sfbus_mapanalysis

October 29, 2020

1 San Francisco Restaurant Inspections

1.1 Description

NOTE: This is taken from a project that I completed for a Data Science course (DS100) at University of California, Berkeley. This is an addendum that I added because I wished to learned more about different ways of visualizing data for analysis.

In this project, I investigated restaurant food safety scores for restaurants in San Francisco. The scores and violation information have been made available by the San Francisco Department of Public Health.

In cleaning and exploring the data, I gained practice with: * Pandas * Data cleaning: identifying type of data collected, missing values, anomalies, etc. * Data Analysis * Exploring different ways of visualizing data: distributions, mapping

1.2 Process and Reflection

I focused a lot of the time on the project learning about new ways to visualize data. In the process, I also learned about different ways of cleaning data in order to get the correct format or remove missing data.

Some new tools that I discovered include: * DATA CLEANING * **geopy + geocoders:** An issue that I noticed while cleaning points to graph was that a lot of businesses were missing longitude, latitutde. * I wanted to looked at ways to use the address in order to find longitude and latitude points, which led me to finding geopy and its Nominatim library. * *ISSUES:* I didn't end up using this idea because of the following reasons: * A problem I ran into here was how inefficient it was in terms of finding all the geocode from the address. Given how much data was missing, it would have taken at least half an hour to process. * In addition, I found that if the addresses were not exact, whether due to commas or different ways of writing numbers (i.e. leading with 0 for single digit numbers), the library would not work. This was an issue because this was the case for many addresses. * *SOLUTION:* Instead, I found that zipcodes were much more readily available for each address. Thus, I decided that mapping scores based on zipcode was a better idea. * VISUALIZATIONS * **shapefile + geopandas:** My goal was to visualize the location of the inspections and the scores to get a look at the possibility of any patterns related to location and inspection scores. * Thus, I discovered the use of shapefiles and geopandas that enable mappings of data points using longitude and latitude. * Another thing I learned is that after using geopandas to import and graph the shapefiles, you can use any other plotting libraries such as seaborn to plot points (as long as you have longitude, latitude points. * **geojson and bokeh:** I wanted to visualize the median scores within each zipcode. * I found geojson to get the polygon shapes for each zipcode and overlay it onto a map. * I also found other libraries from bokeh to help create a color map using the our data points. * I hope to use bokeh more in the future. I think it has a lot of cool features that are great for data visualization.

[31]:

```
%%capture install
! pip install shapely
! pip install geopandas
! pip install descartes
! pip install geopy
! pip install bokeh
```

[ ]:

```

```

1.3 Libraries

[103]:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

import geopandas as gpd
import descartes
from shapely.geometry import Point, Polygon

from geopy.geocoders import Nominatim

import json
from bokeh.io import output_notebook, show, output_file
from bokeh.plotting import figure
from bokeh.models import GeoJSONDataSource, LinearColorMapper, ColorBar
from bokeh.palettes import brewer


sns.set()
plt.style.use('fivethirtyeight')

%matplotlib inline

import zipfile
import os # Used to interact with the file system
from pathlib import Path
```

1.4 Obtaining the Data

[4]:

```
dsDir = Path('data')

bus = pd.read_csv(dsDir/'bus.csv', encoding = 'ISO-8859-1')
ins2vio = pd.read_csv(dsDir/'ins2vio.csv')
ins = pd.read_csv(dsDir/'ins.csv')
vio = pd.read_csv(dsDir/'vio.csv')
```

[5]:

```
display(bus.head())
display(ins.head())
display(vio.head())
```

```
   business id column                         name               address  \
0                1000       HEUNG YUEN RESTAURANT          3279 22nd St
1              100010       ILLY CAFFE SF_PIER 39        PIER 39  K-106-B
2              100017  AMICI'S EAST COAST PIZZERIA           475 06th St
3              100026              LOCAL CATERING       1566 CARROLL AVE
4              100030             OUI OUI! MACARON  2200 JERROLD AVE STE C

            city state postal_code     latitude   longitude  phone_number
0  San Francisco    CA       94110    37.755282 -122.420493         -9999
1  San Francisco    CA       94133 -9999.000000 -9999.000000   14154827284
2  San Francisco    CA       94103 -9999.000000 -9999.000000   14155279839
3  San Francisco    CA       94124 -9999.000000 -9999.000000   14155860315
4  San Francisco    CA       94124 -9999.000000 -9999.000000   14159702675


               iid                    date  score                     type
0  100010_20190329  03/29/2019 12:00:00 AM     -1         New Construction
1  100010_20190403  04/03/2019 12:00:00 AM    100    Routine - Unscheduled
2  100017_20190417  04/17/2019 12:00:00 AM     -1           New Ownership
3  100017_20190816  08/16/2019 12:00:00 AM     91    Routine - Unscheduled
4  100017_20190826  08/26/2019 12:00:00 AM     -1   Reinspection/Followup


