

final asignment deel 2

February 15, 2019

```
In [137]: import numpy as np # useful for many scientific computing in Python
import pandas as pd # primary data structure library
```

```
In [2]: !conda install -c conda-forge folium=0.5.0 --yes
import folium

print('Folium installed and imported!')
```

Collecting package metadata: done
Solving environment: done

Package Plan

environment location: /home/jupyterlab/conda

added / updated specs:
- folium=0.5.0

The following packages will be downloaded:

package	build	
conda-4.6.4	py36_0	877 KB conda-forge
Total:		877 KB

The following packages will be UPDATED:

conda 4.6.3-py36_0 --> 4.6.4-py36_0

Downloading and Extracting Packages

conda-4.6.4 | 877 KB | ##### | 100%

Preparing transaction: done

Verifying transaction: done

Executing transaction: done
Folium installed and imported!

In [138]: df_sanfran = pd.read_csv('Police_Department_Incidents_-_Previous_Year__2016_.csv')

In [139]: df_sanfran.head()

```
Out[139]:
```

	IncidntNum	Category	Descript \
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE
2	141059263	WARRANTS	WARRANT ARREST
3	160013662	NON-CRIMINAL	LOST PROPERTY
4	160002740	NON-CRIMINAL	LOST PROPERTY

	DayOfWeek	Date	Time	PdDistrict	Resolution \
0	Friday	01/29/2016	12:00:00 AM	11:00 SOUTHERN	ARREST, BOOKED
1	Friday	01/29/2016	12:00:00 AM	11:00 SOUTHERN	ARREST, BOOKED
2	Monday	04/25/2016	12:00:00 AM	14:59 BAYVIEW	ARREST, BOOKED
3	Tuesday	01/05/2016	12:00:00 AM	23:50 TENDERLOIN	NONE
4	Friday	01/01/2016	12:00:00 AM	00:30 MISSION	NONE

	Address	X	Y \
0	800 Block of BRYANT ST	-122.403405	37.775421
1	800 Block of BRYANT ST	-122.403405	37.775421
2	KEITH ST / SHAFTER AV	-122.388856	37.729981
3	JONES ST / OFARRELL ST	-122.412971	37.785788
4	16TH ST / MISSION ST	-122.419672	37.765050

	Location	PdId
0	(37.775420706711, -122.403404791479)	12005827212120
1	(37.775420706711, -122.403404791479)	12005827212168
2	(37.7299809672996, -122.388856204292)	14105926363010
3	(37.7857883766888, -122.412970537591)	16001366271000
4	(37.7650501214668, -122.419671780296)	16000274071000

In [111]: df_sanfran.tail()

```
Out[111]:
```

	IncidntNum	Category \
150495	161061000	ASSAULT
150496	176000742	NON-CRIMINAL
150497	176000758	LARCENY/THEFT
150498	176000764	LARCENY/THEFT
150499	179002868	OTHER OFFENSES

	Descript	DayOfWeek \
150495	BATTERY	Friday
150496	LOST PROPERTY	Friday
150497	PETTY THEFT OF PROPERTY	Thursday

150498	GRAND THEFT OF PROPERTY	Friday
150499	FRAUDULENT GAME OR TRICK, OBTAINING MONEY OR P...	Friday

	Date	Time	PdDistrict	Resolution	\
150495	12/30/2016	12:00:00 AM	21:01	PARK	NONE
150496	12/30/2016	12:00:00 AM	08:00	CENTRAL	NONE
150497	12/29/2016	12:00:00 AM	20:00	CENTRAL	NONE
150498	12/30/2016	12:00:00 AM	10:00	CENTRAL	NONE
150499	12/02/2016	12:00:00 AM	14:00	SOUTHERN	NONE

	Address	X	Y	\
150495	OAK ST / STANYAN ST	-122.453982	37.771428	
150496	JACKSON ST / SANSOME ST	-122.401857	37.796626	
150497	PINE ST / TAYLOR ST	-122.412269	37.790673	
150498	200 Block of STOCKTON ST	-122.406659	37.788275	
150499	800 Block of BRYANT ST	-122.403405	37.775421	

