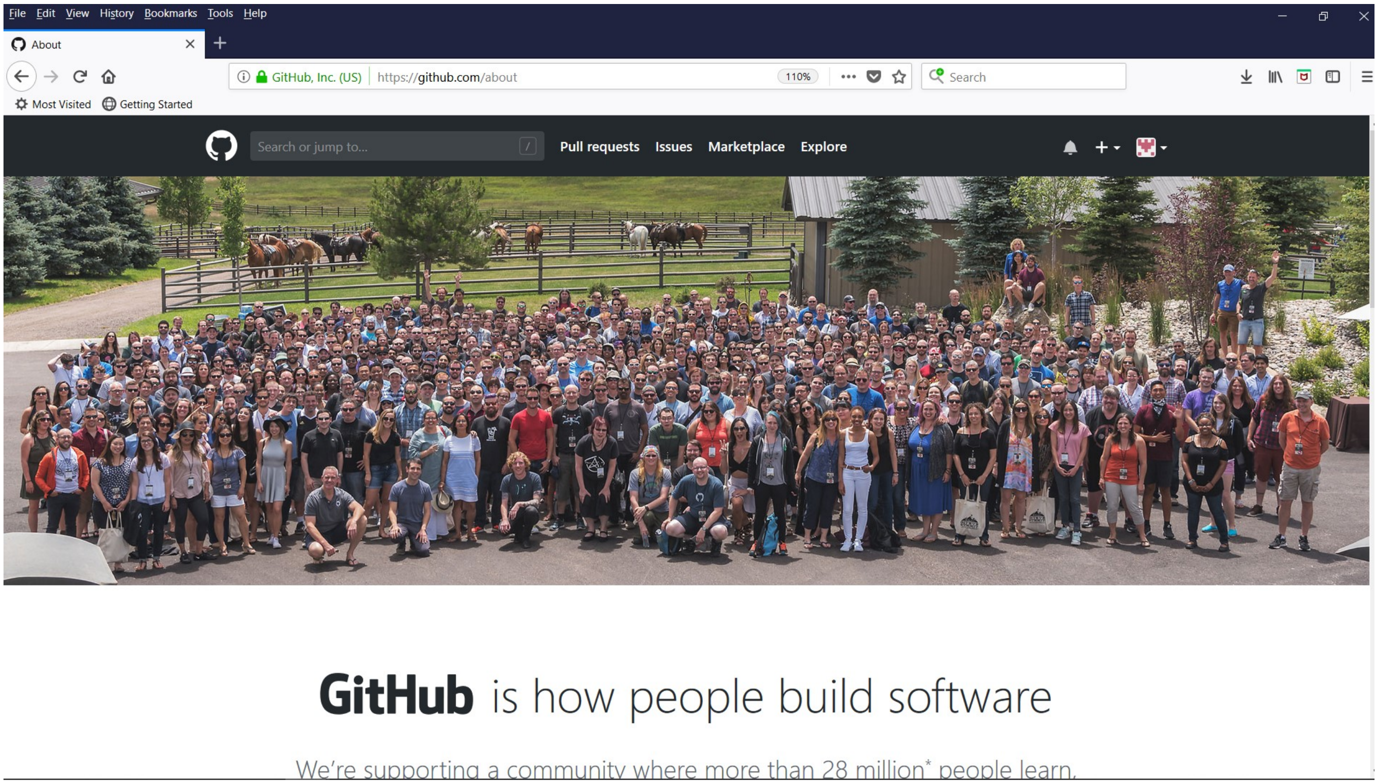
Bonus Assignment

Data Visualization for GitHub Issues

Author: Atef Bader Last Edit: 11/24/2018

```
In [1]: import matplotlib.pyplot as plt #for showing the image
        import matplotlib.image as mpimg #for reading the image
        import numpy as np #just to be safe
        %matplotlib inline
```

GitHub



GitHub is how people build software

We're supporting a community where more than 28 million* people learn,

Deliverables:

- Submit a single zip-compressed file that has the name: YourLastName_Bonus_Assignment_1 that has the following files:

  1. Your **PDF document** that has your Source code and output
  2. Your **ipynb script** that has your Source code and output

Objectives:

- Learn how to process data stored in **JSON** file
- Learn how to visualize data in **Stacked Chart**
- Learn how to plot data on **HeatMap**

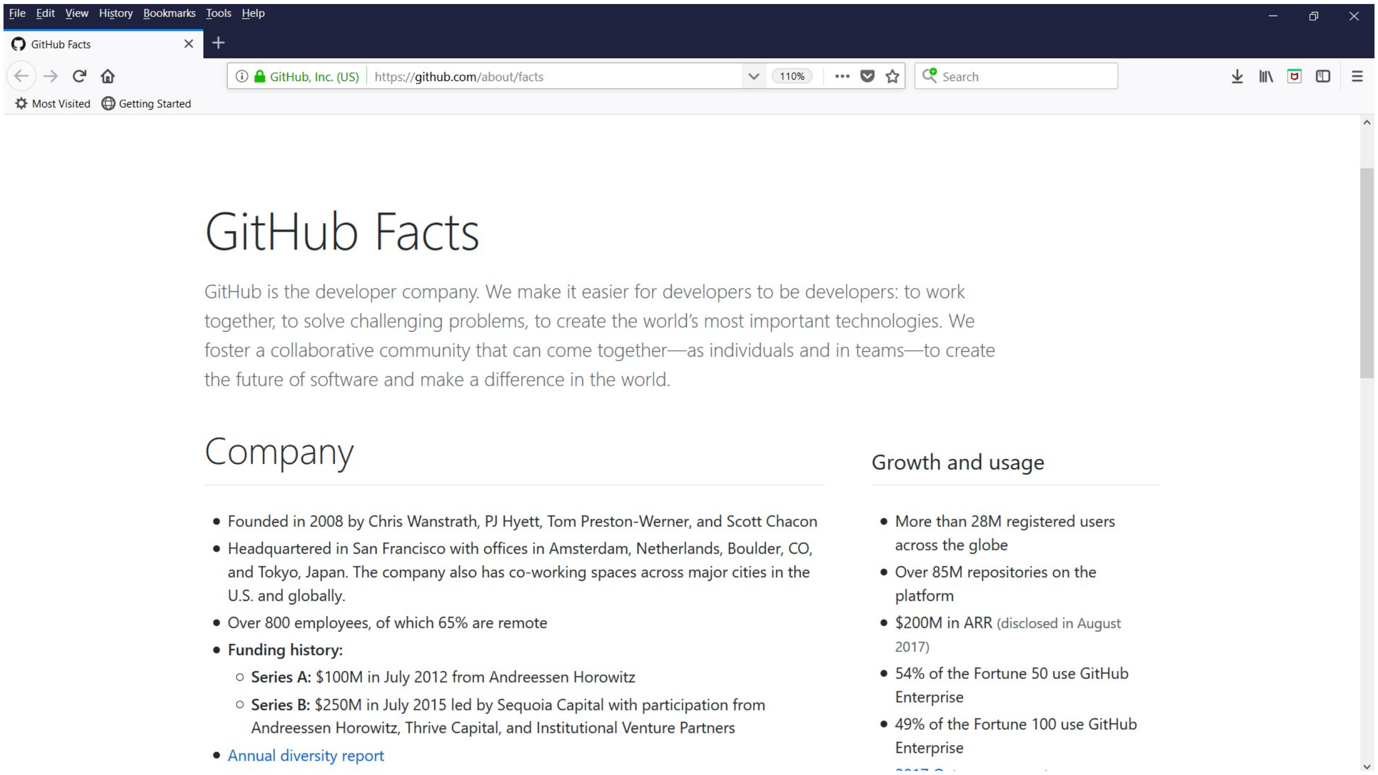
Submission Formats :