                                         description  risk_category     vid
0  Consumer advisory not provided for raw or unde...  Moderate Risk  103128
1                      Contaminated or adulterated food      High Risk  103108
2             Discharge from employee nose mouth or eye  Moderate Risk  103117
3                         Employee eating or smoking  Moderate Risk  103118
4                             Food in poor condition  Moderate Risk  103123
```

[121]:

```
display(ins2vio.head())
```

```
             iid     vid
0  97975_20190725  103124
1  85986_20161011  103114
```

```
2   95754_20190327   103124
3   77005_20170429   103120
4    4794_20181030   103138
```

1.5 I. Data Cleaning: bus and ins

1.5.1 bus: Renaming Business ID

The `bus` dataframe contains a column called `business id column` which probably corresponds to a unique business id. We renamed the column to `bid` to assist with readibility.

[112]:
```python
bus = bus.rename(columns={'business id column': 'bid'})
```

1.5.2 bus: Postal Code

Here we examine the number of restaurants per zipcode.

[106]:
```python
zip_counts = bus.groupby('postal_code').size().sort_values(ascending = False)
print(zip_counts.to_string())
```

```
postal_code
94103          562
94110          555
94102          456
94107          408
94133          398
94109          382
94111          259
94122          255
94105          249
94118          231
94115          230
94108          229
94124          218
94114          200
-9999          194
94112          192
94117          189
94123          177
94121          157
94104          142
94132          132
94116           97
94158           90
```

```
94134         82
94127         67
94131         49
94130          8
94143          5
94013          2
94188          2
CA             2
94301          2
94101          2
95122          1
941033148      1
95133          1
95132          1
94102-5917     1
94014          1
941            1
94080          1
94105-2907     1
92672          1
64110          1
00000          1
94105-1420     1
941102019      1
95117          1
95112          1
95109          1
95105          1
94901          1
94621          1
94602          1
94544          1
94518          1
94117-3504     1
94120          1
94122-1909     1
94123-3106     1
94124-1917     1
94129          1
Ca             1
```

I noticed that there were a lot of missing, invalid and differently formatted zip codes.

So next I want to get a new column that gets the first 5 numbers of zip codes and have None for those with invalid or missing zipcodes.

[107]:

```
#list of valid zipcodes in sf
valid_zips = pd.read_json('data/sf_zipcodes.json',dtype= str)['zip_codes']
```

```
valid_zips.head(5)
```

```
[107]: 0    94102
       1    94103
       2    94104
       3    94105
       4    94107
       Name: zip_codes, dtype: object
```

```
[110]: bus['postal5'] = bus['postal_code'].str[:5]
       invalid_postal5 = bus[~bus['postal5'].isin(valid_zips)]['postal5'].unique()
       bus['postal5'].replace(invalid_postal5, [None] * len(invalid_postal5), inplace = True)
       bus.head()
```

```
[110]:       bid                         name                address           city  \
       0    1000         HEUNG YUEN RESTAURANT            3279 22nd St  San Francisco
       1  100010          ILLY CAFFE SF_PIER 39        PIER 39  K-106-B  San Francisco
       2  100017  AMICI'S EAST COAST PIZZERIA             475 06th St  San Francisco
       3  100026                LOCAL CATERING        1566 CARROLL AVE  San Francisco
       4  100030              OUI OUI! MACARON  2200 JERROLD AVE STE C  San Francisco

         state postal_code     latitude    longitude  phone_number postal5
       0    CA       94110    37.755282  -122.420493         -9999   94110
       1    CA       94133 -9999.000000 -9999.000000   14154827284   94133
       2    CA       94103 -9999.000000 -9999.000000   14155279839   94103
       3    CA       94124 -9999.000000 -9999.000000   14155860315   94124
       4    CA       94124 -9999.000000 -9999.000000   14159702675   94124
```

1.5.3 ins: Extract bid from each Inspection id

```
[111]: ins.head(5)
```

```
[111]:                iid                    date  score                 type  \
       0  100010_20190329  03/29/2019 12:00:00 AM     -1     New Construction
       1  100010_20190403  04/03/2019 12:00:00 AM    100  Routine - Unscheduled
       2  100017_20190417  04/17/2019 12:00:00 AM     -1        New Ownership
       3  100017_20190816  08/16/2019 12:00:00 AM     91  Routine - Unscheduled
       4  100017_20190826  08/26/2019 12:00:00 AM     -1  Reinspection/Followup

