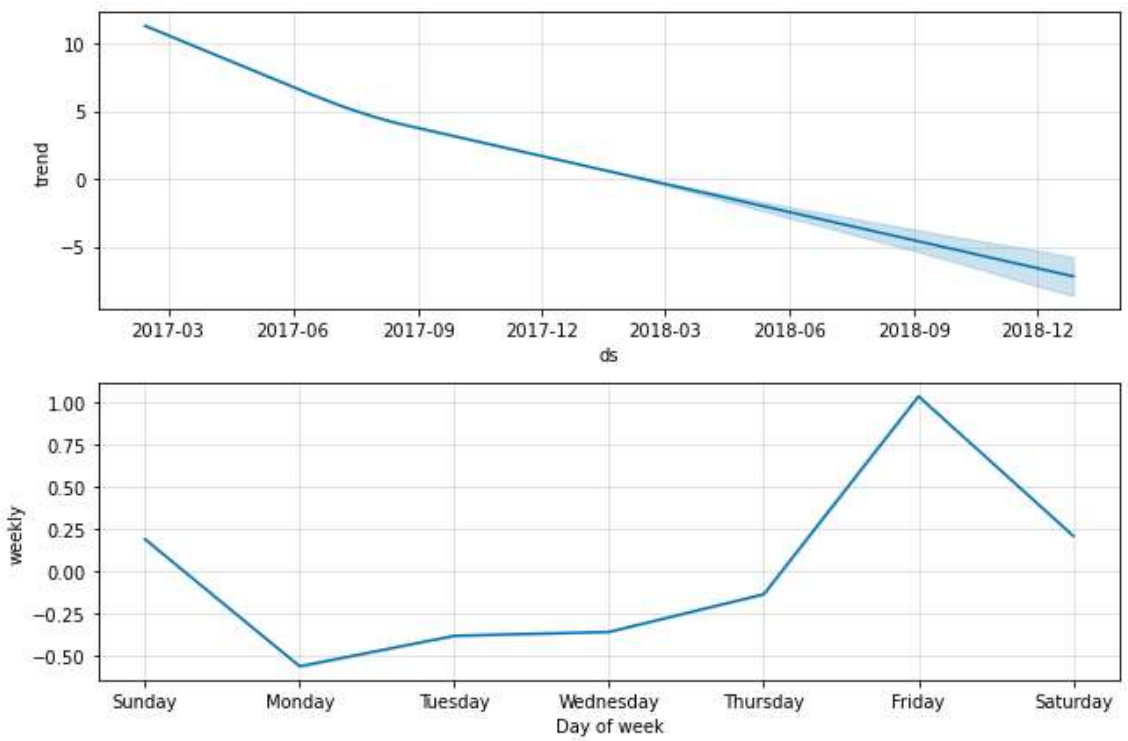
In [11]:
```
#Q1 : The day of the week maximum number of issues ceated : Monday

forcast_fig2 = m.plot_components(forecast)
```



In [12]:
```
# Q2 : The day of the week maximum number of issues closed

# and

# Q3 : The month of the year that has maximum number of issues closed

DailyIssue = df.groupby(['closed_at']).created_at.count()
DailyIssue.plot(figsize= (25, 10))

df1 = df.groupby(['closed_at'], as_index = False).count()
dataFrame = df1[['closed_at','issue_number']]
dataFrame.columns = ['ds', 'y']
dataFrame
dataFrame.to_csv (r'github_data.csv', index = None, header=True)

dataFrame = pd.read_csv('github_data.csv')
m = Prophet()
m.fit(dataFrame)

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

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

forcast_fig2 = m.plot_components(forecast)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_season
ality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonal
ity=True to override this.
```

