cogs9_proj

December 15, 2020

1 Cogs9 Project: Does the SD police racially profile

Group Name: CV GANG
Data Sources: - Races - Stops - Census

By Andrew Cheng

[30]:

```
#Modules
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import datetime

%matplotlib inline
plt.style.use('fivethirtyeight')
```

[2]:

```
data_race_raw = pd.read_csv('ripa_race_datasd.csv')
data_race_raw
```

[2]:

```
        stop_id  pid                     race
0          2443    1                    White
1          2444    1                    White
2          2447    1       Hispanic/Latino/a
3          2447    2       Hispanic/Latino/a
4          2448    1                    White
…             …  …                        …
394970   356019    1  Black/African American
394971   356025    1  Black/African American
394972   356080    1                    White
394973   356300    1  Black/African American
394974   356303    1  Black/African American

[394975 rows x 3 columns]
```

[3]:

```
data_stops_raw = pd.read_csv('ripa_stops_datasd.csv', low_memory = False)
data_stops_raw
```

[3]:

```
        stop_id        ori agency  exp_years   date_stop time_stop  \
0          2443  CA0371100      SD         10  2018-07-01  00:01:37
1          2444  CA0371100      SD         18  2018-07-01  00:03:34
2          2447  CA0371100      SD          1  2018-07-01  00:05:43
3          2447  CA0371100      SD          1  2018-07-01  00:05:43
4          2448  CA0371100      SD          3  2018-07-01  00:19:06
…         …        …    …        …         …       …
391129   356019  CA0371100      SD          1  2020-09-30  23:05:00
391130   356025  CA0371100      SD          1  2020-09-30  23:38:00
391131   356080  CA0371100      SD         18  2020-09-30  15:31:00
391132   356300  CA0371100      SD         18  2020-09-30  19:30:00
391133   356303  CA0371100      SD          1  2020-09-30  19:37:52

        stopduration  stop_in_response_to_cfs  officer_assignment_key  \
0                 30                        0                       1
1                 10                        0                       1
2                 15                        1                      10
3                 15                        1                      10
4                  5                        0                       1
…              …                      …                     …
391129             7                        1                       1
391130            30                        1                       1
391131             5                        0                       1
391132           180                        1                       1
391133            45                        0                       1

                                           assignment  …  \
0       Patrol, traffic enforcement, field operations  …
1       Patrol, traffic enforcement, field operations  …
2                                               Other  …
3                                               Other  …
4       Patrol, traffic enforcement, field operations  …
…                                               …  …
391129  Patrol, traffic enforcement, field operations  …
391130  Patrol, traffic enforcement, field operations  …
391131  Patrol, traffic enforcement, field operations  …
391132  Patrol, traffic enforcement, field operations  …
391133  Patrol, traffic enforcement, field operations  …

                 beat_name  pid  isstudent  perceived_limited_english  \
0         Pacific Beach 122    1          0                          0
1         Mission Beach 121    1          0                          0
2            El Cerrito 822    1          0                          0
3            El Cerrito 822    2          0                          0
4           Ocean Beach 614    1          0                          0
…                      …  …       …                       …
391129       Harborview 527    1          0                          0
```

```
391130    Core-Columbia 524    1         0                         0
391131           Unknown 999    1         0                         0
391132  Carmel Mountain 232    1         0                         0
391133      Golden Hill 517    1         0                         0

       perceived_age  perceived_gender gender_nonconforming gend  gend_nc  \
0                 25              Male                    0    1      NaN
1                 25              Male                    0    1      NaN
2                 30              Male                    0    1      NaN
3                 30            Female                    0    2      NaN
4                 23              Male                    0    1      NaN
…              …               …                  …  …      …
391129            50            Female                    0    2      NaN
391130            35              Male                    0    1      NaN
391131            60              Male                    0    1      NaN
391132            25              Male                    0    1      NaN
391133            28              Male                    0    1      NaN

       perceived_lgbt
0                  No
1                  No
2                  No
3                  No
4                  No
…               …
391129             No
391130             No
391131             No
391132             No
391133             No

[391134 rows x 29 columns]
```

[4]:

```
data_census_race = pd.DataFrame({'percentage of population': [42.8,6.4,30.3,2.9,16.7,.5,.4]},
index = ['White','Black/African American','Hispanic/Latino/a','Middle Eastern or South Asian','Asian','Native American','Pacific Islander'])

data_census_race
```