             bid  timestamp  year  Missing Score
       0  100010 2019-03-29  2019           True
       1  100010 2019-04-03  2019          False
       2  100017 2019-04-17  2019           True
       3  100017 2019-08-16  2019          False
       4  100017 2019-08-26  2019           True
```

We notice that the column `iid` probably corresponds to an inspection id and has two numbers. The first number likely is the `bid` for the inspection. Next we are creating a new bid column in the ins data frame.

[10]:
```
ins['bid'] = ins['iid'].str.split("_").apply(lambda b: int(b[0]))
ins
```

[10]:
```
                   iid                    date  score                     type  \
0      100010_20190329  03/29/2019 12:00:00 AM     -1         New Construction
1      100010_20190403  04/03/2019 12:00:00 AM    100    Routine - Unscheduled
2      100017_20190417  04/17/2019 12:00:00 AM     -1            New Ownership
3      100017_20190816  08/16/2019 12:00:00 AM     91    Routine - Unscheduled
4      100017_20190826  08/26/2019 12:00:00 AM     -1    Reinspection/Followup
…                …                     …    …                    …
26658     999_20180924  09/24/2018 12:00:00 AM     -1      Routine - Scheduled
26659     999_20181102  11/02/2018 12:00:00 AM     -1    Reinspection/Followup
26660     999_20190909  09/09/2019 12:00:00 AM     80    Routine - Unscheduled
26661      99_20171207  12/07/2017 12:00:00 AM     82    Routine - Unscheduled
26662      99_20180808  08/08/2018 12:00:00 AM     84    Routine - Unscheduled

          bid
0      100010
1      100010
2      100017
3      100017
4      100017
…       …
26658     999
26659     999
26660     999
26661      99
26662      99

[26663 rows x 5 columns]
```

1.5.4 ins: Year Column

We want to get the year for each inspection for data analysis.

[116]:
```
ins_date_type = type(ins['date'][0])
ins['timestamp'] = pd.to_datetime(ins['date'])
ins['year'] = ins['timestamp'].dt.year
ins.head()
```

[116]:
```
               iid                    date  score                     type  \
0  100010_20190329  03/29/2019 12:00:00 AM     -1         New Construction
1  100010_20190403  04/03/2019 12:00:00 AM    100    Routine - Unscheduled
```

```
2  100017_20190417  04/17/2019 12:00:00 AM     -1          New Ownership
3  100017_20190816  08/16/2019 12:00:00 AM     91  Routine - Unscheduled
4  100017_20190826  08/26/2019 12:00:00 AM     -1  Reinspection/Followup

      bid  timestamp  year  Missing Score
0  100010 2019-03-29  2019           True
1  100010 2019-04-03  2019          False
2  100017 2019-04-17  2019           True
3  100017 2019-08-16  2019          False
4  100017 2019-08-26  2019           True
```

1.5.5 Types of Inspections per Year 2016-19

[117]:

```
ins_pivot = ins.pivot_table(index = 'type', columns = 'year', values = 'iid',␣
 ↪aggfunc = 'count', fill_value = 0)
ins_pivot['Total'] = ins_pivot.sum(numeric_only=True, axis = 1)

ins_pivot_sorted = ins_pivot.sort_values('Total', ascending = False)

ins_pivot_sorted
```

[117]:

```
year                               2016  2017  2018  2019  Total
type
Routine - Unscheduled               966  4057  4373  4681  14077
Reinspection/Followup               445  1767  1935  2292   6439
New Ownership                        99   506   528   459   1592
Complaint                            91   418   512   437   1458
New Construction                    102   485   218   189    994
Non-inspection site visit            51   276   253   231    811
New Ownership - Followup              0    45   219   235    499
Structural Inspection                 1   153    50   190    394
Complaint Reinspection/Followup      19    68    70    70    227
Foodborne Illness Investigation       1    29    50    35    115
Routine - Scheduled                   0     9     8    29     46
Administrative or Document Review     2     1     1     0      4
Multi-agency Investigation            0     0     1     2      3
Special Event                         0     3     0     0      3
Community Health Assessment           1     0     0     0      1
```

1.5.6 ins: Missing Scores

[118]:

```
ins['score'].value_counts().head()
```

```
[118]: -1       12632
        100      1993
        96       1681
        92       1260
        94       1250
       Name: score, dtype: int64
```

Something that I noticed was that there are a large number of inspections with the `'score'` of `-1`. This is probably a placeholder for missing scores.

Let's see which types of inspections had missing scores.

```
[119]: ins_tf = ins
       ins_tf['Missing Score'] = ins_tf['score'] == -1
       ins_missing_score_pivot = ins_tf.pivot_table(index = 'type', columns = 'Missing␣
        ↪Score', aggfunc = 'count',values = 'iid', fill_value = 0)
       ins_missing_score_pivot['Total'] = ins_missing_score_pivot.
        ↪sum(numeric_only=True, axis = 1)
       ins_missing_score_pivot = ins_missing_score_pivot.sort_values('Total',␣
        ↪ascending = False)
       ins_missing_score_pivot
```