	Location	PdId
150495	(37.7714278595913, -122.453981622365)	16106100004134
150496	(37.7966261239618, -122.401857374739)	17600074271000
150497	(37.7906727649886, -122.41226909106)	17600075806372
150498	(37.7882745285785, -122.406658711008)	17600076406374
150499	(37.775420706711, -122.403404791479)	17900286809024

```
In [112]: df_sanfran.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150500 entries, 0 to 150499
Data columns (total 13 columns):
IncidentNum    150500 non-null int64
Category       150500 non-null object
Descript       150500 non-null object
DayOfWeek      150500 non-null object
Date           150500 non-null object
Time           150500 non-null object
PdDistrict     150499 non-null object
Resolution     150500 non-null object
Address        150500 non-null object
X              150500 non-null float64
Y              150500 non-null float64
Location       150500 non-null object
PdId           150500 non-null int64
dtypes: float64(2), int64(2), object(9)
memory usage: 14.9+ MB
```

```
In [113]: df_sanfran.columns.values, df_sanfran.index.values
```

```
Out[113]: (array(['IncidentNum', 'Category', 'Descript', 'DayOfWeek', 'Date', 'Time',
                  'PdDistrict', 'Resolution', 'Address', 'X', 'Y', 'Location',
```

```

        'PdId'], dtype=object),
array([    0,     1,     2, ..., 150497, 150498, 150499]))

```

```
In [117]: df_sanfran ['Category']
```

```

Out[117]: 0          WEAPON LAWS
1          WEAPON LAWS
2          WARRANTS
3      NON-CRIMINAL
4      NON-CRIMINAL
5          ASSAULT
6      OTHER OFFENSES
7      NON-CRIMINAL
8          WARRANTS
9      MISSING PERSON
10         LARCENY/THEFT
11         NON-CRIMINAL
12      OTHER OFFENSES
13         ASSAULT
14         BURGLARY
15      STOLEN PROPERTY
16         ROBBERY
17         ASSAULT
18         ASSAULT
19         LARCENY/THEFT
20         FRAUD
21         WARRANTS
22         DRUG/NARCOTIC
23      OTHER OFFENSES
24         ASSAULT
25      OTHER OFFENSES
26      OTHER OFFENSES
27         BURGLARY
28      OTHER OFFENSES
29         NON-CRIMINAL

...
150470         FRAUD
150471         NON-CRIMINAL
150472         NON-CRIMINAL
150473         NON-CRIMINAL
150474         NON-CRIMINAL
150475         NON-CRIMINAL
150476         NON-CRIMINAL
150477         LARCENY/THEFT
150478         NON-CRIMINAL
150479         LARCENY/THEFT
150480         NON-CRIMINAL
150481         NON-CRIMINAL

```

```

150482      NON-CRIMINAL
150483      NON-CRIMINAL
150484      NON-CRIMINAL
150485      NON-CRIMINAL
150486      LARCENY/THEFT
150487      LARCENY/THEFT
150488      NON-CRIMINAL
150489      FRAUD
150490      ASSAULT
150491      ASSAULT
150492      ASSAULT
150493      ASSAULT
150494      SECONDARY CODES
150495      ASSAULT
150496      NON-CRIMINAL
150497      LARCENY/THEFT
150498      LARCENY/THEFT
150499      OTHER OFFENSES
Name: Category, Length: 150500, dtype: object

```

```
In [35]: df_sanfran.columns.tolist()
```

```

Out[35]: ['IncidntNum',
          'Category',
          'Descript',
          'DayOfWeek',
          'Date',
          'Time',
          'Neighbourhood',
          'Resolution',
          'Address',
          'X',
          'Y',
          'Location',
          'PdId']

```

```
In [125]: df_sanfran.shape
df_sanfran.columns.values
```

```

Out[125]: array(['Count', 'Category', 'Descript', 'DayOfWeek', 'Date', 'Time',
                 'Neighborhood', 'Resolution', 'Address', 'X', 'Y', 'Location',
                 'PdId'], dtype=object)