[4]:

```
                               percentage of population
White                                              42.8
Black/African American                              6.4
Hispanic/Latino/a                                  30.3
Middle Eastern or South Asian                       2.9
Asian                                              16.7
Native American                                     0.5
```

```
Pacific Islander                                    0.4
```

[5]:

```
data_race = data_race_raw.set_index('stop_id')
data_race
```

[5]:

```
         pid                    race
stop_id
2443       1                   White
2444       1                   White
2447       1       Hispanic/Latino/a
2447       2       Hispanic/Latino/a
2448       1                   White
…        …                       …
356019     1  Black/African American
356025     1  Black/African American
356080     1                   White
356300     1  Black/African American
356303     1  Black/African American

[394975 rows x 2 columns]
```

[6]:

```
data_date = pd.DataFrame().assign(date = data_stops_raw.get('date_stop'),␣
 ↪stop_id = data_stops_raw.get('stop_id')).set_index('stop_id')
data_date
```

[6]:

```
               date
stop_id
2443     2018-07-01
2444     2018-07-01
2447     2018-07-01
2447     2018-07-01
2448     2018-07-01
…               …
356019   2020-09-30
356025   2020-09-30
356080   2020-09-30
356300   2020-09-30
356303   2020-09-30

[391134 rows x 1 columns]
```

[7]:

```
#Merge race data set with the dates from the stop data set with the stop_id
data_merged = data_race.merge(data_date,left_index = True, right_index = True)
data_merged
```

[7]:

```
         pid                    race        date
stop_id
```

```
2443      1                   White  2018-07-01
2444      1                   White  2018-07-01
2447      1       Hispanic/Latino/a  2018-07-01
2447      1       Hispanic/Latino/a  2018-07-01
2447      2       Hispanic/Latino/a  2018-07-01
…         …                      …           …
356019    1  Black/African American  2020-09-30
356025    1  Black/African American  2020-09-30
356080    1                   White  2020-09-30
356300    1  Black/African American  2020-09-30
356303    1  Black/African American  2020-09-30

[595128 rows x 3 columns]
```

[42]:

```
#Remove Duplicates and include data within subjected time interval
data_final = data_merged.drop_duplicates()

#Get the year from the date string
def to_year(date):
    dt = datetime.datetime.strptime(date,'%Y-%m-%d')
    return dt.year

data_final = data_final[data_final.get('date').apply(to_year) == 2019]
data_final
```

[42]:

```
        pid                           race        date
stop_id
84362     1              Hispanic/Latino/a  2019-01-01
84364     1                          White  2019-01-01
84369     1         Black/African American  2019-01-01
84372     2              Hispanic/Latino/a  2019-01-01
84376     1  Middle Eastern or South Asian  2019-01-01
…         …                             …           …
254761    8                          White  2019-12-31
254771    2                          White  2019-12-31
254776    1                Native American  2019-12-31
255002    4                          White  2019-12-31
255002    5                          White  2019-12-31

[8398 rows x 3 columns]
```

[43]:

```
#Generate Race Table
race_percentage = data_final.groupby('race').count()/data_final.shape[0]*100
race_percentage = race_percentage.drop(columns = ['date']).
 ↪rename(columns={'pid':'percentage stopped'})
race_percentage
```