```
[119]: Missing Score                        False  True  Total
       type
       Routine - Unscheduled                14031    46  14077
       Reinspection/Followup                    0  6439   6439
       New Ownership                            0  1592   1592
       Complaint                                0  1458   1458
       New Construction                         0   994    994
       Non-inspection site visit                0   811    811
       New Ownership - Followup                 0   499    499
       Structural Inspection                    0   394    394
       Complaint Reinspection/Followup          0   227    227
       Foodborne Illness Investigation          0   115    115
       Routine - Scheduled                      0    46     46
       Administrative or Document Review        0     4      4
       Multi-agency Investigation               0     3      3
       Special Event                            0     3      3
       Community Health Assessment              0     1      1
```

I noticed that inspection scores appear only to be assigned to `Routine - Unscheduled` inspections. It is reasonable that for inspection types such as `New Ownership` and `Complaint` to have no associated inspection scores, but we might be curious why there are no inspection scores for the `Reinspection/Followup` inspection type.

1.6 II. Data Analysis: Inspections

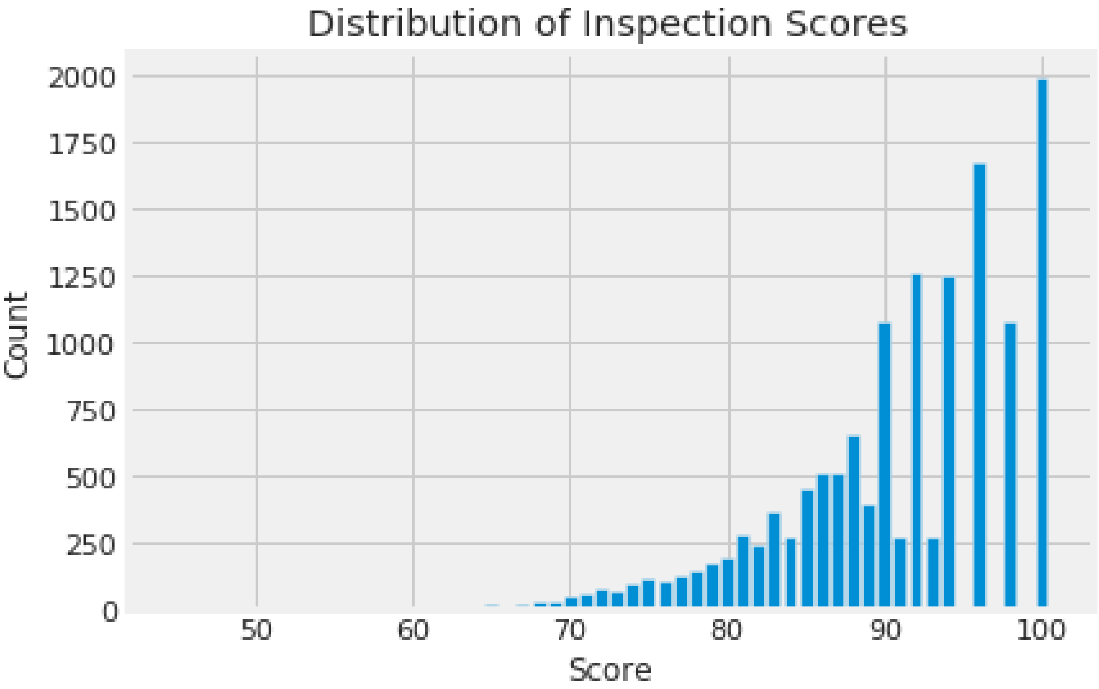
Here, I want to analyze the distribution of inspection scores.

[120]:

```
filtered_ins = ins[ins['Missing Score'] == False]
plt.bar(filtered_ins['score'].value_counts().keys(), filtered_ins['score'].
 ↪value_counts())
plt.xlabel("Score")
plt.ylabel("Count")
plt.title("Distribution of Inspection Scores")
```

[120]:

```
Text(0.5, 1.0, 'Distribution of Inspection Scores')
```



OBSERVATION: The distribution of scores is skewed to the right of the graph, with almost all scores being greater than 60. This will be useful when we are visualizing ’high’ and ’low’ scores relative to each other. The scores that have the highest count are ones in the range of 90-100, which is good because we want scores for restaurants to be higher. We also see that there are also gaps within scores on the top. This can largely be dued to the point deductions from violations being even numbers.

1.7 II. Visualizations

```
[159]: bus.head()
```

```
[159]:       bid                          name               address           city  \
0    1000          HEUNG YUEN RESTAURANT          3279 22nd St  San Francisco
1  100010          ILLY CAFFE SF_PIER 39        PIER 39  K-106-B  San Francisco
2  100017  AMICI'S EAST COAST PIZZERIA            475 06th St  San Francisco
3  100026                 LOCAL CATERING        1566 CARROLL AVE  San Francisco
4  100030               OUI OUI! MACARON  2200 JERROLD AVE STE C  San Francisco

  state postal_code     latitude    longitude phone_number postal5
0    CA       94110    37.755282  -122.420493        -9999   94110
1    CA       94133 -9999.000000 -9999.000000  14154827284   94133
2    CA       94103 -9999.000000 -9999.000000  14155279839   94103
3    CA       94124 -9999.000000 -9999.000000  14155860315   94124
4    CA       94124 -9999.000000 -9999.000000  14159702675   94124
```

1.7.1 Getting relevant columns from business dataframe

We want: * bid * name * longitude * latitude * postal5