Create a folder or directory with all supplementary files with your last name at the beginning of the folder name, compress that folder with zip compression, and post the zip-archived folder under the assignment link in Canvas. The following files should be included in an archive folder/directory that is uploaded as a single zip-compressed file. (Use zip, not StuffIt or any 7z or any other compression method.)

1. Complete IPYNB script that has the source code in Python used to access and analyze the data. The code should be submitted as an IPYNB script that can be be loaded and run in Jupyter Notebook for Python
2. Output from the program, such as console listing/logs, text files, and graphics output for visualizations.
3. List file names and descriptions of files in the zip-compressed folder/directory.

Formatting Python Code When programming in Python, refer to Kenneth Reitz' PEP 8: The Style Guide for Python Code: http://pep8.org/ (http://pep8.org/) (Links to an external site.)Links to an external site. There is the Google style guide for Python at https://google.github.io/styleguide/pyguide.html (https://google.github.io/styleguide/pyguide.html) (Links to an external site.)Links to an external site. Comment often and in detail.

GitHub Facts



File Edit View History Bookmarks Tools Help
GitHub Facts
GitHub, Inc. (US) https://github.com/about/facts
110%
Search
Most Visited
Getting Started

GitHub Facts

GitHub is the developer company. We make it easier for developers to be developers: to work together, to solve challenging problems, to create the world's most important technologies. We foster a collaborative community that can come together—as individuals and in teams—to create the future of software and make a difference in the world.

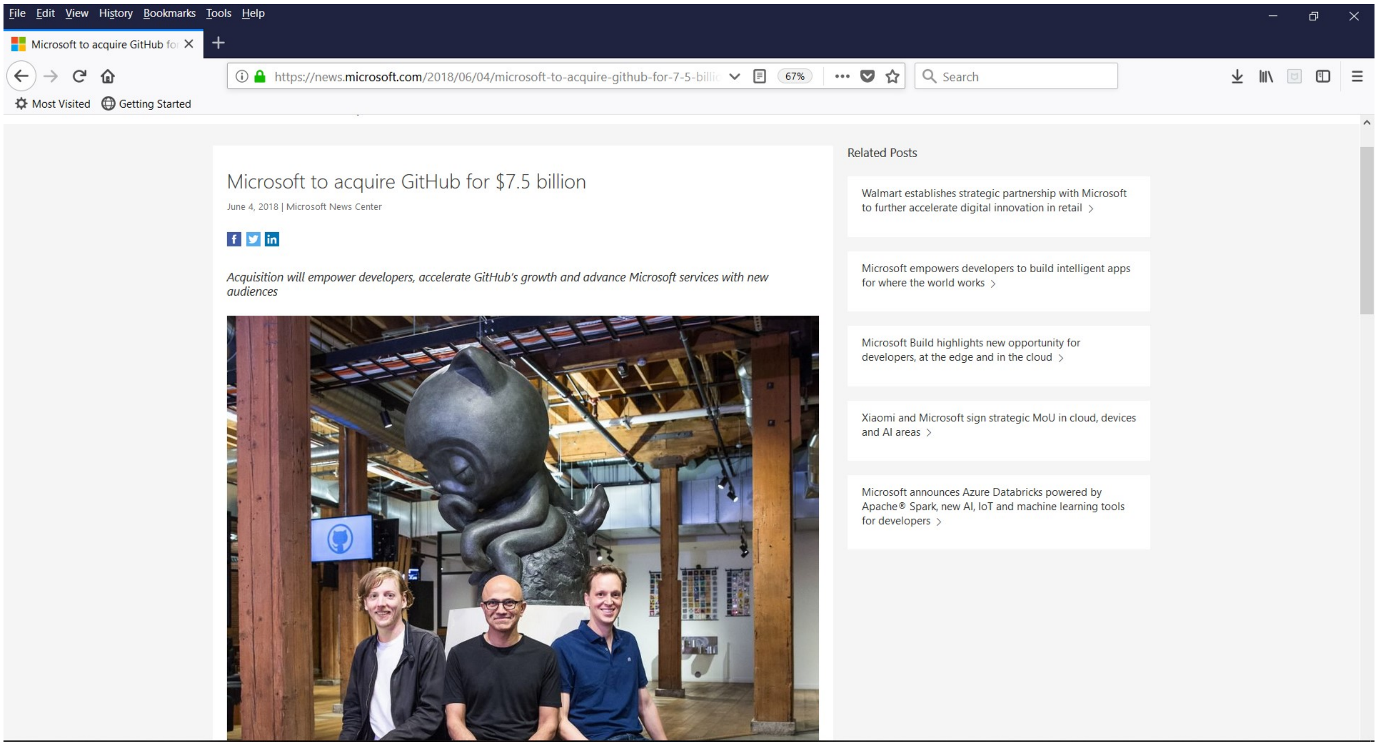
Company

- Founded in 2008 by Chris Wanstrath, PJ Hyett, Tom Preston-Werner, and Scott Chacon
- Headquartered in San Francisco with offices in Amsterdam, Netherlands, Boulder, CO, and Tokyo, Japan. The company also has co-working spaces across major cities in the U.S. and globally.
- Over 800 employees, of which 65% are remote
- **Funding history:**
  - **Series A:** $100M in July 2012 from Andreessen Horowitz
  - **Series B:** $250M in July 2015 led by Sequoia Capital with participation from Andreessen Horowitz, Thrive Capital, and Institutional Venture Partners
- Annual diversity report