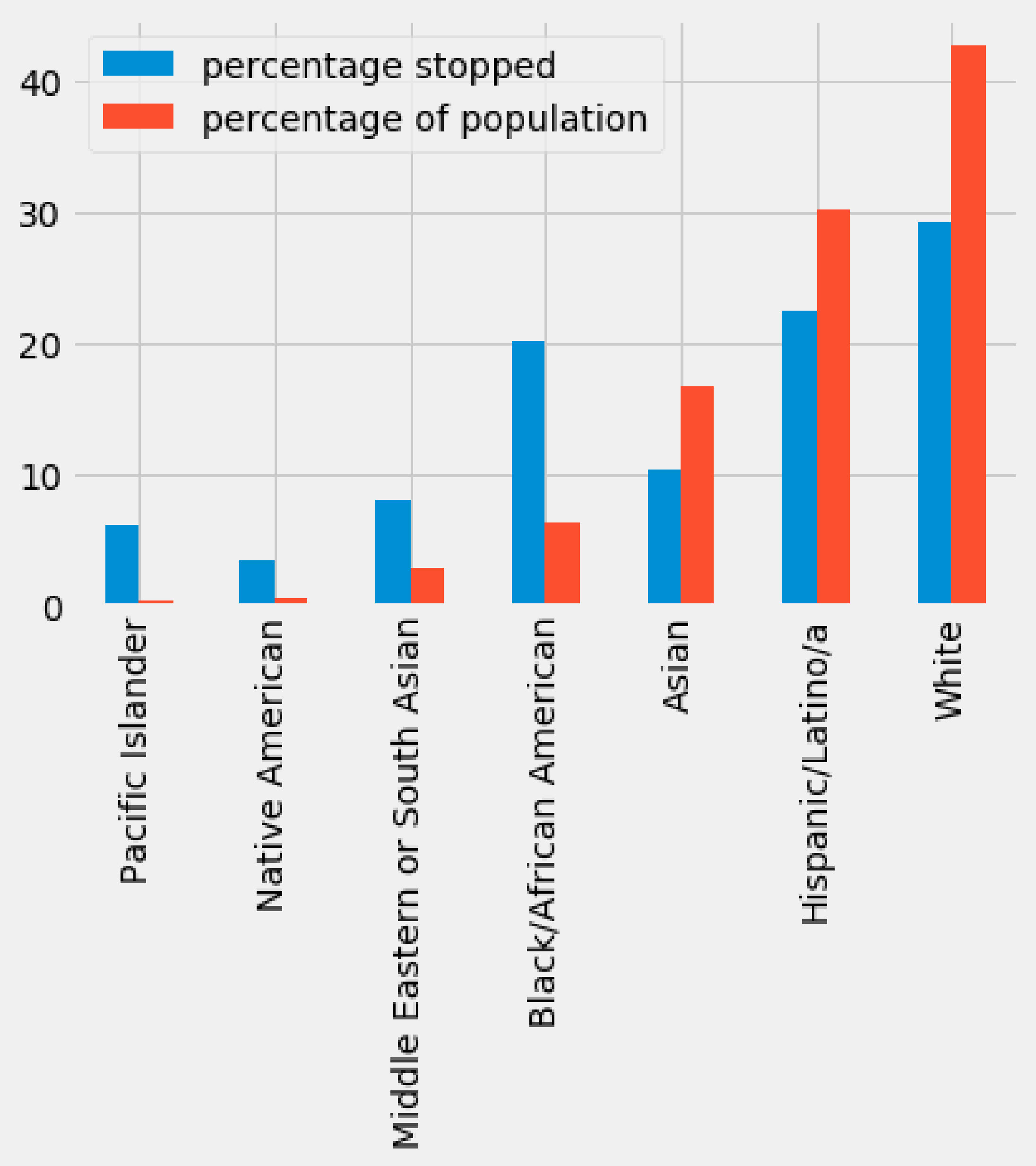
```
[43]:                                 percentage stopped
      race
      Asian                                    10.431055
      Black/African American                   20.159562
      Hispanic/Latino/a                        22.552989
      Middle Eastern or South Asian             8.001905
      Native American                           3.346035
      Pacific Islander                          6.215766
      White                                    29.292689
```

```
[44]: #Now merge the census data and sort by lowest population to highest
      race_census_percentage = race_percentage.merge(data_census_race,left_index =␣
       ↪True,right_index = True)
      race_census_percentage = race_census_percentage.sort_values('percentage of␣
       ↪population', ascending = True)
      race_census_percentage
```

```
[44]:                                 percentage stopped  percentage of population
      Pacific Islander                          6.215766                       0.4
      Native American                           3.346035                       0.5
      Middle Eastern or South Asian             8.001905                       2.9
      Black/African American                   20.159562                       6.4
      Asian                                    10.431055                      16.7
      Hispanic/Latino/a                        22.552989                      30.3
      White                                    29.292689                      42.8
```

```
[45]: #Visualization
      race_census_percentage.plot(kind = 'bar')
```

```
[45]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0e0eb16da0>
```



Lets do some Hypothesis Testing to see if our results are possibly due to chance

Null: There is no significant difference between the percentage of races stopped respective to their demographic Alternate: There is a significant difference between the percentage of race stopped respective to their demographic

[46]:
```
#Test Statistic will be the Mean Difference
test_stat = abs(race_census_percentage.get('percentage stopped')
                - race_census_percentage.get('percentage of population')).mean()
test_stat
```