```
[133]: bus_clean = bus[["bid","name", "latitude","longitude", "postal5"]]
       bus_clean
```

```
[133]:         bid                          name     latitude    longitude postal5
0       1000          HEUNG YUEN RESTAURANT    37.755282  -122.420493   94110
1     100010          ILLY CAFFE SF_PIER 39 -9999.000000 -9999.000000   94133
2     100017  AMICI'S EAST COAST PIZZERIA -9999.000000 -9999.000000   94103
3     100026                 LOCAL CATERING -9999.000000 -9999.000000   94124
4     100030               OUI OUI! MACARON -9999.000000 -9999.000000   94124
…        …                            …           …            …      …
6248   99948              SUSIECAKES BAKERY -9999.000000 -9999.000000   94118
6249   99988                  HINODEYA SOMA -9999.000000 -9999.000000   94107
6250   99991                        TON TON -9999.000000 -9999.000000   94102
6251   99992    URBAN EXPRESS KITCHENS LLC -9999.000000 -9999.000000   94103
6252   99993              THE BRIXTON SOUTH -9999.000000 -9999.000000   94102

[6253 rows x 5 columns]
```

1.7.2 Joining inspection with business information

```
[134]: ins_named = ins.merge(bus_clean, how = 'left', left_on = 'bid', right_on='bid')
       ins_named
```

```
[134]:
                   iid                    date  score                   type  \
0      100010_20190329  03/29/2019 12:00:00 AM     -1       New Construction
1      100010_20190403  04/03/2019 12:00:00 AM    100  Routine - Unscheduled
2      100017_20190417  04/17/2019 12:00:00 AM     -1          New Ownership
3      100017_20190816  08/16/2019 12:00:00 AM     91  Routine - Unscheduled
4      100017_20190826  08/26/2019 12:00:00 AM     -1  Reinspection/Followup
…                    …                     …    …                      …
26658     999_20180924  09/24/2018 12:00:00 AM     -1    Routine - Scheduled
26659     999_20181102  11/02/2018 12:00:00 AM     -1  Reinspection/Followup
26660     999_20190909  09/09/2019 12:00:00 AM     80  Routine - Unscheduled
26661      99_20171207  12/07/2017 12:00:00 AM     82  Routine - Unscheduled
26662      99_20180808  08/08/2018 12:00:00 AM     84  Routine - Unscheduled

          bid  timestamp  year  Missing Score                         name  \
0      100010 2019-03-29  2019           True        ILLY CAFFE SF_PIER 39
1      100010 2019-04-03  2019          False        ILLY CAFFE SF_PIER 39
2      100017 2019-04-17  2019           True  AMICI'S EAST COAST PIZZERIA
3      100017 2019-08-16  2019          False  AMICI'S EAST COAST PIZZERIA
4      100017 2019-08-26  2019           True  AMICI'S EAST COAST PIZZERIA
…        …         …   …             …                            …
26658     999 2018-09-24  2018           True           SERRANO'S PIZZA II
26659     999 2018-11-02  2018           True           SERRANO'S PIZZA II
26660     999 2019-09-09  2019          False           SERRANO'S PIZZA II
26661      99 2017-12-07  2017          False  J & M A-1 CAFE RESTAURANT LLC
26662      99 2018-08-08  2018          False  J & M A-1 CAFE RESTAURANT LLC

          latitude    longitude postal5
0     -9999.000000 -9999.000000   94133
1     -9999.000000 -9999.000000   94133
2     -9999.000000 -9999.000000   94103
3     -9999.000000 -9999.000000   94103
4     -9999.000000 -9999.000000   94103
…              …           …      …
26658    37.756997  -122.420534   94110
26659    37.756997  -122.420534   94110
26660    37.756997  -122.420534   94110
26661    37.794293  -122.405967   94108
26662    37.794293  -122.405967   94108

[26663 rows x 12 columns]
```

1.7.3 Getting the shapefile

I found the shapefile from datasf.org

```
[146]: street_map = gpd.read_file('geo_export_b35327a2-c448-435e-b713-677e799d2ba5.
        ↪shp')
```

1.8 Visualizing Median Score of Routine - Unscheduled Inspections

1.8.1 Get the median score for each business

```
[147]: ins_median = ins_named[ins_named['Missing Score'] == False].groupby(
           ['bid',
            'longitude',
            'latitude'],
           as_index = False)['score'].median().rename(columns = {'score':'median␣
        ↪score'})
       ins_median
```