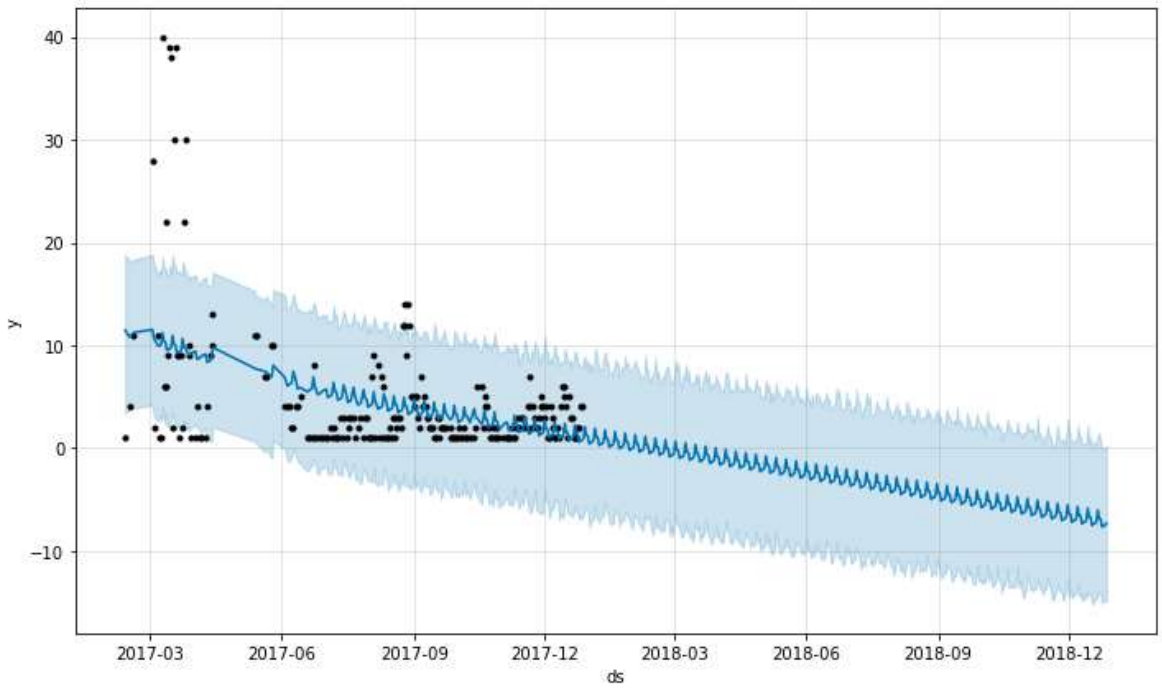




In [13]:
```
# answer 2 : day when maximum issues were closed is Friday

# answer 3 : Maximum issues were closed in March - 2017.
```

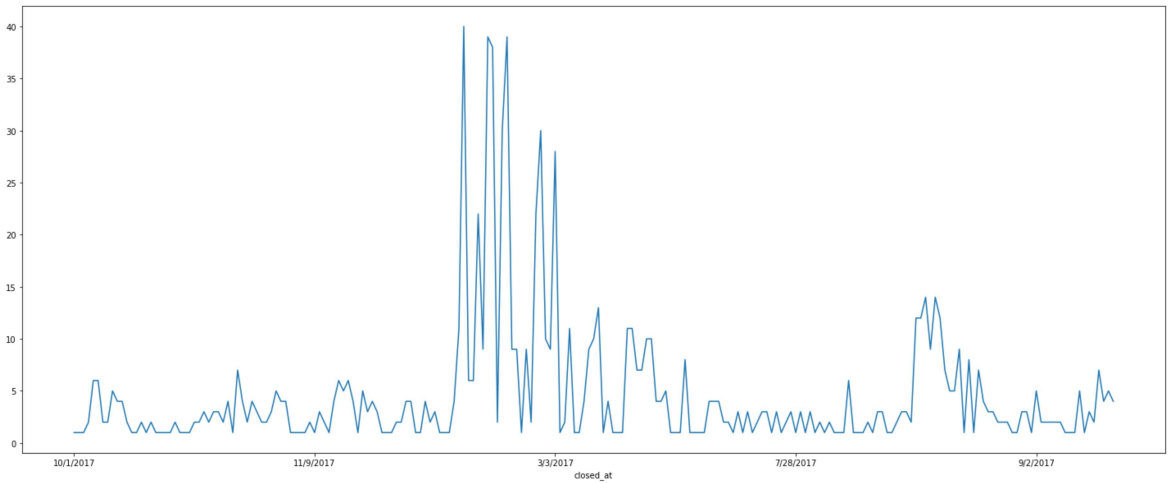
In [14]:
```
# Q4 : Plot the created issues forecast by calling the Prophet.plot method an


model = fbprophet.Prophet()
model.fit(dataFrame)

future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)

plot = model.plot(forecast)
```