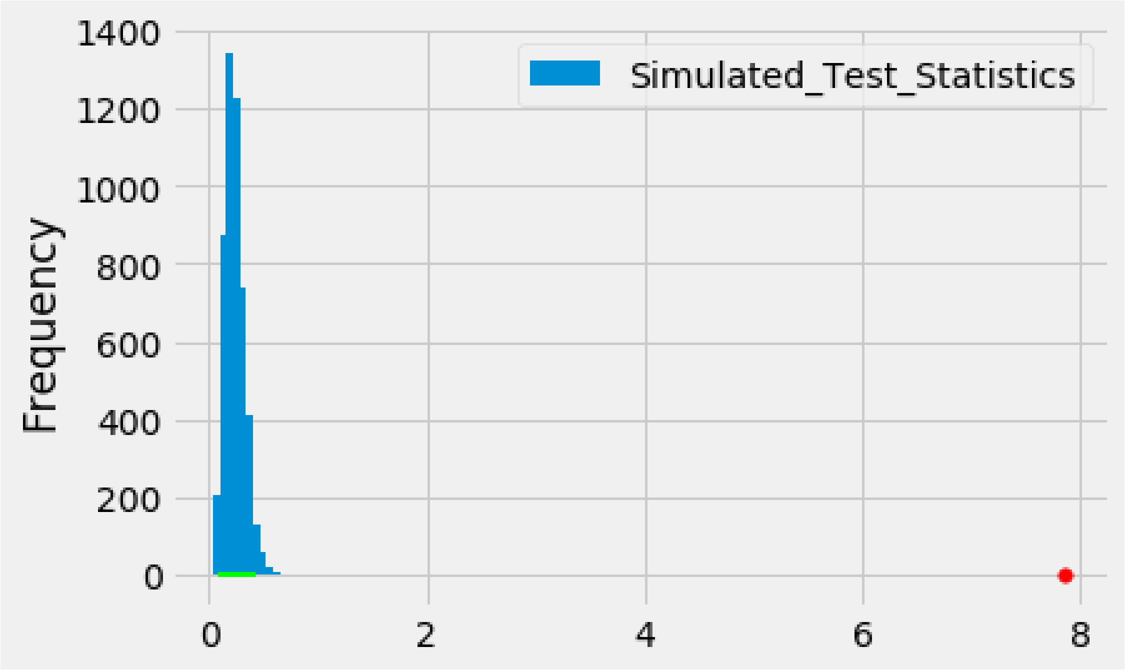
```
[46]: 7.863790698465621
```

```
[47]: #We'll generate about 5000 sample test stats using the census data to create a␣
       ↪95% confidence interval

      num_repetitions = 5000
      population = data_final.shape[0]

      simulated_test_stats = np.array([])

      for i in range(num_repetitions):
          model_proportions = race_census_percentage.get('percentage of population')/
       ↪100
          sample = np.random.multinomial(population, model_proportions)/population
          sim_test_stat = abs(model_proportions-sample).mean()*100
          simulated_test_stats = np.append(simulated_test_stats, sim_test_stat)

      simulated_test_stats
```

```
[47]: array([0.07503827, 0.34724594, 0.19416868, …, 0.14493927, 0.14452421,
             0.21215255])
```

```
[48]: #Lets look at the distribution and generate the 95% confidence interval
      t = pd.DataFrame().assign(Simulated_Test_Statistics = simulated_test_stats)
      t.plot(kind='hist')

      confidence_interval = [np.percentile(simulated_test_stats,2.5),np.
       ↪percentile(simulated_test_stats,97.5)]
      plt.scatter(test_stat, 0, color='red', s=30);
      plt.plot(confidence_interval,[0,0], color = 'lime', linewidth = 2)
      print('Confidence Interval: [' + str(confidence_interval[0]) +', ' +␣
       ↪str(confidence_interval[1]) + ']')
```

```
Confidence Interval: [0.08480488551695932, 0.43254380294627953]
```



[49]:
```
#Now lets generate a p value
p_value = np.count_nonzero(simulated_test_stats >= test_stat)/
 ↪simulated_test_stats.shape[0]
p_value
```

[49]:
```
0.0
```

We reject the null, therefore the difference in the percentage of races being stopped is statistically significant

Geospatial Analysis

Lets see if the frequency a police stops at a location has an effect on the mean difference of races to demographic stopped. This will tell us if there's any bias in our data and how severe.

[ ]:
```
#Data Wrangling
```