```
[147]:          bid    longitude     latitude  median score
       0          19  -122.421547    37.786848          95.0
       1          24  -122.403135    37.792888          98.0
       2          31  -122.419004    37.807155          95.0
       3          45  -122.413641    37.747114          88.0
       4          48  -122.465749    37.764013          90.5
       …         …            …            …             …
       5719   101853 -9999.000000 -9999.000000         100.0
       5720   102067 -9999.000000 -9999.000000         100.0
       5721   102257 -9999.000000 -9999.000000          94.0
       5722   102336 -9999.000000 -9999.000000          82.0
       5723   102398 -9999.000000 -9999.000000          90.0

       [5724 rows x 4 columns]
```

GeoPandas

Here we are going to use GeoPandas to help with mapping points.

```
[148]: geometry = [Point(xy) for xy in zip (ins_median["longitude"],␣
        ↪ins_median['latitude'])]
```

```
[149]: geo_df = gpd.GeoDataFrame(ins_median,
                                 crs = 4326,
                                 geometry = geometry)
       geo_df.head()
```

```
[149]:    bid   longitude   latitude  median score                     geometry
       0   19 -122.421547  37.786848          95.0  POINT (-122.42155 37.78685)
       1   24 -122.403135  37.792888          98.0  POINT (-122.40314 37.79289)
       2   31 -122.419004  37.807155          95.0  POINT (-122.41900 37.80716)
       3   45 -122.413641  37.747114          88.0  POINT (-122.41364 37.74711)
```

```
4   48 -122.465749  37.764013          90.5  POINT (-122.46575 37.76401)
```

Geocode Bounds for SF

[150]:

```
sf_ulat = 37.84
sf_llat = 37.70
sf_ulon = -122.34
sf_llon = -122.52
```

[151]:

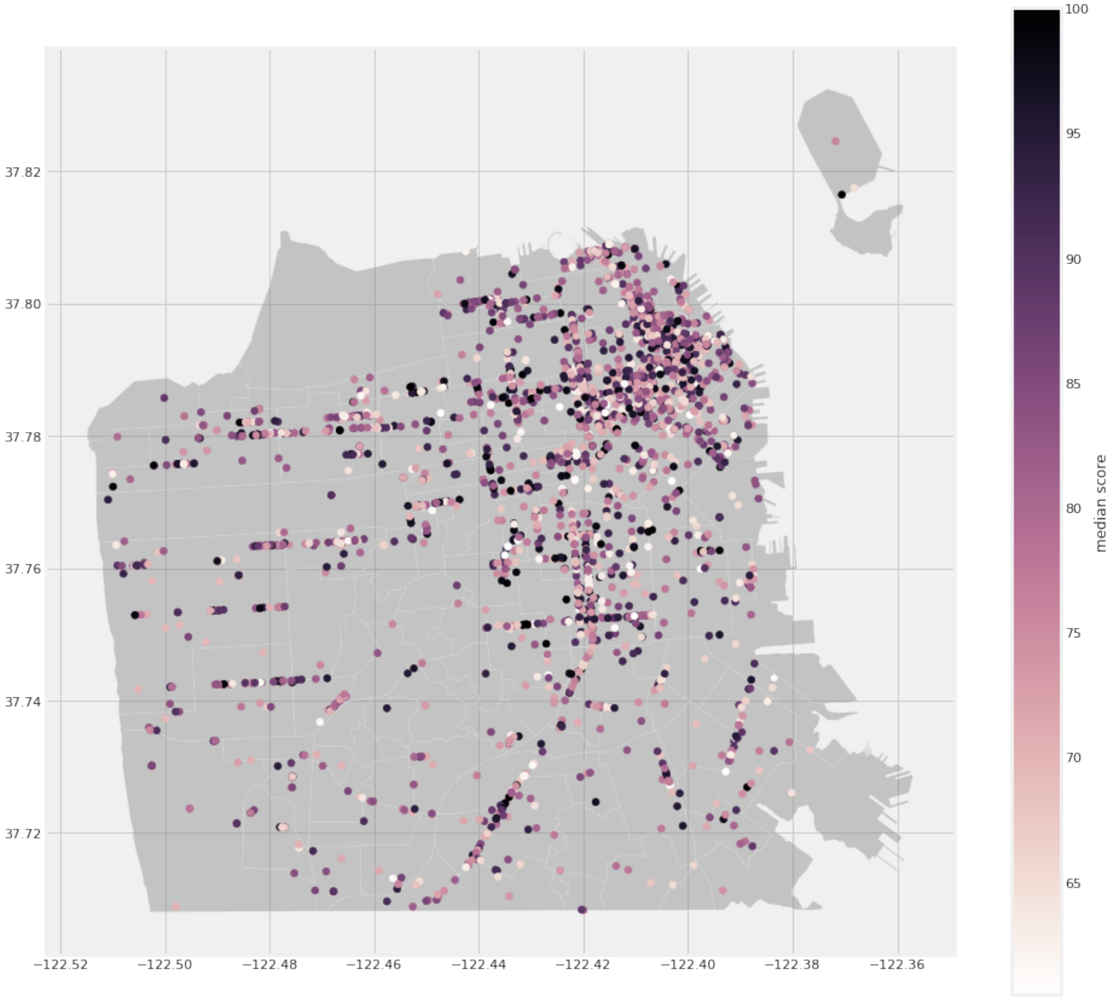
```
clean_geo = geo_df.loc[(geo_df['latitude'].between(sf_llat, sf_ulat))
                      & (geo_df['longitude'].between(sf_llon, sf_ulon))]
print(len(clean_geo))
```