Growth and usage

- More than 28M registered users across the globe
- Over 85M repositories on the platform
- $200M in ARR (disclosed in August 2017)
- 54% of the Fortune 50 use GitHub Enterprise
- 49% of the Fortune 100 use GitHub Enterprise

Microsoft acquired GitHub for $7.5 Billion in 2018



File Edit View History Bookmarks Tools Help
Microsoft to acquire GitHub fo
https://news.microsoft.com/2018/06/04/microsoft-to-acquire-github-for-7-5-billic
67%
Search
Most Visited
Getting Started

Microsoft to acquire GitHub for $7.5 billion

June 4, 2018 | Microsoft News Center

Acquisition will empower developers, accelerate GitHub's growth and advance Microsoft services with new audiences

Related Posts

- Walmart establishes strategic partnership with Microsoft to further accelerate digital innovation in retail >
- Microsoft empowers developers to build intelligent apps for where the world works >
- Microsoft Build highlights new opportunity for developers, at the edge and in the cloud >
- Xiaomi and Microsoft sign strategic MoU in cloud, devices and AI areas >
- Microsoft announces Azure Databricks powered by Apache® Spark, new AI, IoT and machine learning tools for developers >

Documentation of GitHub Issues

Tutorial and Documentation on how issues are created and managed on GitHub can be found at this URL :

[Managing GitHub Issues (https://help.github.com/categories/managing-your-work-on-github/)](https://help.github.com/categories/managing-your-work-on-github/)

Data Viualization for GitGub Issues

In this assignment you will learn how to plot the Graph for sample data of GitHub issues with different labels created and closed on different dates for a sample of data created for experimental purposes on GitHub

Examples of Issue Form Filled out

"issue_number": 1219, "created_at": "2018-07-11", "closed_at": "2018-08-12", "labels": ["Address:111 W Jackson Blvd Chicago 60604", "Category:Bug", "DetectionPhase:Design", "OriginationPhase:Design", "Priority:Critical", "Status:Approved"], "State": "closed", "Author": "SEngineer68H" "issue_number": 11, "created_at": "2018-01-09", "closed_at": null, "labels": ["Address:600 E GRAND AVE", "Category:Bug", "DetectionPhase:Testing", "Latitude:41.891551", "Longitude:-87.607375", "OriginationPhase:Testing", "Priority:Critical", "Status:Approved"], "State": "open", "Author": "HEngineer69D"

Data Set File: issues.json

In [2]:

```
import os

import _pickle as pickle

import pandas as pd                                    # panda's nickname is pd

import numpy as np                                     # numpy as np

from pandas import DataFrame, Series                   # for convenience

import matplotlib.pyplot as plt

%matplotlib inline
```

Reading the Dataset stored in JSON file :

Lets read the issues from the JSON file and plot them in a stacked chart

In [3]:

```
# Read the JSON file and load the data into a list of dictionaries

import json

list_of_issues_dict_data = [json.loads(line) for line in open('GitHub_issues.json
')]
```

In [4]:

```
list_of_issues_dict_data[0].keys()
```

Out[4]:

```
dict_keys(['issue_number', 'created_at', 'closed_at', 'labels', 'State', 'Author
'])
```

```
In [5]: list_of_issues_dict_data
```

```
Out[5]: [{'issue_number': 803,
          'created_at': '2018-04-02',
          'closed_at': '2018-04-09',
          'labels': ['Address:2525 S Martin Luther King Drive',
           'Category:Bug',
           'DetectionPhase:Testing',
           'Latitude:41.853136',
           'Longitude:-87.633160',
           'OriginationPhase:Design',
           'Priority:Critical',
           'Status:Rejected'],
          'State': 'closed',
          'Author': 'SPM587SP18'},
         {'issue_number': 802,
          'created_at': '2018-03-30',
          'closed_at': '2018-04-06',
          'labels': ['Address:2525 S Martin Luther King Drive',
           'Category:Bug',
           'DetectionPhase:Testing',
           'Latitude:41.853136',
           'Longitude:-87.633160',
           'OriginationPhase:Design',
           'Priority:Critical',
           'Status:Rejected'],
          'State': 'closed',
          'Author': 'SPM587SP18'},
         {'issue_number': 894,
          'created_at': '2018-05-10',
          'closed_at': '2018-08-10',
          'labels': ['Address:111 W JACKSON',
           'Category:Bug',
           'DetectionPhase:Design',
           'Latitude:41.877817',
           'Longitude:-87.631247',
           'OriginationPhase:Requirements',
           'Priority:Medium',
           'Status:Approved'],
          'State': 'closed',
          'Author': 'PEngineer54P'},
         {'issue_number': 891,
          'created_at': '2018-04-10',
          'closed_at': '2018-08-11',
          'labels': ['Address:1919 Dempster Street',
           'Category:Bug',
           'DetectionPhase:Design',
           'Latitude:42.041392',
           'Longitude:-87.700113',
           'OriginationPhase:Design',
           'Priority:Medium',
           'Status:pendingReview'],
          'State': 'closed',
          'Author': 'PEngineer99P'},
         {'issue_number': 888,
          'created_at': '2018-06-10',
          'closed_at': '2018-08-11',
          'labels': ['Address:1919 Dempster Street',
           'Category:Bug',
           'Latitude:42.041392',
           'Longitude:-87.700113'],
          'State': 'closed',
          'Author': 'JEngineer54B'},
         {'issue_number': 887,
          'created_at': '2018-04-10',
          'closed_at': '2018-06-11',
```

In [6]:

```
# Create the DataFrame object for the list_of_issues_dict_data object

issues_df = DataFrame(list_of_issues_dict_data)
```

In [7]:

```
issues_df.keys()
```

Out[7]:

```
Index(['Author', 'State', 'closed_at', 'created_at', 'issue_number', 'labels'],
dtype='object')
```

In [8]:

```
issues_df.dtypes
```

Out[8]:

```
Author          object
State           object
closed_at       object
created_at      object
issue_number     int64
labels          object
dtype: object
```

In [9]:

```
# Sanity test: print first 5 rows in our DataFrame

issues_df.head()
```

Out[9]:

	Author	State	closed_at	created_at	issue_number	labels
0	SPM587SP18	closed	2018-04-09	2018-04-02	803	[Address:2525 S Martin Luther King Drive, Cate...
1	SPM587SP18	closed	2018-04-06	2018-03-30	802	[Address:2525 S Martin Luther King Drive, Cate...
2	PEngineer54P	closed	2018-08-10	2018-05-10	894	[Address:111 W JACKSON, Category:Bug, Detectio...
3	PEngineer99P	closed	2018-08-11	2018-04-10	891	[Address:1919 Dempster Street, Category:Bug, D...
4	JEngineer54B	closed	2018-08-11	2018-06-10	888	[Address:1919 Dempster Street, Category:Bug, L...