```

```
In [118]: df_sanfran.rename(columns={"PdDistrict":"Neighborhood"},inplace=True)
df_sanfran.rename(columns={"IncidntNum":"Count"},inplace=True)
```

```
In [132]: df_ng=df_sanfran [['Neighborhood',"Count"]]
df_ng
```

```

Out[132]:
Neighborhood      Count
0      SOUTHERN  120058272
1      SOUTHERN  120058272
2      BAYVIEW   141059263
3      TENDERLOIN 160013662
4      MISSION   160002740
5      NORTHERN  160002869
6      SOUTHERN  160003130
7      TENDERLOIN 160003259
8      SOUTHERN  160003970
9      BAYVIEW   160003641
10     TARAVAL   160086863
11     TARAVAL   160004053
12     BAYVIEW   160073014
13     INGLESIDE 140776777
14     CENTRAL   160004069
15     SOUTHERN  160004150
16     MISSION   160004241
17     MISSION   160004558
18     INGLESIDE 160004655
19     SOUTHERN  160004837
20     NORTHERN  160005421
21     NORTHERN  160005750
22     MISSION   160005841
23     MISSION   160005841
24     RICHMOND  160006071
25     TARAVAL   160006598
26     MISSION   160006770
27     NORTHERN  160006786
28     MISSION   160006952
29     INGLESIDE 160007057
...
150470  NORTHERN  166108011
150471  RICHMOND  166108403
150472  SOUTHERN  166108522
150473  CENTRAL   166115408
150474  SOUTHERN  166132446
150475  CENTRAL   166148001
150476  SOUTHERN  166149178
150477  CENTRAL   166149219
150478  SOUTHERN  166154882
150479  PARK       166167891
150480  CENTRAL   166193175
150481  TARAVAL   166205532
150482  TARAVAL   166222229
150483  TENDERLOIN  166246780
150484  SOUTHERN  166280675
150485  SOUTHERN  166281758

```

150486	CENTRAL	166285910
150487	TARAVAL	166293509
150488	MISSION	166297727
150489	SOUTHERN	169004125
150490	BAYVIEW	160661158
150491	SOUTHERN	160801508
150492	MISSION	160823994
150493	MISSION	160858327
150494	BAYVIEW	170302853
150495	PARK	161061000
150496	CENTRAL	176000742
150497	CENTRAL	176000758
150498	CENTRAL	176000764
150499	SOUTHERN	179002868

[150500 rows x 2 columns]

```
In [133]: df_ng=df_ng.groupby("Neighborhood",axis=0).count()
```

```
In [134]: df_ng
```

```
Out[134]:
```

	Neighborhood	Count
	BAYVIEW	14303
	CENTRAL	17666
	INGLESIDE	11594
	MISSION	19503
	NORTHERN	20100
	PARK	8699
	RICHMOND	8922
	SOUTHERN	28445
	TARAVAL	11325
	TENDERLOIN	9942

```
In [135]: df_ng.reset_index(inplace=True)
```

```
In [136]: df_ng
```

```
Out[136]:
```

	Neighborhood	Count
0	BAYVIEW	14303
1	CENTRAL	17666
2	INGLESIDE	11594
3	MISSION	19503
4	NORTHERN	20100
5	PARK	8699
6	RICHMOND	8922
7	SOUTHERN	28445
8	TARAVAL	11325
9	TENDERLOIN	9942

```
In [169]: df_ng
```

```
Out[169]:
```

	Neighborhood	Count
0	BAYVIEW	14303
1	CENTRAL	17666
2	INGLESIDE	11594
3	MISSION	19503
4	NORTHERN	20100
5	PARK	8699
6	RICHMOND	8922
7	SOUTHERN	28445
8	TARAVAL	11325
9	TENDERLOIN	9942

```
In [84]: !conda install -c conda-forge folium=0.5.0 --yes
import folium
```

```
print('Folium installed and imported!')
```

Collecting package metadata: done

Solving environment: done

All requested packages already installed.

Folium installed and imported!

```
In [175]: #df_ng['Count'].min(), df_ng['Count'].max()
          #df_ng.info()
          #df_ng['Neighborhood']
          # for sake of consistency, let's also make all column labels of type string
          df_ng.columns = list(map(str, df_ng.columns))
```

```
In [176]: # San Francisco latitude and longitude values
          latitude = 37.77
          longitude = -122.42

          # create map and display it
          sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)
```

```
In [178]: sanfran_geo = r'san-francisco.geojson'

          # create a numpy array of length 6
          threshold_scale = np.linspace(df_ng['Count'].min(),
                                         df_ng['Count'].max(),
                                         6, dtype=int)

          threshold_scale = threshold_scale.tolist() # change the numpy array to a list
          threshold_scale[-1] = threshold_scale[-1] + 1 # make sure that the last value of the l
```



```

# let Folium determine the scale.
#sanfran_map = folium.Map(location=[0, 0], zoom_start=2, tiles='Mapbox Bright')
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)

sanfran_map.choropleth(
    geo_data=sanfran_geo,
    data=df_ng,
    columns=['Neighborhood', 'Count'],
    key_on='feature.properties.DISTRICT',
    threshold_scale=threshold_scale,
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Crime Rate in San Francisco',
    reset=True
)
sanfran_map

```

```
Out[178]: <folium.folium.Map at 0x7fc19d9d7358>
```

```
In [ ]:
```

```
In [ ]:
```