```
2719
```

1.8.2 Map of Median Inspection Scores in SF

[152]:

```
fig, axs = plt.subplots(figsize = (15,15))
streets_fig = street_map.plot(ax = axs, alpha = 0.4, color = 'grey')

cmap = sns.cubehelix_palette(light = 1, dark = 0,as_cmap=True)

point_fig = clean_geo.plot(ax = axs,
                           c = 'median score',
                           cmap = cmap,)

norm = plt.Normalize(clean_geo['median score'].min(), clean_geo['median score'].
 ↪max())

sm = plt.cm.ScalarMappable(cmap = cmap, norm=norm)
sm.set_array([])
# Remove the legend and add a colorbar
axs.figure.colorbar(sm).set_label('median score')
```



1.9 Inspection Type Visualization and Analysis

Let's look at the different types of inspections.

[161]:
```
ins_types = pd.Series(ins_pivot.index)
ins_types
```

[161]:
```
0     Administrative or Document Review
1           Community Health Assessment
2                             Complaint
3       Complaint Reinspection/Followup
4       Foodborne Illness Investigation
5            Multi-agency Investigation
6                      New Construction
7                         New Ownership
```

```
8             New Ownership - Followup
9            Non-inspection site visit
10               Reinspection/Followup
11                 Routine - Scheduled
12               Routine - Unscheduled
13                       Special Event
14               Structural Inspection
Name: type, dtype: object
```

[174]:

```
clean_type = ins_named.loc[(ins_named['latitude'].between(sf_llat, sf_ulat))
                     & (ins_named['longitude'].between(sf_llon, sf_ulon)),
                          ['bid', 'name', 'longitude',␣
 ↪'latitude','type','year',]]
clean_type
```

[174]:

```
        bid                           name   longitude   latitude  \
59     1000          HEUNG YUEN RESTAURANT -122.420493  37.755282
60     1000          HEUNG YUEN RESTAURANT -122.420493  37.755282
61     1000          HEUNG YUEN RESTAURANT -122.420493  37.755282
62     1000          HEUNG YUEN RESTAURANT -122.420493  37.755282
63     1000          HEUNG YUEN RESTAURANT -122.420493  37.755282
…     …                            …         …        …
26658   999             SERRANO'S PIZZA II -122.420534  37.756997
26659   999             SERRANO'S PIZZA II -122.420534  37.756997
26660   999             SERRANO'S PIZZA II -122.420534  37.756997
26661    99  J & M A-1 CAFE RESTAURANT LLC -122.405967  37.794293
26662    99  J & M A-1 CAFE RESTAURANT LLC -122.405967  37.794293

                        type  year
59     Reinspection/Followup  2016
60     Routine - Unscheduled  2017
61     Reinspection/Followup  2017
62     Routine - Unscheduled  2018
63     Reinspection/Followup  2018
…                        …    …
26658    Routine - Scheduled  2018
26659  Reinspection/Followup  2018
26660  Routine - Unscheduled  2019
26661  Routine - Unscheduled  2017
26662  Routine - Unscheduled  2018

