

DOCUMENTATION

1. Aim

Build a software to output a .geojson file for the visualisation of input csv file related to the crimes in California.

2. Working

- Import the required modules required to process the data. (csv, Counter, matplotlib and numpy)
- Parse the input data, and test if parsed data can be visualized based on type and days.
- Write a function to generate a .geojson file.
- Paste the .geojson file in gist.github.com
- The data will be visualized

CODE

```
import geojson
from collections import Counter
import csv
import matplotlib.pyplot as plt
import numpy as np
```

```
MY_FILE = "sample_sfpd_incident_all.csv"
```

```
def parse(raw_file, delimiter):
    """Raw csv to JSON"""
    #open file
    opened_file = open(raw_file)
    #read file
    csv_data = csv.reader(opened_file, delimiter = delimiter)
    #empty list
    parsed_data = []
    #find row with headers
    field = next(csv_data)
    #iterate through the loop
    for row in csv_data:
        parsed_data.append(dict(zip(field,row)))
    #close file
    opened_file.close()
    return parsed_data
```

```
def visualize_days():
    data_file = parse(MY_FILE, ",")
    counter = Counter(item["DayOfWeek"] for item in data_file)
    data_list =
[counter["Monday"],counter["Tuesday"],counter["Wednesday"],counter["Thursday"],counter["Friday"],counter["Saturday"],counter["Sunday"]]
    day_tuple = tuple(["Mon", "Tues", "Wed", "Thurs", "Fri", "Sat", "Sun"])
    plt.plot(data_list)
    plt.xticks(range(len(day_tuple)), day_tuple)
    # Save the plot
    plt.savefig("Days.png")
    # Close figure
    plt.clf()
```

```
def visualize_type():
    data_file = parse(MY_FILE, ",")
    counter = Counter(item["Category"] for item in data_file)
```

```
labels = tuple(counter.keys())
xlocations = np.array(range(len(labels)))+0.5
width = 0.5 #Of each bar
plt.bar(xlocations, counter.values(), width=width)
plt.xticks(xlocations + width / 2, labels, rotation=90)
plt.subplots_adjust(bottom=0.4)
plt.rcParams['figure.figsize'] = 12, 8
plt.savefig("Type.png")
plt.clf()

def create_map(data_file):
    geo_map = {"type": "FeatureCollection"}

    # Define empty list to collect each point to graph
    item_list = []

    # Iterate over our data to create GeoJSON document.
    for index, line in enumerate(data_file):

        if line['X'] == "0" or line['Y'] == "0":
            continue

        # Setup a new dictionary for each iteration.
        data = {}

        # Assigne line items to appropriate GeoJSON fields.
        data['type'] = 'Feature'
        data['id'] = index
        data['properties'] = {'title': line['Category'],
                             'description': line['Descript'],
                             'date': line['Date']}
        data['geometry'] = {'type': 'Point',
                           'coordinates': (line['X'], line['Y'])}

        # Add data dictionary to our item_list
        item_list.append(data)
    for point in item_list:
        geo_map.setdefault('features', []).append(point)

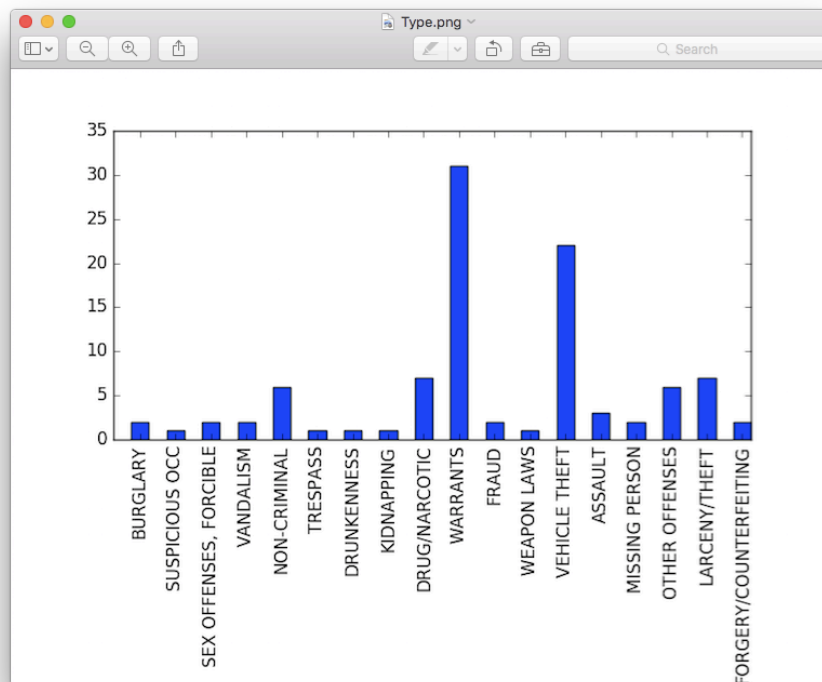
    with open('file_sf.geojson', 'w') as f:
        f.write(geojson.dumps(geo_map))

def main():
    #visualize_type() FOR TESTING
    #visualize_days() FOR TESTING
    data = parse(MY_FILE, ",")
    return create_map(data)

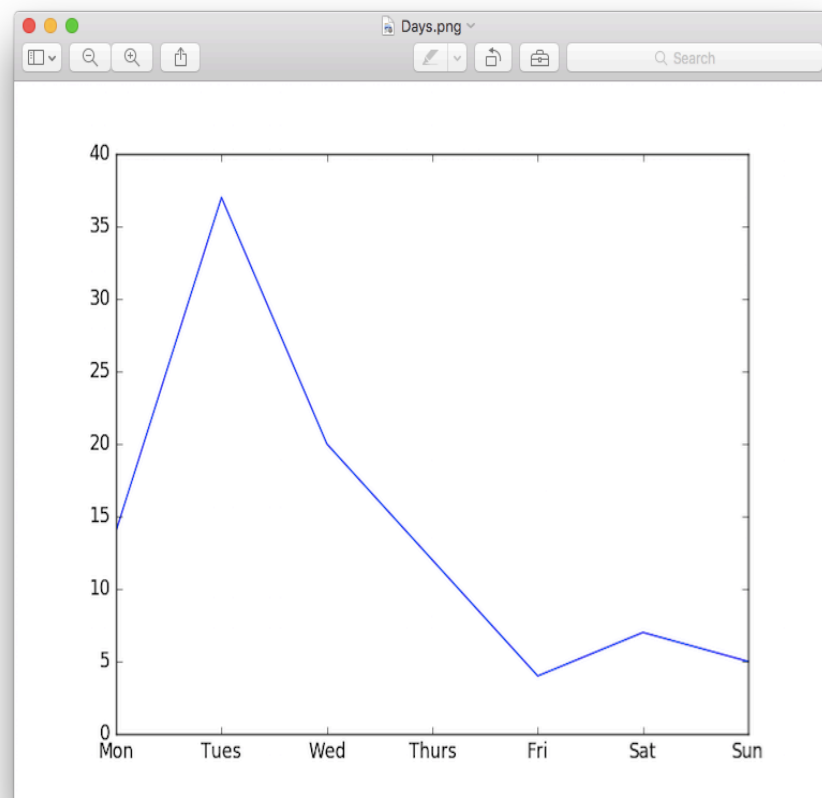
if __name__ == "__main__":
    main()
```

OUTPUTS

visualize_type()



visualize_days()



Visualisation of Crimes in California

Abhinav Goel
Semester IV, 2016

map()
(on gist.github.com)