In [10]:

```
# Prepare and Clean the dataframe object

wrangled_issues_df = issues_df[['Author','State','closed_at','created_at','issue_nu
mber']]
wrangled_issues_df.loc[0:len(wrangled_issues_df), 'OriginationPhase']= np.NaN

wrangled_issues_df.loc[0:len(wrangled_issues_df),'DetectionPhase']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Category']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Priority']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Status']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Address']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Latitude']= np.NaN
wrangled_issues_df.loc[0:len(wrangled_issues_df),'Longitude']= np.NaN
```

In [11]:
```
#Sanity test the content of the datframe object

wrangled_issues_df.head()
```

Out[11]:

	Author	State	closed_at	created_at	issue_number	OriginationPhase	DetectionPhase	Category	P
0	SPM587SP18	closed	2018-04-09	2018-04-02	803	NaN	NaN	NaN	
1	SPM587SP18	closed	2018-04-06	2018-03-30	802	NaN	NaN	NaN	
2	PEngineer54P	closed	2018-08-10	2018-05-10	894	NaN	NaN	NaN	
3	PEngineer99P	closed	2018-08-11	2018-04-10	891	NaN	NaN	NaN	
4	JEngineer54B	closed	2018-08-11	2018-06-10	888	NaN	NaN	NaN	

In [12]:
```
# we need to create a list of the key:value pairs in labels

for i in range(0, len(issues_df)):

    if issues_df.iloc[i]['labels']:
        for label in issues_df.iloc[i]['labels']:
#            print(label)
            label_name= (label.split(':'))[0]
            label_value= (label.split(':'))[1]
            wrangled_issues_df.loc[i, label_name]=label_value
```

In [13]:
```
#Sanity test the content of the datframe object

wrangled_issues_df.head()
```

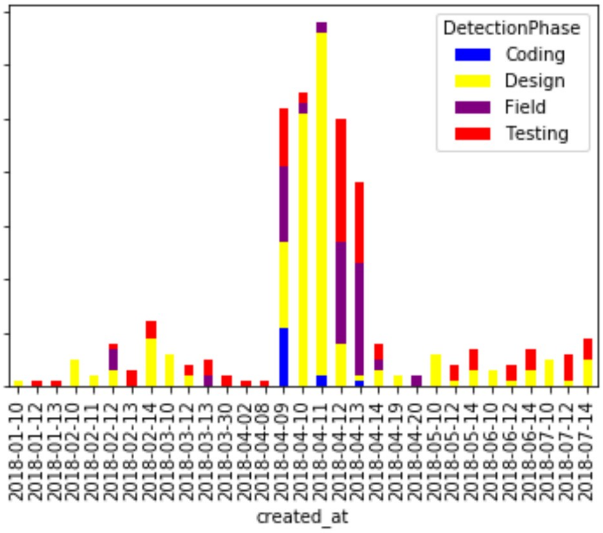
Out[13]:

	Author	State	closed_at	created_at	issue_number	OriginationPhase	DetectionPhase	Category	P
0	SPM587SP18	closed	2018-04-09	2018-04-02	803	Design	Testing	Bug	(
1	SPM587SP18	closed	2018-04-06	2018-03-30	802	Design	Testing	Bug	(
2	PEngineer54P	closed	2018-08-10	2018-05-10	894	Requirements	Design	Bug	M
3	PEngineer99P	closed	2018-08-11	2018-04-10	891	Design	Design	Bug	M
4	JEngineer54B	closed	2018-08-11	2018-06-10	888	NaN	NaN	Bug	

Plot in Stacked Bar Chart the total number of issues created every day for every Detaction Phase

In [14]:

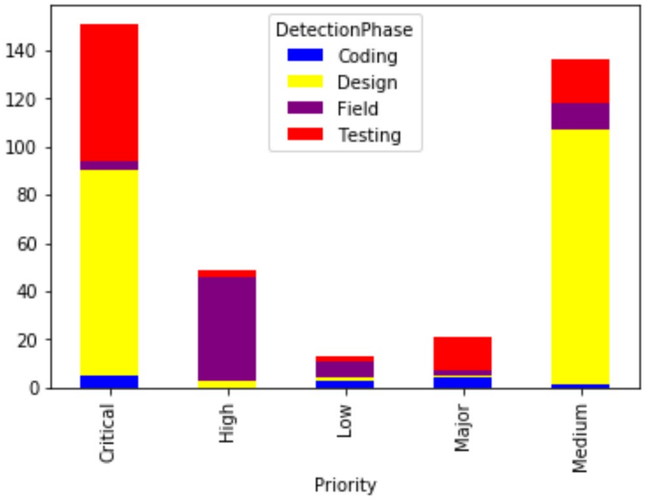
```
github_issues_by_date_created_detectionphase = wrangled_issues_df.groupby(['created
_at','DetectionPhase']).created_at.count()

github_issues_by_date_created_detectionphase_fig = github_issues_by_date_created_de
tectionphase.unstack().plot(kind='bar',stacked=True,  color=['blue','yellow', 'purp
le', 'red', 'green'], grid=False)
```



Plot in Stacked Bar Chart the total number of issues created for detection Phase based on thier priorites

In [15]:

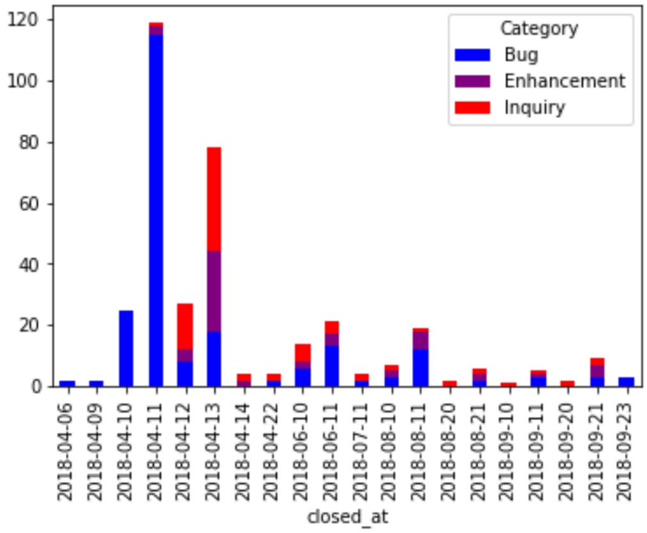
```
# Plot in Stacked Bar Chart the total number of issues created for detection Phase
based on thier priorites

github_issues_by_priority_detectionphase = wrangled_issues_df.groupby(['Priority','
DetectionPhase']).created_at.count()

github_issues_by_priority_detectionphase_fig = github_issues_by_priority_detectionp
hase.unstack().plot(kind='bar',stacked=True,  color=['blue','yellow', 'purple', 're
d', 'green'], grid=False)
```



Plot in Stacked Bar Chart the total number of issues closed every day for every Category

In [16]:

```
# Plot in Stacked Bar Chart the total number of issues closed every day for every Category
github_issues_by_closed_date_category = wrangled_issues_df.groupby(['closed_at','Category']).closed_at.count()

github_issues_by_closed_date_category_fig = github_issues_by_closed_date_category.unstack().plot(kind='bar',stacked=True,  color=['blue', 'purple', 'red'], grid=False)
```



Lets plot the issues on a HeatMap

We will use Folium HeatMap to plot on a HeatMap our GitHub issues using Latitude and Longitude pairs. Here is the API documentation for Folium/HeatMap

Folium/HeatMap API (http://python-visualization.github.io/folium/docs-v0.5.0/plugins.html)

Here is the command that you execute from the terminal window in order to install Folium:

- conda install -c conda-forge folium

In [17]:

```
import folium
from folium import plugins
```

In [18]:

```
# Lets take a VERTICAL SLICE  ['Latitude','Longitude'] of the dataframe object

df_lat_lng = wrangled_issues_df[['Latitude','Longitude']]
```

In [19]:

```
#Sanity test the content of the datframe object

df_lat_lng.head()
```

Out[19]:

	Latitude	Longitude
0	41.853136	-87.633160
1	41.853136	-87.633160
2	41.877817	-87.631247
3	42.041392	-87.700113
4	42.041392	-87.700113

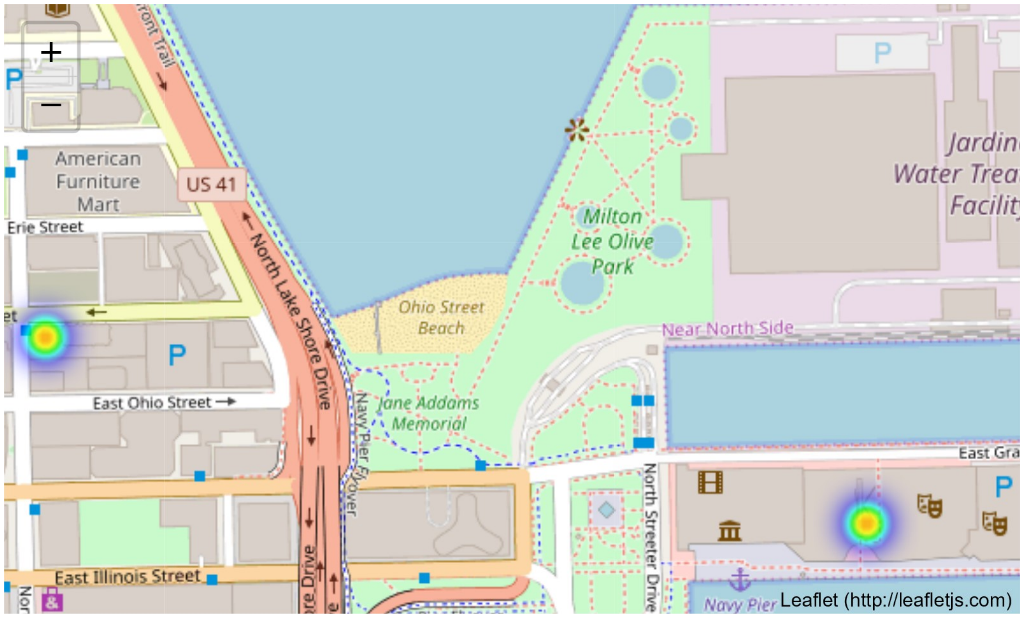
In [20]:

```
# Lets create a list of ['Latitude','Longitude'] pairs to feed into the HeatMap obj
ect

github_issues_coord = []

for i in range(0, len(df_lat_lng)):
    location_ll = []
    if ( pd.notnull(df_lat_lng.iloc[i]['Latitude']) and  pd.notnull(df_lat_lng.iloc
[i]['Longitude'])):
#        print(df_lat_lng.iloc[i]['Latitude'], df_lat_lng.iloc[i]['Longitude'])
        location_ll.append(float(df_lat_lng.iloc[i]['Latitude']))
        location_ll.append(float(df_lat_lng.iloc[i]['Longitude']))
        github_issues_coord.append(location_ll)
```

In [21]:

```
github_issues_heat_map = folium.Map([41.891551, -87.607375],zoom_start = 16)
github_issues_heat_map.add_child(plugins.HeatMap(github_issues_coord,radius=15))

# interact with the map below by zooming in/out
# Experiment with zoom_start and radius parameters of the HeatMap by using differen
values
```

Out[21]:



Requirement #1:

Plot in Stacked Bar Chart the total number of issues closed every day for every Origination Phase

```
In [22]: wrangled_issues_df['State'].value_counts()
```

```
Out[22]: closed    354
         open       56
         Name: State, dtype: int64
```

```
In [23]: wrangled_issues_df['OriginationPhase'].value_counts()
```

```
Out[23]: Requirements     159
         Design           136
         Coding            49
         Testing           19
         Documentation     11
         Field             10
         Name: OriginationPhase, dtype: int64
```

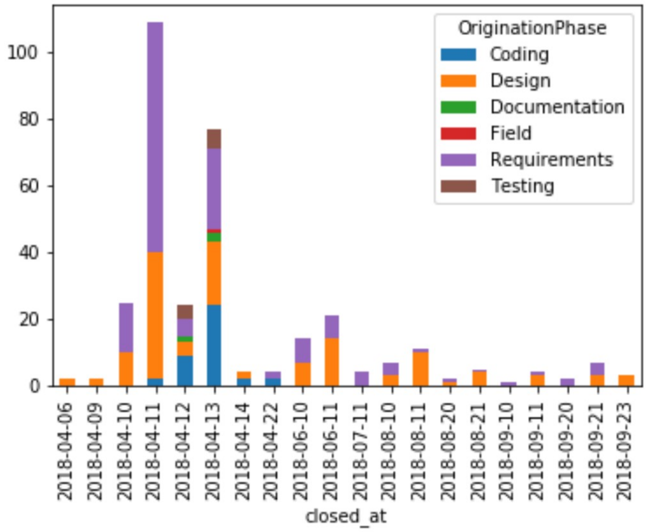
In [24]:

```
df1 = wrangled_issues_df.groupby(['closed_at','OriginationPhase']).closed_at.count
()
df1
```

Out[24]:

```
closed_at   OriginationPhase
2018-04-06  Design               2
2018-04-09  Design               2
2018-04-10  Design              10
            Requirements        15
2018-04-11  Coding               2
            Design              38
            Requirements        69
2018-04-12  Coding               9
            Design               4
            Documentation        2
            Requirements         5
            Testing              4
2018-04-13  Coding              24
            Design              19
            Documentation        3
            Field                1
            Requirements        24
            Testing              6
2018-04-14  Coding               2
            Design               2
2018-04-22  Coding               2
            Requirements         2
2018-06-10  Design               7
            Requirements         7
2018-06-11  Design              14
            Requirements         7
2018-07-11  Requirements         4
2018-08-10  Design               3
            Requirements         4
2018-08-11  Design              10
            Requirements         1
2018-08-20  Design               1
            Requirements         1
2018-08-21  Design               4
            Requirements         1
2018-09-10  Requirements         1
2018-09-11  Design               3
            Requirements         1
2018-09-20  Requirements         2
2018-09-21  Design               3
            Requirements         4
2018-09-23  Design               3
Name: closed_at, dtype: int64
```

In [25]:
```
df1_fig = df1.unstack().plot(kind='bar',stacked=True, grid=False);
```



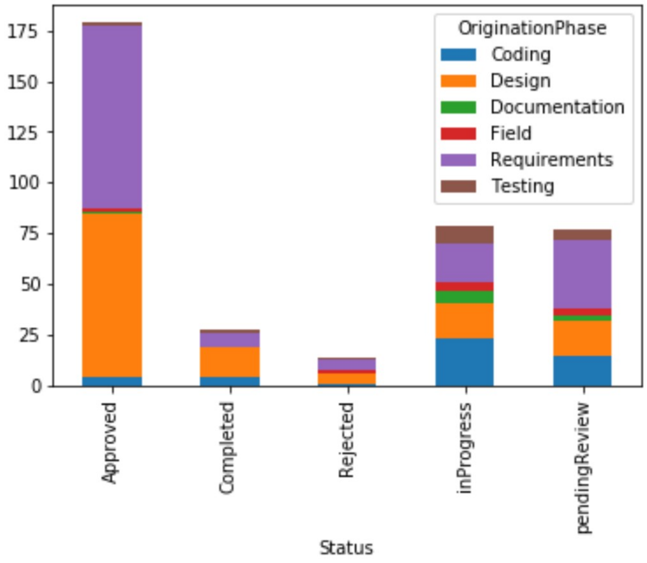
Requirement #2:

Create two bar charts:

1) Plot in stacked bar chart the total number of issues created for Origination Phase based on status

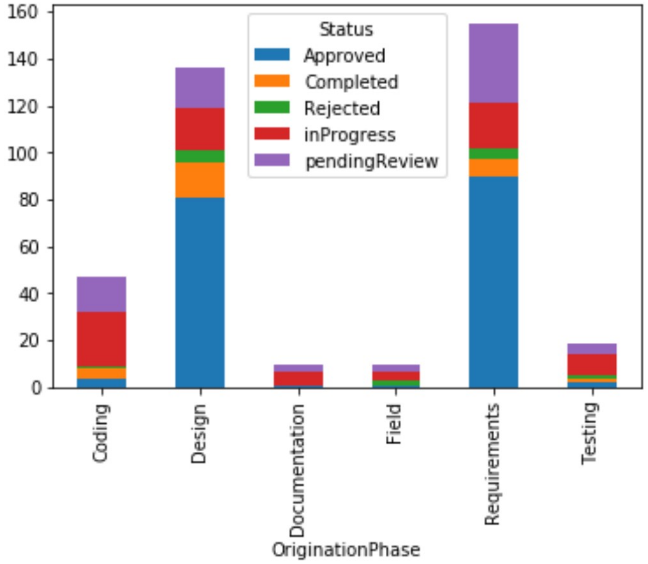
2) Plot in stacked bar chart the total number of issues created for Detection Phase based on status

In [26]:
```
# Plot in stacked bar chart the total number of issues created for Origination Phase based on status
df2 = wrangled_issues_df.groupby(['Status','OriginationPhase']).created_at.count()

df2_fig = df2.unstack().plot(kind='bar',stacked=True, grid=False)
```



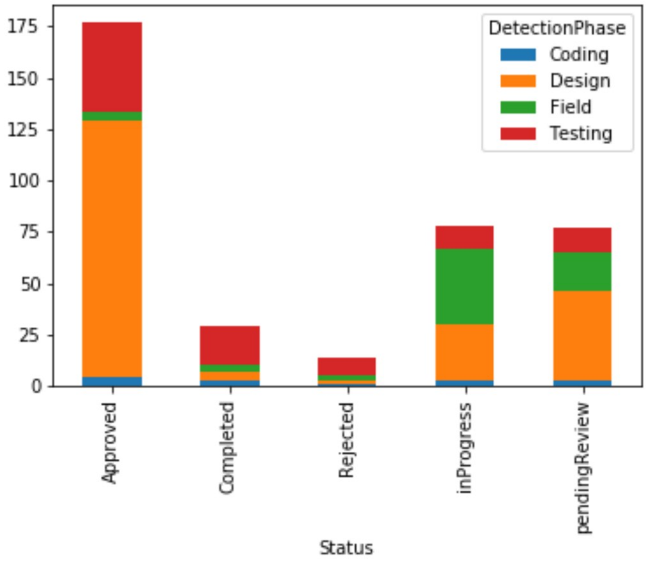
In [27]:

```
# another way of looking at the problem
df2a = wrangled_issues_df.groupby(['OriginationPhase', 'Status']).created_at.count
()

df2a_fig = df2a.unstack().plot(kind='bar',stacked=True, grid=False)
```

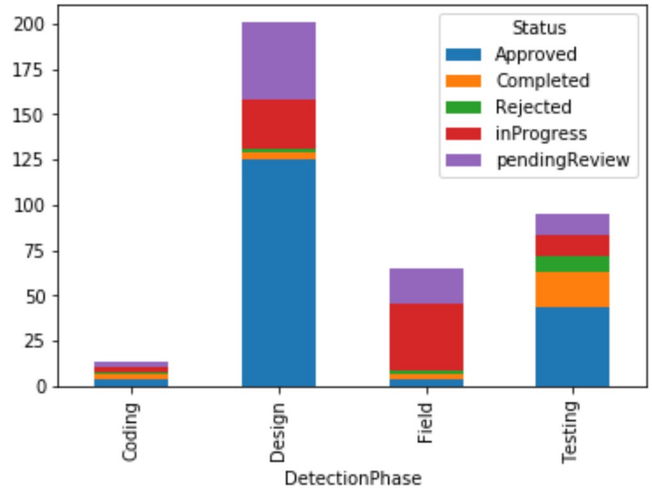


In [28]:

```
# Plot in stacked bar chart the total number of issues created for Detection Phase
based on status
df3 = wrangled_issues_df.groupby(['Status','DetectionPhase']).created_at.count()

df3_fig = df3.unstack().plot(kind='bar',stacked=True, grid=False)
```



In [29]:
```
# another way of looking at the problem
df3a = wrangled_issues_df.groupby(['DetectionPhase', 'Status']).created_at.count()

df3a_fig = df3a.unstack().plot(kind='bar',stacked=True, grid=False)
```



Requirement #3:

Create Folium HeatMap for issues that have Priority equals to Critical

In [30]:
```
df_lat_lng = wrangled_issues_df[wrangled_issues_df['Priority'] == 'Critical'][['Latitude','Longitude']]
df_lat_lng.head()
```

Out[30]:

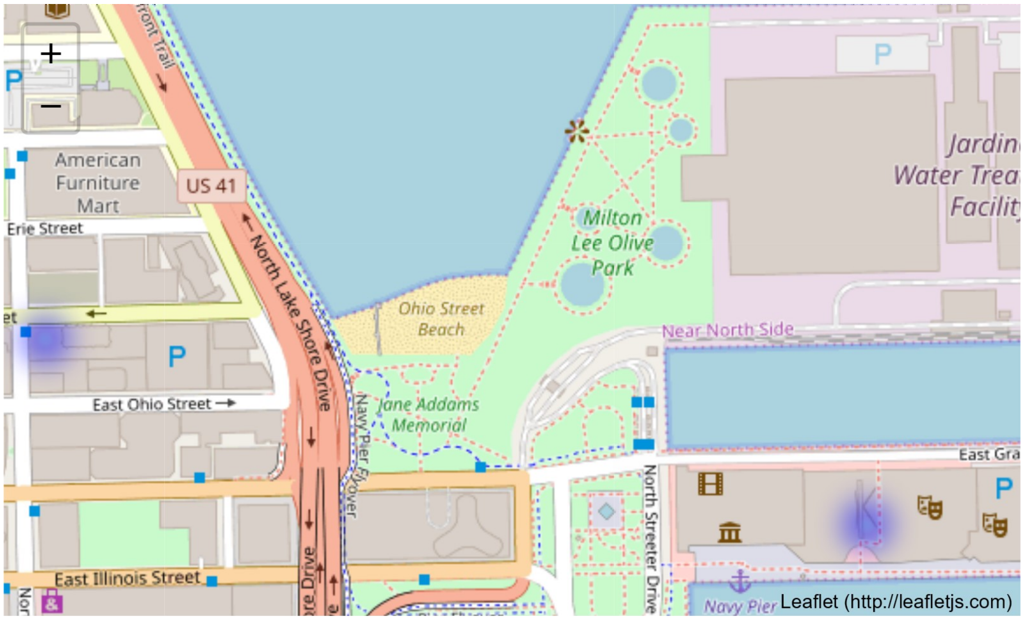
	Latitude	Longitude
0	41.853136	-87.633160
1	41.853136	-87.633160
5	42.041392	-87.700113
7	42.041392	-87.700113
9	42.041392	-87.700113

In [31]:
```
github_issues_coord = []

for i in range(0, len(df_lat_lng)):
    location_ll = []
    if ( pd.notnull(df_lat_lng.iloc[i]['Latitude']) and  pd.notnull(df_lat_lng.iloc[i]['Longitude'])):
        location_ll.append(float(df_lat_lng.iloc[i]['Latitude']))
        location_ll.append(float(df_lat_lng.iloc[i]['Longitude']))
        github_issues_coord.append(location_ll)

print(df_lat_lng.iloc[i]['Latitude'], df_lat_lng.iloc[i]['Longitude'])
```

```
42.040640 -87.680340
```

```
In [32]: github_issues_heat_map = folium.Map([41.891551, -87.607375],zoom_start = 16)
         github_issues_heat_map.add_child(plugins.HeatMap(github_issues_coord,radius=15))
```

Out[32]:



Requirement #4:

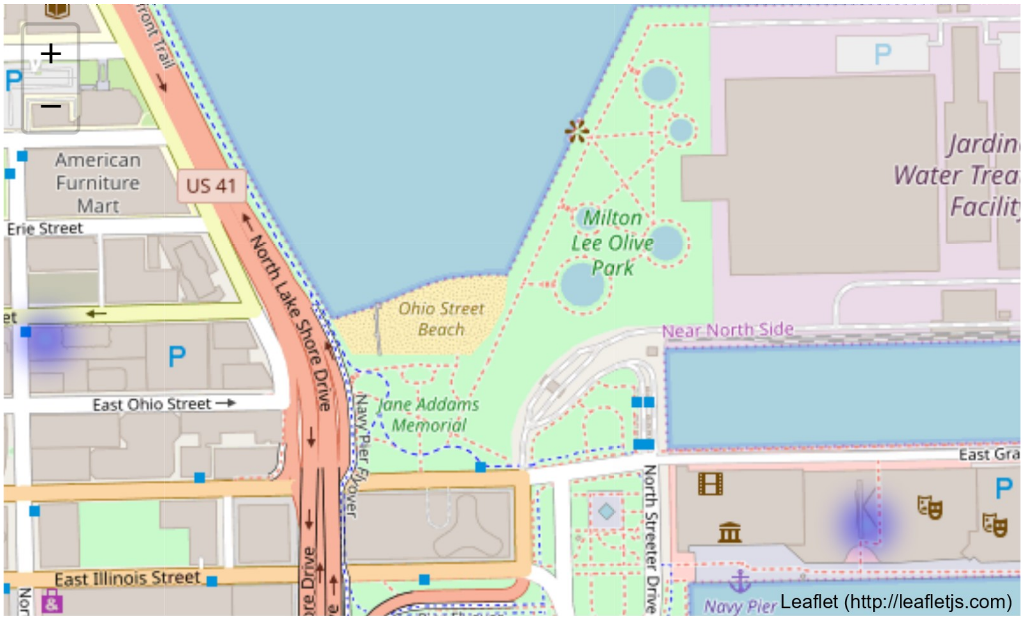
Create Folium HeatMap for issues that have Priority equals to Critical or High AND Status is Approved or inProgress

```
In [33]: df_lat_lng = wrangled_issues_df[((wrangled_issues_df['Priority'] == 'Critical') |
         (wrangled_issues_df['Priority'] == 'High')) &
                             ((wrangled_issues_df['Status'] == 'Approved') | (wrangled_issues
         _df['Status'] == 'inProgress'))]\
                              [['Latitude','Longitude']]
```

```
In [34]: github_issues_coord = []

         for i in range(0, len(df_lat_lng)):
             location_ll = []
             if ( pd.notnull(df_lat_lng.iloc[i]['Latitude']) and  pd.notnull(df_lat_lng.iloc
         [i]['Longitude'])):
                 location_ll.append(float(df_lat_lng.iloc[i]['Latitude']))
                 location_ll.append(float(df_lat_lng.iloc[i]['Longitude']))
                 github_issues_coord.append(location_ll)

         print(df_lat_lng.iloc[i]['Latitude'], df_lat_lng.iloc[i]['Longitude'])

         42.040640 -87.680340
```

```
In [35]: github_issues_heat_map = folium.Map([41.891551, -87.607375],zoom_start = 16)
         github_issues_heat_map.add_child(plugins.HeatMap(github_issues_coord,radius=15))
```

Out[35]:



Requirement #5:

Create Folium HeatMap for issues that have Field as the DetectionPhase and created during the month of April, 2018.

```
In [36]: wrangled_issues_df['DetectionPhase'].value_counts()
```

```
Out[36]: Design     201
         Testing     95
         Field       68
         Coding      14
         Name: DetectionPhase, dtype: int64
```

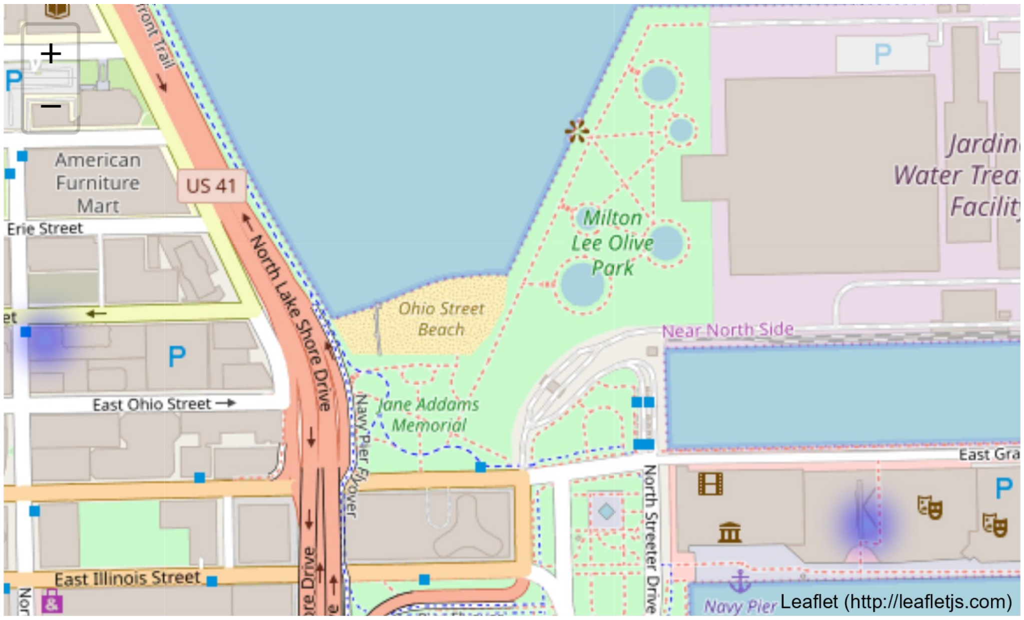
```
In [37]: ## ((pd.to_datetime(wrangled_issues_df['created_at']) > pd.to_datetime('2018-04-01
         ')) &
         ##  (pd.to_datetime(wrangled_issues_df['created_at']) <= pd.to_datetime('2018-04-30
         ')))

         df_lat_lng = wrangled_issues_df[(wrangled_issues_df['DetectionPhase'] == 'Field') &
                                         (wrangled_issues_df['created_at'].str[0:7] == '2018
         -04')][['Latitude','Longitude']]
```

```
In [38]: github_issues_coord = []

         for i in range(0, len(df_lat_lng)):
             location_ll = []
             if ( pd.notnull(df_lat_lng.iloc[i]['Latitude']) and  pd.notnull(df_lat_lng.iloc
         [i]['Longitude'])):
                 location_ll.append(float(df_lat_lng.iloc[i]['Latitude']))
                 location_ll.append(float(df_lat_lng.iloc[i]['Longitude']))
                 github_issues_coord.append(location_ll)
```

In [39]:

```
github_issues_heat_map = folium.Map([41.891551, -87.607375],zoom_start = 16)
github_issues_heat_map.add_child(plugins.HeatMap(github_issues_coord,radius=15))
```

Out[39]:



In [40]:

```
#wrangled_issues_df[((wrangled_issues_df['DetectionPhase'] == 'Field') &
#                                  (wrangled_issues_df['created_at'].str[0:7] == '201
8-04'))]
```