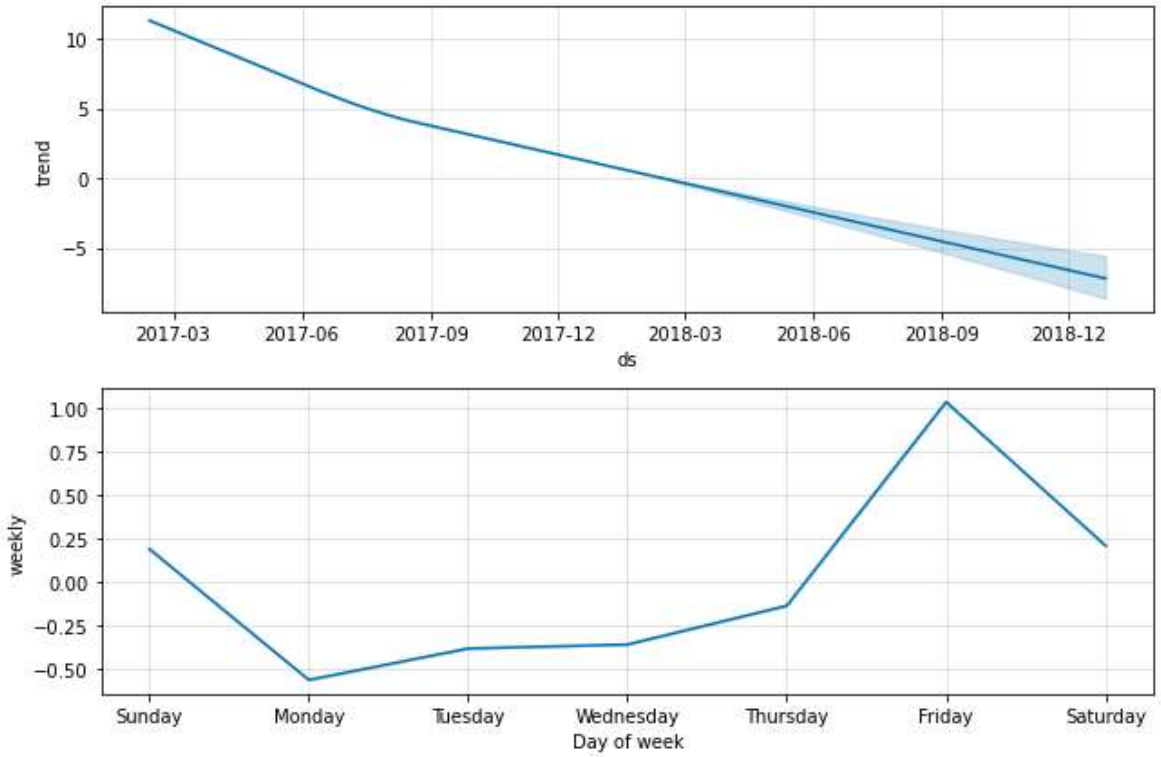
```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_season
ality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonal
ity=True to override this.
```



In [15]:
```
#5. Plot the closed issues forecast; use the Prophet.plot_components method.
# see the trend, yearly seasonality, and weekly
# seasonality of the time series. If you include holidays, you'll see those h

DailyIssue = df.groupby(['closed_at']).created_at.count()
DailyIssue.plot(figsize= (25, 10))

df1 = df.groupby(['closed_at'], as_index = False).count()
dataFrame = df1[['closed_at','issue_number']]
dataFrame.columns = ['ds', 'y']
dataFrame
dataFrame.to_csv (r'github_data.csv', index = None, header=True)

dataFrame = pd.read_csv('github_data.csv')
m = Prophet()
m.fit(dataFrame)

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

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

forcast_fig2 = m.plot_components(forecast)

#answer 5
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_season
ality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonal
ity=True to override this.
```




10
5
0
−5
trend
2017-03
2017-06
2017-09
2017-12
2018-03
2018-06
2018-09
2018-12
ds
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week

In [ ]:

In [ ]:

In [ ]: