BayAreaTwitter_DemographicVariablesAnalysis

August 8, 2019

[1]:
```
Name= 'Catherine Bui'
Organization= 'Center For Community Innovation'
Project= 'Twitter Displacement Study'
```

[1]:
```
%%time
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
import subprocess
from pathlib import Path
import csv
import numpy as np
import dask.dataframe as das
import sqlite3
import datetime
```

```
CPU times: user 1.44 s, sys: 240 ms, total: 1.68 s
Wall time: 2.32 s
```

0.0.1 READING THE CSV FILE

1. Clean and prepare the data

[2]:
```
%%time
file = "/scratch/public/catherinebui/sf_with_homeloc_one_tweet_per_day.csv"
rawfile = "/scratch/public/catherinebui/sf_with_homeloc.csv"
df = das.read_csv(rawfile)
sf_o = pd.read_csv(rawfile)
df = df.rename(columns = {'sf_with_homeloc.csv': 'id'})
sf_o = sf_o.rename(columns = {'sf_with_homeloc.csv': 'id'})
```

```
CPU times: user 3min 18s, sys: 17.7 s, total: 3min 36s
Wall time: 3min 36s
```

[5]:
```
#Dropping Nan values
df=df.dropna(subset = ['home_tract'])
df = df.dropna(subset= ['tract'])
```

```
df['tract'] = df['tract'].astype('int')
df['home_tract'] = df['home_tract'].astype('int')
```

[ ]:
```
#Creating the neigbhor binary variable for each tweet

#Creating a map of neigbhors to the source tract
sf_nn = pd.read_csv('sfbay_13county_nearest_neigbbor_new.csv')
def neighbors(src):
    for i in sf_nn['SRC_GEOID'].unique():
        if i == src:
            table=sf_nn[sf_nn['SRC_GEOID'] == i]
            return np.array(table['NBR_GEOID'])
    return []
n_ = []
for i in sf_nn['SRC_GEOID'].unique():
    n_.append(neighbors(i))
nn = pd.DataFrame(sf_nn['SRC_GEOID'].unique())
nn['neighbors'] = n_
nn = nn.rename(columns = {0: 'SRC_GEOID'})
neighbormap = dict(zip(nn['SRC_GEOID'], nn['neighbors']))

#Function that adds a 1 or 0 to each tweet if a tract is a neighbor or not of␣
 ↪home_tract
def neighbor_check(x):
    if np.any(neighbormap.get(x['home_tract']) != None):
        if x['tract'] in neighbormap.get(x['home_tract']):
            return 1
        else:
            return 0
    else:
        return 2
#Applying the neighbor_check to the dataframe
df['neighbor'] = df.apply(lambda x: neighbor_check(x), axis = 1, meta =␣
 ↪'float64')
df= df[(df["tract"].isin(neighbormap.keys())) & (df["home_tract"].
 ↪isin(neighbormap.keys()))]
#creating the non-neighbor table
other = df[(df['neighbor'] == 0) & (df['tract'] != df['home_tract'])]
```

[21]:
```
other.groupby(['tract']).aggregate({'id': 'count'}).reset_index().
 ↪to_csv('newnn2_bayarea_totaltweets_*.csv')
```

[21]:
```
['newnn2_bayarea_totaltweets_0.csv']
```

[43]:
```
demo_var = pd.read_csv('home_tract_variables.csv')
```

[44]:
```
demo_var = demo_var.rename(
    columns = {'geo_fips': 'home_tract'})
```

[45]:
```
demo_var
```

[45]:

	home_tract	aboverm_per_col15	aboverm_per_nonwhite15	aboverm_empd14	\
0	6001400100	1.0	0.0	0	
1	6001400200	1.0	0.0	1	
2	6001400300	1.0	0.0	1	
3	6001400400	1.0	0.0	1	
4	6001400500	1.0	0.0	1	
5	6001400600	1.0	1.0	1	
6	6001400700	0.0	1.0	1	
7	6001400800	1.0	1.0	1	
8	6001400900	0.0	0.0	1	
9	6001401000	0.0	1.0	1	
10	6001401100	1.0	0.0	1	
11	6001401200	1.0	0.0	1	
12	6001401300	0.0	1.0	1	
13	6001401400	0.0	1.0	1	
14	6001401500	0.0	1.0	1	
15	6001401600	0.0	1.0	1	
16	6001401700	1.0	1.0	1	
17	6001401800	0.0	1.0	1	
18	6001402200	0.0	1.0	0	
19	6001402400	0.0	1.0	1	
20	6001402500	0.0	1.0	1	
21	6001402600	0.0	1.0	1	
22	6001402700	0.0	1.0	1	
23	6001402800	0.0	1.0	1	
24	6001402900	0.0	1.0	1	
25	6001403000	0.0	1.0	1	
26	6001403100	0.0	1.0	1	
27	6001403300	1.0	1.0	1	
28	6001403400	1.0	1.0	1	
29	6001403501	1.0	1.0	1	
...	...	...	...	...	
2101	6113010505	1.0	0.0	0	
2102	6113010508	1.0	0.0	0	
2103	6113010509	1.0	0.0	0	
2104	6113010510	1.0	0.0	0	
2105	6113010511	1.0	0.0	1	
2106	6113010512	1.0	0.0	0	
2107	6113010513	1.0	0.0	0	
2108	6113010602	1.0	0.0	1	
2109	6113010605	1.0	0.0	0	
2110	6113010606	1.0	0.0	1	
2111	6113010607	1.0	0.0	0	
2112	6113010608	1.0	1.0	1	
2113	6113010701	1.0	0.0	1	
2114	6113010703	1.0	1.0	1	
2115	6113010704	1.0	0.0	1	

2116	6113010800	0.0	1.0	1
2117	6113010901	0.0	1.0	1
2118	6113010902	0.0	1.0	1
2119	6113011001	0.0	1.0	1
2120	6113011002	1.0	0.0	1
2121	6113011101	0.0	1.0	1
2122	6113011102	0.0	1.0	1
2123	6113011103	0.0	1.0	0
2124	6113011203	1.0	0.0	1
2125	6113011204	0.0	1.0	0
2126	6113011205	0.0	1.0	1
2127	6113011206	0.0	0.0	0
2128	6113011300	0.0	1.0	0
2129	6113011400	0.0	1.0	0
2130	6113011500	0.0	1.0	0

	aboverm_per_nhblk15	aboverm_per_asian15	aboverm_density15	\
0	1.0	0.0	0	
1	0.0	0.0	1	
2	1.0	0.0	1	
3	1.0	0.0	1	
4	1.0	0.0	1	
5	1.0	0.0	1	
6	1.0	0.0	1	
7	1.0	0.0	1	
8	1.0	0.0	1	
9	1.0	0.0	1	
10	1.0	0.0	1	
11	1.0	0.0	1	
12	1.0	0.0	1	
13	1.0	0.0	1	
14	1.0	0.0	1	
15	1.0	0.0	1	
16	1.0	0.0	0	
17	1.0	0.0	1	
18	1.0	0.0	1	
19	1.0	0.0	1	
20	1.0	0.0	1	
21	1.0	1.0	1	
22	1.0	0.0	1	
23	1.0	0.0	1	
24	1.0	1.0	1	
25	0.0	1.0	1	
26	1.0	1.0	1	
27	1.0	1.0	1	
28	1.0	0.0	1	
29	1.0	0.0	1	

```
...                  ...                  ...                 ...
2101                 0.0                  1.0                   0
2102                 1.0                  0.0                   0
2103                 0.0                  1.0                   1
2104                 1.0                  1.0                   0
2105                 0.0                  1.0                   1
2106                 1.0                  1.0                   1
2107                 0.0                  1.0                   1
2108                 1.0                  1.0                   1
2109                 0.0                  1.0                   0
2110                 1.0                  1.0                   1
2111                 1.0                  1.0                   0
2112                 1.0                  1.0                   1
2113                 0.0                  0.0                   1
2114                 1.0                  1.0                   1
2115                 1.0                  1.0                   1
2116                 0.0                  0.0                   0
2117                 0.0                  0.0                   1
2118                 1.0                  0.0                   1
2119                 0.0                  0.0                   1
2120                 0.0                  0.0                   0
2121                 0.0                  0.0                   1
2122                 0.0                  0.0                   1
2123                 0.0                  1.0                   1
2124                 0.0                  0.0                   0
2125                 0.0                  0.0                   0
2126                 0.0                  1.0                   1
2127                 0.0                  1.0                   0
2128                 0.0                  0.0                   0
2129                 1.0                  0.0                   0
2130                 1.0                  0.0                   0

      aboverm_per_hisp15                                      disp_type  \
0                    0.0                       MHI - At Risk of Exclusion
1                    0.0                          MHI - Ongoing Exclusion
2                    0.0                      LI - Ongoing Gentrification
3                    0.0                          Advanced Gentrification
4                    0.0                      LI - Ongoing Gentrification
5                    0.0                      LI - Ongoing Gentrification
6                    0.0  LI - Not Losing Low Income Households or Very ...
7                    0.0                      LI - Ongoing Gentrification
8                    0.0                      LI - Ongoing Gentrification
9                    1.0                      LI - Ongoing Gentrification
10                   0.0                      LI - Ongoing Gentrification
11                   1.0                          Advanced Gentrification
12                   0.0                      LI - Ongoing Gentrification
13                   1.0                      LI - Ongoing Gentrification
```

14	0.0	LI - Ongoing Gentrification
15	0.0	LI - At Risk of Gentrification and/or Displace...
16	1.0	LI - Ongoing Gentrification
17	1.0	LI - Ongoing Gentrification
18	1.0	LI - Ongoing Gentrification
19	0.0	LI - Ongoing Gentrification
20	0.0	LI - Ongoing Gentrification
21	0.0	LI - At Risk of Gentrification and/or Displace...
22	0.0	LI - At Risk of Gentrification and/or Displace...
23	0.0	LI - Ongoing Gentrification
24	0.0	LI - Ongoing Gentrification
25	0.0	LI - Ongoing Gentrification
26	0.0	LI - Ongoing Gentrification
27	0.0	LI - Ongoing Gentrification
28	0.0	LI - Ongoing Gentrification
29	0.0	LI - At Risk of Gentrification and/or Displace...
...	...	...
2101	0.0	MHI - At Risk of Exclusion
2102	0.0	MHI - At Risk of Exclusion
2103	0.0	MHI - At Risk of Exclusion
2104	0.0	LI - Not Losing Low Income Households or Very ...
2105	0.0	MHI - Ongoing Exclusion
2106	0.0	LI - Not Losing Low Income Households or Very ...
2107	0.0	LI - Not Losing Low Income Households or Very ...
2108	0.0	LI - Not Losing Low Income Households or Very ...
2109	0.0	MHI - Ongoing Exclusion
2110	0.0	MHI - At Risk of Exclusion
2111	0.0	MHI - At Risk of Exclusion
2112	0.0	College Town
2113	0.0	LI - Not Losing Low Income Households or Very ...
2114	0.0	College Town
2115	0.0	LI - Not Losing Low Income Households or Very ...
2116	1.0	LI - Ongoing Gentrification
2117	1.0	LI - Not Losing Low Income Households or Very ...
2118	1.0	LI - Not Losing Low Income Households or Very ...
2119	1.0	LI - Not Losing Low Income Households or Very ...
2120	1.0	MHI - At Risk of Exclusion
2121	1.0	LI - Not Losing Low Income Households or Very ...
2122	1.0	LI - Not Losing Low Income Households or Very ...
2123	1.0	Advanced Gentrification
2124	0.0	MHI - At Risk of Exclusion
2125	1.0	MHI - Not Losing Low Income Households or Very...
2126	1.0	MHI - Not Losing Low Income Households or Very...
2127	1.0	MHI - At Risk of Exclusion
2128	1.0	Advanced Gentrification
2129	1.0	LI - At Risk of Gentrification and/or Displace...
2130	1.0	LI - At Risk of Gentrification and/or Displace...

	LI_under80AMI	HI_above120AMI	MI_80_120AMI
0	0	1	0
1	0	1	0
2	0	0	1
3	0	1	0
4	0	0	1
5	0	0	1
6	1	0	0
7	0	0	1
8	1	0	0
9	1	0	0
10	0	0	1
11	0	0	1
12	1	0	0
13	1	0	0
14	1	0	0
15	1	0	0
16	0	0	1
17	1	0	0
18	1	0	0
19	1	0	0
20	1	0	0
21	1	0	0
22	1	0	0
23	1	0	0
24	1	0	0
25	1	0	0
26	1	0	0
27	1	0	0
28	1	0	0
29	1	0	0
...	...	...	...
2101	0	1	0
2102	0	0	1
2103	0	0	1
2104	0	0	1
2105	0	1	0
2106	0	1	0
2107	0	0	1
2108	1	0	0
2109	0	1	0
2110	0	1	0
2111	0	1	0
2112	0	0	1
2113	1	0	0
2114	1	0	0

```
2115              0              0              1
2116              1              0              0
2117              0              0              1
2118              1              0              0
2119              0              0              1
2120              0              1              0
2121              0              0              1
2122              1              0              0
2123              0              0              1
2124              0              1              0
2125              0              1              0
2126              0              1              0
2127              0              1              0
2128              0              0              1
2129              0              0              1
2130              0              0              1

[2131 rows x 12 columns]
```

[46]:
```
#Joining the non-neighbor table with the home tract variables csv (demo_var)
other = other.merge(demo_var, on = 'home_tract', how = 'inner')
```

[99]:
```
other.groupby(['tract']).aggregate({'id': 'count'}).reset_index().
↪to_csv('new_nn_bayarea_totaltweets_*.csv')
```

[99]:
```
['new_nn_bayarea_totaltweets_0.csv']
```

[100]:
```
df[df['neighbor'] == 1].groupby(['tract']).aggregate({'id': 'count'}).
↪reset_index().to_csv('neighborbayarea_totaltweets_*.csv')
```

[100]:
```
['neighborbayarea_totaltweets_0.csv']
```

[ ]:
```
df[df['home_tract'] == df['tract']].groupby(['tract']).aggregate({'id':␣
↪'count'}).reset_index().to_csv('localbayarea_totaltweets_*.csv')
```

0.1 Aggregation:

Creating csv files for all the numbers and percentages of tweets sent from non-neighbor users in a tract with specific demographic characteristics.

[47]:
```
%%time
#aboverm_per_col15
aboverm_per_col15 = other.groupby(['tract', 'aboverm_per_col15']).aggregate(
    {'u_id': 'count'}).reset_index()
aboverm_per_col15 = aboverm_per_col15.categorize(columns = 'aboverm_per_col15')
aboverm_per_col15 = aboverm_per_col15.pivot_table(values = 'u_id',
                                                  columns = 'aboverm_per_col15',
                                                  index = 'tract')
```

```
CPU times: user 56min 18s, sys: 4min 28s, total: 1h 47s
Wall time: 54min 4s
```

[48]:
```
#Function to take the variable and return the csv of the counts of non-neighbor␣
 ↪tweets with that condition
def create_csv(demo_variable):
    g = other.groupby(['tract', demo_variable]).aggregate(
        {'u_id': 'count'}).reset_index()
    g = g.categorize(columns = demo_variable)
    g = g.pivot_table(values = 'u_id',
                      columns = demo_variable,
                      index = 'tract')
    g.to_csv(demo_variable + '_*.csv')
```

[49]:
```
%%time
create_csv('aboverm_per_nonwhite15')
```

```
CPU times: user 1h 52min 31s, sys: 8min 37s, total: 2h 1min 9s
Wall time: 1h 48min 3s
```

[50]:
```
%%time
create_csv('aboverm_empd14')
```

```
CPU times: user 1h 51min 14s, sys: 8min 5s, total: 1h 59min 19s
Wall time: 1h 46min 22s
```

[51]:
```
%%time
create_csv('aboverm_per_nhblk15')
```

```
CPU times: user 1h 45min 4s, sys: 7min 10s, total: 1h 52min 14s
Wall time: 1h 40min 12s
```

[52]:
```
%%time
create_csv('aboverm_per_asian15')
```

```
CPU times: user 1h 48min 4s, sys: 7min 52s, total: 1h 55min 56s
Wall time: 1h 43min 18s
```

[53]:
```
%%time
create_csv('aboverm_density15')
```

```
CPU times: user 1h 48min 27s, sys: 7min 53s, total: 1h 56min 21s
Wall time: 1h 43min 40s
```

[54]:
```
%%time
create_csv('aboverm_per_hisp15')
```

```
CPU times: user 1h 46min 54s, sys: 7min 35s, total: 1h 54min 30s
Wall time: 1h 42min 9s
```

[55]:
```
%%time
create_csv('LI_under80AMI')
```

```
CPU times: user 1h 47min 39s, sys: 7min 48s, total: 1h 55min 28s
Wall time: 1h 42min 59s
```

[56]:
```
%%time
create_csv('HI_above120AMI')
```

```
CPU times: user 1h 46min 25s, sys: 7min 32s, total: 1h 53min 58s
Wall time: 1h 41min 41s
```

[57]:
```
%%time
create_csv('MI_80_120AMI')
```

```
CPU times: user 1h 48min 9s, sys: 7min 33s, total: 1h 55min 43s
Wall time: 1h 43min 18s
```

[58]:
```
%%time
create_csv('disp_type')
```

```
CPU times: user 1h 47min 47s, sys: 7min 38s, total: 1h 55min 25s
Wall time: 1h 42min 50s
```

[59]:
```
%%time
aboverm_per_col15.to_csv('aboverm_per_col15_*.csv')
```

```
CPU times: user 53min 8s, sys: 3min 37s, total: 56min 46s
Wall time: 50min 41s
```

[59]:
```
['aboverm_per_col15_0.csv']
```

[60]:
```
[i + '_0.csv' for i in demo_var.columns if i != 'home_tract']
```

[60]:
```
['aboverm_per_col15_0.csv',
 'aboverm_per_nonwhite15_0.csv',
 'aboverm_empd14_0.csv',
 'aboverm_per_nhblk15_0.csv',
 'aboverm_per_asian15_0.csv',
 'aboverm_density15_0.csv',
 'aboverm_per_hisp15_0.csv',
 'disp_type_0.csv',
 'LI_under80AMI_0.csv',
 'HI_above120AMI_0.csv',
 'MI_80_120AMI_0.csv']
```

Question: Are tweets sent from users in neighboring tracts sent in tracts that are considered commercial or residential?

0.2 OUTPUT FILE

Creating the final result table.

Labeling the columns with its correct name

Dividing the count by total tweets to get percentages

[7]:
```
col15 = pd.read_csv('aboverm_per_col15_0.csv') # 0 and 1
nonwhite = pd.read_csv('aboverm_per_nonwhite15_0.csv') # 0 and 1
empd = pd.read_csv( 'aboverm_empd14_0.csv') # 0 and 1
nhblk = pd.read_csv( 'aboverm_per_nhblk15_0.csv') # 0 and 1
asian = pd.read_csv('aboverm_per_asian15_0.csv') # 0 and 1
density = pd.read_csv('aboverm_density15_0.csv') # 0 and 1
hisp = pd.read_csv('aboverm_per_hisp15_0.csv') # 0 and 1
disptype = pd.read_csv('disp_type_0.csv') # 9 variables
under80 = pd.read_csv('LI_under80AMI_0.csv') # 0 and 1
above120 = pd.read_csv('HI_above120AMI_0.csv') # 0 and 1
MI = pd.read_csv('MI_80_120AMI_0.csv') # 0 and 1
totaltweets= pd.read_csv('new_nn_bayarea_totaltweets_0.csv')
```

[8]:
```
MI = MI[['tract', '1']]
totaltweets = totaltweets[['tract', 'id']].rename({'id': 'total_tweets'}, axis =␣
↪1)
MI = MI.rename({'1': 'ct_othertweets_MI_80_120AMI'}, axis =1)
above120 = above120[['tract', '1']].rename({'1':␣
↪'ct_othertweets_HI_above_120AMI'}, axis =1)
under80 = under80[['tract', '1']].rename({'1': 'ct_othertweets_LI_under80AMI'},␣
↪axis =1)
disptype.columns = ['tract'] + ['ct_othertweets_' + k for k in disptype.columns␣
↪if k != 'tract']
hisp = hisp.rename({'1.0': 'ct_othertweets_aboverm_per_hisp15',
                    '0.0': 'ct_othertweets_underm_per_hisp15'}, axis =1)
density = density.rename({'1': 'ct_othertweets_aboverm_density15',
                    '0': 'ct_othertweets_underm_density15'}, axis =1)
asian = asian.rename({'1.0': 'ct_othertweets_aboverm_per_asian15',
                      '0.0': 'ct_othertweets_underm_per_asian15'}, axis =1)
nhblk = nhblk.rename({'1.0': 'ct_othertweets_aboverm_per_nhblk15',
                      '0.0': 'ct_othertweets_underm_per_nhblk15'}, axis = 1)
empd = empd.rename({'1': 'ct_othertweets_aboverm_empd14',
                    '0': 'ct_othertweets_underm_empd14'}, axis = 1)
nonwhite = nonwhite.rename({'1.0': 'ct_othertweets_aboverm_per_nonwhite15',
                      '0.0': 'ct_othertweets_underm_per_nonwhite15'}, axis = 1)
col15 = col15.rename({'1.0': 'ct_othertweets_aboverm_per_col15',
                      '0.0': 'ct_othertweets_underm_per_col15'}, axis =1)
```

[9]:
```
twitter_demo_sf= MI.merge(above120, on = 'tract', how = 'inner').merge(
under80, on = 'tract', how ='inner').merge(
disptype, on ='tract', how ='inner').merge(
hisp, on ='tract', how ='inner').merge(
density, on ='tract', how ='inner').merge(
```

```
asian, on ='tract', how = 'inner').merge(
nhblk, on = 'tract', how ='inner').merge(
empd, on ='tract', how ='inner').merge(
nonwhite, on ='tract', how = 'inner').merge(
col15, on ='tract', how = 'inner').merge(
totaltweets, on = 'tract', how = 'inner')
```

[10]:
```
twitter_demo_sf = twitter_demo_sf.fillna(0)
twitter_demo_sf['total_nonneighbortweets'] =␣
 ↪twitter_demo_sf['ct_othertweets_HI_above_120AMI'] +␣
 ↪twitter_demo_sf['ct_othertweets_LI_under80AMI'] +␣
 ↪twitter_demo_sf['ct_othertweets_MI_80_120AMI']
for i in twitter_demo_sf.columns:
    if i not in ['total_nonneighbortweets', 'tract']:
        twitter_demo_sf['%_'+ i[2:len(i)]] = twitter_demo_sf[i]/
 ↪twitter_demo_sf['total_nonneighbortweets']
```

[ ]:
```
twitter_demo_sf[['total_tweets', 'total_nonneighbortweets']]
```

[73]:
```
twitter_demo_sf.to_csv('Twitter_NonNeighborBayArea_demog_10_23.csv')
```

[16]:
```
outside = pd.read_csv('nnfromoutside_bayarea_totaltweets_0.csv')
```

[3]:
```
pf = pd.read_csv(rawfile)
```

[4]:
```
sf_nn = pd.read_csv('sfbay_13county_nearest_neigbbor_new.csv')
alameda = []
for i in sf_nn['SRC_GEOID'].unique():
    if 6001000000 <= i & i < 6002000000:
        alameda.append(i)
```

[5]:
```
alamedatwitter = df[(df['tract'] < 6002000000.0) & (df['tract'] >= 6001000000.0)]
```

[ ]:
```
%%time
alamedatwitter.to_csv('alamedatwitter_*.csv')
```

[28]:
```
import reverse_geocoder as rg
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)

<ipython-input-28-53e6315bf5a1> in <module>
----> 1 import reverse_geocoder as rg


ImportError: No module named 'reverse_geocoder'
```

[26]:
```
%%time
pro = [str(k) for k in list(Path('/scratch/public/catherinebui/SF Profiles').
 ↪glob('*.csv'))]
p0= pd.read_csv(pro[0])
# p0 = p0.merge(sf_o, on = 'id', how = 'inner')
# p0 =  p0[(p0['tract'] < 6002000000.0) & (p0['tract'] >= 6001000000.0)]
```

```
CPU times: user 14.4 s, sys: 1.53 s, total: 16 s
Wall time: 16 s
```

[41]:
```
np.count_nonzero(p0.groupby(
    ['u_id', 'u_location']).aggregate(
    {'u_id' : 'count'}).rename(
    {'u_id': 'count'}, axis =1).reset_index().groupby(
    'u_id').count()['u_location'] > 1)
```

[41]:
```
962
```

[27]:
```
p1=pd.read_csv(pro[1])
# p1['id'] = p1['id'].astype(float)
# p1 = p1.merge(sf_o, on = 'id', how = 'inner')
# # p1 =  p1[(p1['tract'] < 6002000000.0) & (p1['tract'] >= 6001000000.0)]

p2=pd.read_csv(pro[2])
# p2['id'] = p2['id'].astype(float)
# p2 = p2.merge(sf_o, on = 'id', how = 'inner')
# p2 =  p2[(p2['tract'] < 6002000000.0) & (p2['tract'] >= 6001000000.0)]
```

[28]:
```
%%time
p3=pd.read_csv(pro[3])
# p3['id'] = p3['id'].astype(float)
# p3 = p3.merge(sf_o, on = 'id', how = 'inner')
# p3 =  p3[(p3['tract'] < 6002000000.0) & (p3['tract'] >= 6001000000.0)]

p4=pd.read_csv(pro[4])
# p4['id'] = p4['id'].astype(float)
# p4 = p4.merge(sf_o, on = 'id', how = 'inner')
# p4 =  p4[(p4['tract'] < 6002000000.0) & (p4['tract'] >= 6001000000.0)]
```

```
CPU times: user 35.4 s, sys: 2.85 s, total: 38.2 s
Wall time: 38.3 s
```

[29]:
```
pro[4], pro[5]
```

[29]:
```
('/scratch/public/catherinebui/SF Profiles/SANFRANCISCO_2014-01.csv',
 '/scratch/public/catherinebui/SF Profiles/SANFRANCISCO_2014-06.csv')
```

[30]:
```
p5=pd.read_csv(pro[5])
# p5['id'] = p5['id'].astype(float)
```

```
# p5 = p5.merge(sf_o, on = 'id', how = 'inner')
# p5 =  p5[(p5['tract'] < 6002000000.0) & (p5['tract'] >= 6001000000.0)]

p6=pd.read_csv(pro[6], encoding='iso-8859-1')
# p6 = p6.merge(sf_o, on = 'id', how = 'inner')
# p6 =  p6[(p6['tract'] < 6002000000.0) & (p6['tract'] >= 6001000000.0)]
```

[31]:
```
p7=pd.read_csv(pro[7], encoding='iso-8859-1')
# p7['id'] = p7['id'].astype(float)
# p7 = p7.merge(sf_o, on = 'id', how = 'inner')
# p7 =  p7[(p7['tract'] < 6002000000.0) & (p7['tract'] >= 6001000000.0)]

p8=pd.read_csv(pro[8])
# p8['id'] = p8['id'].astype(float)
# p8 = p8.merge(sf_o, on = 'id', how = 'inner')
# p8 =  p8[(p8['tract'] < 6002000000.0) & (p8['tract'] >= 6001000000.0)]
```

[32]:
```
p9=pd.read_csv(pro[9])
# p9['id'] = p9['id'].astype(float)
# p9 = p9.merge(sf_o, on = 'id', how = 'inner')
# p9 =  p9[(p9['tract'] < 6002000000.0) & (p9['tract'] >= 6001000000.0)]

p10=pd.read_csv(pro[10])
# p10['id'] = p10['id'].astype(float)
# p10 = p10.merge(sf_o, on = 'id', how = 'inner')
# p10 =  p10[(p10['tract'] < 6002000000.0) & (p10['tract'] >= 6001000000.0)]
```

[33]:
```
p11=pd.read_csv(pro[11])
# p11['id'] = p11['id'].astype(float)
# p11 = p11.merge(sf_o, on = 'id', how = 'inner')
# p11 =  p11[(p11['tract'] < 6002000000.0) & (p11['tract'] >= 6001000000.0)]

p12=pd.read_csv(pro[12])
# p12['id'] = p12['id'].astype(float)
# p12 = p12.merge(sf_o, on = 'id', how = 'inner')
# p12 =  p12[(p12['tract'] < 6002000000.0) & (p12['tract'] >= 6001000000.0)]
```

[34]:
```
p13=pd.read_csv(pro[13])
# p13['id'] = p13['id'].astype(float)
# p13 = p13.merge(sf_o, on = 'id', how = 'inner')
# p13 =  p13[(p13['tract'] < 6002000000.0) & (p13['tract'] >= 6001000000.0)]

p14= pd.read_csv(pro[14])
# p14['id'] = p14['id'].astype(float)
# p14 = p14.merge(sf_o, on = 'id', how = 'inner')
# p14 =  p14[(p14['tract'] < 6002000000.0) & (p14['tract'] >= 6001000000.0)]
```

[35]:
```
p15=pd.read_csv(pro[15])
# p15['id'] = p15['id'].astype(float)
```

```
# p15 = p15.merge(sf_o, on = 'id', how = 'inner')
# p15 =  p15[(p15['tract'] < 6002000000.0) & (p15['tract'] >= 6001000000.0)]

p16=pd.read_csv(pro[16])
# p16['id'] = p16['id'].astype(float)
# p16 = p16.merge(sf_o, on = 'id', how = 'inner')
# p16 =  p16[(p16['tract'] < 6002000000.0) & (p16['tract'] >= 6001000000.0)]
```

[36]:
```
%%time
p17=pd.read_csv(pro[17])
# p17['id'] = p17['id'].astype(float)
# p17 = p17.merge(sf_o, on = 'id', how = 'inner')
# p17 =  p17[(p17['tract'] < 6002000000.0) & (p17['tract'] >= 6001000000.0)]

p18=pd.read_csv(pro[18])
# p18['id'] = p18['id'].astype(float)
# p18 = p18.merge(sf_o, on = 'id', how = 'inner')
# p18 =  p18[(p18['tract'] < 6002000000.0) & (p18['tract'] >= 6001000000.0)]
```

```
CPU times: user 20.9 s, sys: 1.8 s, total: 22.7 s
Wall time: 22.8 s
```

[37]:
```
p19=pd.read_csv(pro[19])
# p19['id'] = p19['id'].astype(float)
# p19 = p19.merge(sf_o, on = 'id', how = 'inner')
# p19 =  p19[(p19['tract'] < 6002000000.0) & (p19['tract'] >= 6001000000.0)]

p20=pd.read_csv(pro[20])
# p20['id'] = p20['id'].astype(float)
# p20 = p20.merge(sf_o, on = 'id', how = 'inner')
# p20 =  p20[(p20['tract'] < 6002000000.0) & (p20['tract'] >= 6001000000.0)]
```

[38]:
```
p21=pd.read_csv(pro[21])
# p21['id'] = p21['id'].astype(float)
# p21 = p21.merge(sf_o, on = 'id', how = 'inner')
# p21 =  p21[(p21['tract'] < 6002000000.0) & (p21['tract'] >= 6001000000.0)]

p22=pd.read_csv(pro[22])
# p22['id'] = p22['id'].astype(float)
# p22 = p22.merge(sf_o, on = 'id', how = 'inner')
# p22 =  p22[(p22['tract'] < 6002000000.0) & (p22['tract'] >= 6001000000.0)]
```

[39]:
```
p23=pd.read_csv(pro[23])
# p23['id'] = p23['id'].astype(float)
# p23 = p23.merge(sf_o, on = 'id', how = 'inner')
# p23 =  p23[(p23['tract'] < 6002000000.0) & (p23['tract'] >= 6001000000.0)]

p24=pd.read_csv(pro[24])
# p24['id'] = p24['id'].astype(float)
```

```
# p24 = p24.merge(sf_o, on = 'id', how = 'inner')
# p24 =  p24[(p24['tract'] < 6002000000.0) & (p24['tract'] >= 6001000000.0)]
```

[40]:
```
p25=pd.read_csv(pro[25])
# p25['id'] = p25['id'].astype(float)
# p25 = p25.merge(sf_o, on = 'id', how = 'inner')
# p25 =  p25[(p25['tract'] < 6002000000.0) & (p25['tract'] >= 6001000000.0)]

p26=pd.read_csv(pro[26])
# p26['id'] = p26['id'].astype(float)
# p26 = p26.merge(sf_o, on = 'id', how = 'inner')
# p26 =  p26[(p26['tract'] < 6002000000.0) & (p26['tract'] >= 6001000000.0)]
```

[41]:
```
p27=pd.read_csv(pro[27])
# p27['id'] = p27['id'].astype(float)
# p27 = p27.merge(sf_o, on = 'id', how = 'inner')
# p27 =  p27[(p27['tract'] < 6002000000.0) & (p27['tract'] >= 6001000000.0)]

p28=pd.read_csv(pro[28])
# p28['id'] = p28['id'].astype(float)
# p28 = p28.merge(sf_o, on = 'id', how = 'inner')
# p28 =  p28[(p28['tract'] < 6002000000.0) & (p28['tract'] >= 6001000000.0)]
```

[42]:
```
p29=pd.read_csv(pro[29])
# p29['id'] = p29['id'].astype(float)
# p29 = p29.merge(sf_o, on = 'id', how = 'inner')
# p29 =  p29[(p29['tract'] < 6002000000.0) & (p29['tract'] >= 6001000000.0)]

p30=pd.read_csv(pro[30])
# p30['id'] = p30['id'].astype(float)
# p30 = p30.merge(sf_o, on = 'id', how = 'inner')
# p30 =  p30[(p30['tract'] < 6002000000.0) & (p30['tract'] >= 6001000000.0)]
```

[43]:
```
p31=pd.read_csv(pro[31])
# p31['id'] = p31['id'].astype(float)
# p31 = p31.merge(sf_o, on = 'id', how = 'inner')
# p31 =  p31[(p31['tract'] < 6002000000.0) & (p31['tract'] >= 6001000000.0)]

p32=pd.read_csv(pro[32])
# p32['id'] = p32['id'].astype(float)
# p32 = p32.merge(sf_o, on = 'id', how = 'inner')
# p32 =  p32[(p32['tract'] < 6002000000.0) & (p32['tract'] >= 6001000000.0)]
```

[44]:
```
p33 = pd.read_csv(pro[33])
# p33['id'] = p33['id'].astype(float)
# p33 = p33.merge(sf_o, on = 'id', how = 'inner')
# p33 =  p33[(p33['tract'] < 6002000000.0) & (p33['tract'] >= 6001000000.0)]

p34 = pd.read_csv(pro[34])
```

```
# p34['id'] = p34['id'].astype(float)
# p34 = p34.merge(sf_o, on = 'id', how = 'inner')
# p34 =  p34[(p34['tract'] < 6002000000.0) & (p34['tract'] >= 6001000000.0)]
```

pd.concat([pd.read_csv(pro[i]) for i in np.arange(17, 35, 1)]).to_csv('/scratch/public/catherinebui/prof_17_35

pd.concat([pd.read_csv(pro[i], encoding = 'iso-8859-1') for i in np.arange(0, 35, 1)]).to_csv('/scratch/public/catherinebui/sf_alameda_twitterprofile.csv')

[22]:
```
pd.concat([p0,p1,p2,p3,p4, p5, p6,p7,p8,
          p9,p10,p11,p12,p13,p14,p15,p16,p17, p18,
          p19, p20, p21, p22, p23, p24, p25, p26, p27, p28,
          p29, p30, p31,p32, p33, p34]).to_csv('/scratch/public/catherinebui/
 ↪full_twitterprofile.csv')
```

[6]:
```
tweets = pd.read_csv('/scratch/public/catherinebui/sf_with_homeloc_scipen_raw.
 ↪csv')
tweets = tweets.rename(columns = {'sf_with_homeloc_scipen.csv': 'id'})
```

[7]:
```
tweets['year'] = tweets['date'].apply(lambda x: str(x)[0:4])
tweets.groupby('year').count()
```

[7]:

year	id	u_id	lat	lon	date	tract	home_tract
2012	4955602	4955602	4955602	4955602	4955602	4944324	4953638
2013	16405119	16405119	16405119	16405119	16405119	16381568	16402688
2014	20638660	20638660	20638660	20638660	20638660	20616554	20636634

[24]:
```
nfo = ['6081610500',
'6081610601'
'6081610602']
```

[ ]:
```
%%time
profiles = [p0,p1,p2,p3,p4,p5, p6,p7,p8,
          p9,p10,p11,p12,p13,p14,p15,p16,p17, p18,
          p19, p20, p21, p22, p23, p24, p25, p26, p27, p28,
          p29, p30, p31,p32, p33, p34]
newdf = []
for prof in profiles:
    merged = tweets.merge(prof, on = 'id', how = 'inner')
    merged['county'] = merged['tract'].apply(lambda x: str(x)[0:4] if str(x)[0] !
 ↪= '9' else str(x)[0:4])
    merged['2015'] = merged['date'].apply(lambda x: x[0:4] == '2015')
    menlopark = merged[merged['2014'] == True]
#     menlopark = merged[merged['tract'] == 6081611700.0]
    newdf.append(menlopark)
    print('1')

menloparkbypart = pd.concat(newdf)

#then do p5 for dask
```

[26]:
```
menloparkbypart.to_csv('/scratch/public/catherinebui/20_2_28.csv', index=False)
```

[22]:
```
profiless.to_csv('/scratch/public/catherinebui/first_half.csv', index=False)
```

[127]:
```
first = das.read_csv('/scratch/public/catherinebui/first_half.csv',␣
 ↪dtype={'home_tract': 'float64', 'created_at': 'float64', 'id': 'object',
       'lat_x': 'object',
       'u_created_at': 'float64',
       'u_followers_count': 'float64',
       'u_id_y': 'float64',
       'u_statuses_count': 'float64'}, encoding='utf-8', engine='c')
```

[106]:
```
new = []
new.append(first)
for prof in [p4]:
    new.append(tweetd.merge(prof, on = 'id', how = 'inner'))
full_profile = das.concat(new)
```

[37]:
```
menlo = ['613900', '612500', '611500', '611600',  '611700']
nfo = ['610601', '610602', '518400', '610500', '610600']
epa = ['611800', '611900', '612000', '612100', '611800']
```

[167]:
```
first['county'] = first['tract'].apply(lambda x: str(x)[0:4] if str(x)[0] != '9'␣
 ↪else str(x)[0:4])
menlopark = first[first['county'].isin(menlo)]
```

[26]:
```
bs2 = pd.read_csv('/scratch/public/catherinebui/bs2.csv')
```

[27]:
```
menloparkcsv= pd.read_csv('/scratch/public/catherinebui/menlopark_6081611700.
 ↪csv')
```

[28]:
```
menloparkcsv= menloparkcsv.rename(columns = {'lat_x': 'lat', 'lon_x': 'lon'})
bs2 = bs2.rename(columns = {'PropertyAddressLatitude': 'lat',␣
 ↪'PropertyAddressLongitude': 'lon'})
```

[29]:
```
menloparkcsv['lat'] = menloparkcsv['lat'].apply(lambda x: float(str(x)[0:7]))
menloparkcsv['lon'] = menloparkcsv['lon'].apply(lambda x: float(str(x)[0:8]))
```

[216]:
```
type(bs2['lat'][0])
```

[216]:
```
numpy.float64
```

[217]:
```
menloparkcsv['lat'][0]
```

[217]:
```
37.4786
```

[30]:
```
menloparkcsv.merge(bs2, on = ['lat', 'lon'], how = 'inner').
 ↪to_csv('menlopark_6081611700_property.csv')
```

[4]:
```
area_6081 = pd.read_csv('/scratch/public/catherinebui/menlopark6081_2_5.csv',␣
 ↪engine = 'python')
```

[30]:
```
area_6081['county6digits'] = area_6081['tract'].apply(lambda x: str(x)[4:-2])
```

[6]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stopwords = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /accounts/projects/mzuk/bui_catherine/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /accounts/projects/mzuk/bui_catherine/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

[ ]:
```
%%time
at6081 = area_6081[area_6081['text'].str.contains('@', na = False)]
username6081 = []
for name in at6081['u_screen_name'].unique():
    username6081.append(name)
at6081['text']= at6081['text'].apply(lambda x: [word for word in␣
 ↪word_tokenize(str(x)) if word not in stopwords])
np.count_nonzero(at6081['text'].apply(lambda x: len([word for word in x if word␣
 ↪in username6081])))
```

```
/usr/local/linux/anaconda3.7/lib/python3.7/site-
packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  """
```

[60]:
```
nfo = area_6081[area_6081['county6digits'].isin(nfo)]
mp = area_6081[area_6081['county6digits'].isin(menlo)]
eastpalo = area_6081[area_6081['county6digits'].isin(epa)]
```

[62]:
```
nfo.to_csv('northfairoaks_prof.csv')
```

[58]:
```
mp.to_csv('menloparkspecifics_prof.csv')
```

[61]:
```
eastpalo.to_csv('eastpaloalto_prof.csv')
```

1 text analysis

1. tokenizing the word-splitting the sentences by word based on whitespace and punctuation.
2. removing stop words

3. replacing html characters
4. removing punctuation but not '@'
5. normalizing the case
6. removing urls

[46]:
```
profiles = [p0,p1,p2,p3,p4,p5, p6,p7,p8,
          p9,p10,p11,p12,p13,p14,p15,p16,p17, p18,
          p19, p20, p21, p22, p23, p24, p25, p26, p27, p28,
          p29, p30, p31,p32, p33, p34]
```

[47]:
```
#removing stop words
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stopwords = set(stopwords.words('english'))
newdf = []
n=0
for prof in profiles:
    merged = tweets.merge(prof, on = 'id', how = 'inner')
    merged['text'] = merged['text'].apply(lambda x: [word for word in␣
 ↪word_tokenize(str(x)) if word not in stopwords])
    newdf.append(merged)
    print(str(n))
    n+=1

output = pd.concat(newdf)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /accounts/projects/mzuk/bui_catherine/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /accounts/projects/mzuk/bui_catherine/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
0
1
2
3
4
5
6
7
8
9
10
```

[ ]:
```
#compare the word distribution between local tweets and nonlocal tweets
notlocal = output[output['tract'] != output['home_tract']]
local = output[output['tract'] == output['home_tract']]

#what tools can we use to figure out if a user is a business

#find out employment from user description
```

[ ]:
```
output['text'][0][0]
```

[ ]:
```
%%time
#find a list of unique usernames for all data
userscreenname = []
for prof in profiles:
    for name in prof['u_screen_name'].unique():
        userscreenname.append(name)
userscreenname
networks = []
n=0
for prof in profiles:
    merged = tweets.merge(prof, on = 'id', how = 'inner')
    print('complete merge')
    merged = merged[merged['text'].str.contains('@', na = False)]
```

```
    print('complete @')
    merged['text'] = merged.apply(lambda x: [word for word in word_tokenize(x)␣
 ↪if word not in stopwords])
    merged['network'] = merged.apply(lambda x: str(x['text']) in userscreenname,␣
 ↪axis = 1)
    networks.append(merged)
    print(str(n))
    n+=1

output2 = pd.concat(networks)
```

[ ]:
```
output2
```

[ ]:
```
output2.to_csv('/scratch/public/catherinebui/networks_twitter.csv')
```

[48]:
```
output.to_csv('/scratch/public/catherinebui/tokenize_twitter.csv')
```

[ ]:
```

```