[12571 rows x 6 columns]
```

1.9.1 New Constructions Per Year

Let's look at the number of new construction inspections per year.

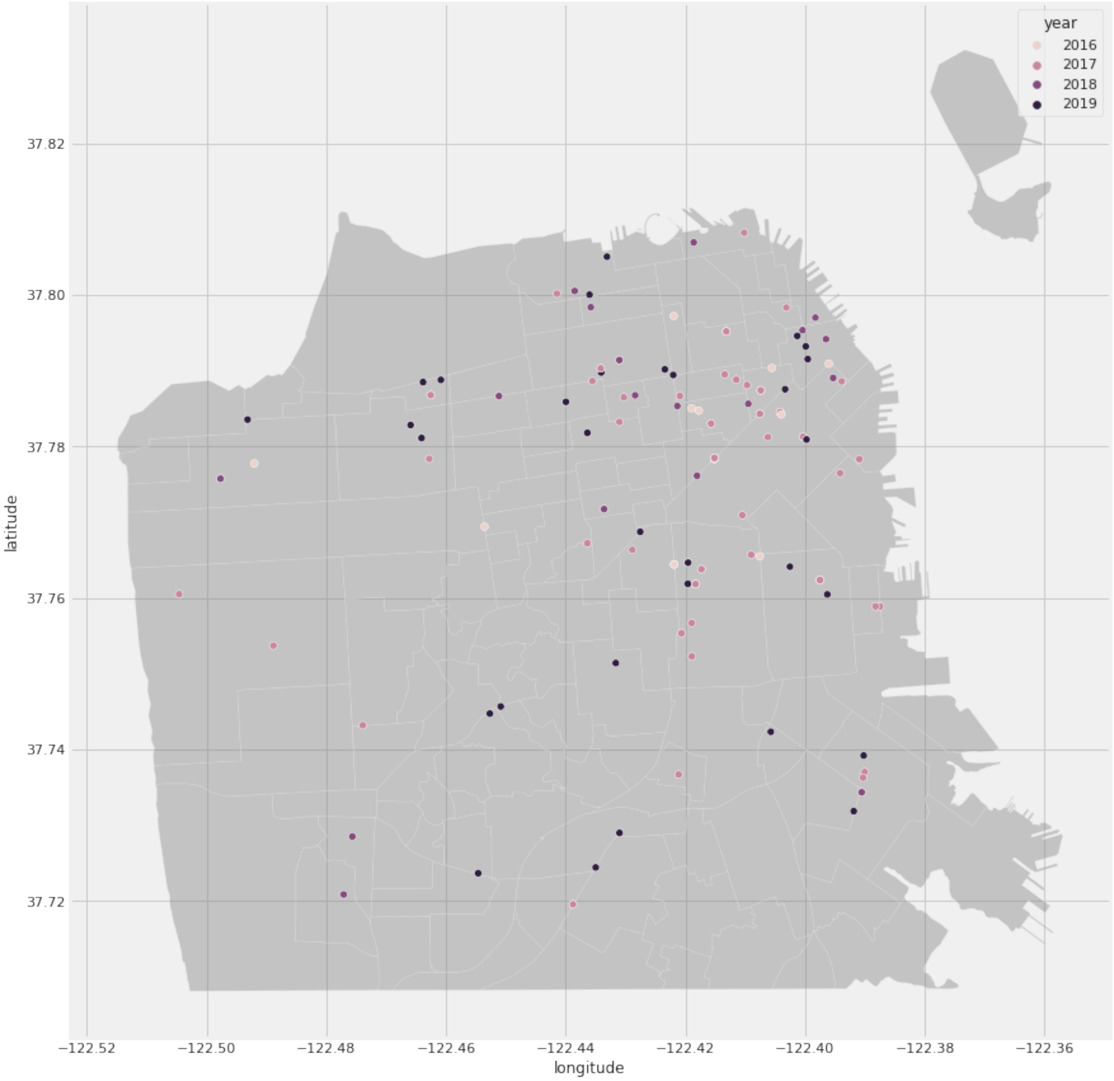
```
[181]: ins_newcon = ins_named[ins_named['type'] == 'New Construction'].loc[:,['bid',
            'name',
            'longitude',
            'latitude',
            'year']]
       ins_newcon
```

```
[181]:          bid                                  name  longitude  latitude  year
       0      100010                  ILLY CAFFE SF_PIER 39    -9999.0   -9999.0  2019
       25     100059                         DUMPLING ALLEY    -9999.0   -9999.0  2019
       26     100059                         DUMPLING ALLEY    -9999.0   -9999.0  2019
       27     100059                         DUMPLING ALLEY    -9999.0   -9999.0  2019
       39     100081  THE MATTERHORN RESTAURANT AND BAKERY    -9999.0   -9999.0  2019
       …        …                                     …        …         …       …
       26638   99948                      SUSIECAKES BAKERY    -9999.0   -9999.0  2019
       26639   99948                      SUSIECAKES BAKERY    -9999.0   -9999.0  2019
       26640   99948                      SUSIECAKES BAKERY    -9999.0   -9999.0  2019
       26649   99993                       THE BRIXTON SOUTH    -9999.0   -9999.0  2019
       26650   99993                       THE BRIXTON SOUTH    -9999.0   -9999.0  2019

       [994 rows x 5 columns]
```

1.9.2 Visual: New Constructions Locations by Year

I decided to use seaborn to plot points here to explore other ways to use the shapefile.

```
[180]: fig, axs = plt.subplots(figsize = (15,15))
       streets_fig = street_map.plot(ax = axs, alpha = 0.4, color = 'grey')

       newcon_fig = sns.scatterplot(ax = axs,
                                    data = clean_type[clean_type['type'] == 'New␣
        ↪Construction'],
                                    x = 'longitude',
                                    y = 'latitude',
                                    hue = 'year',)
```

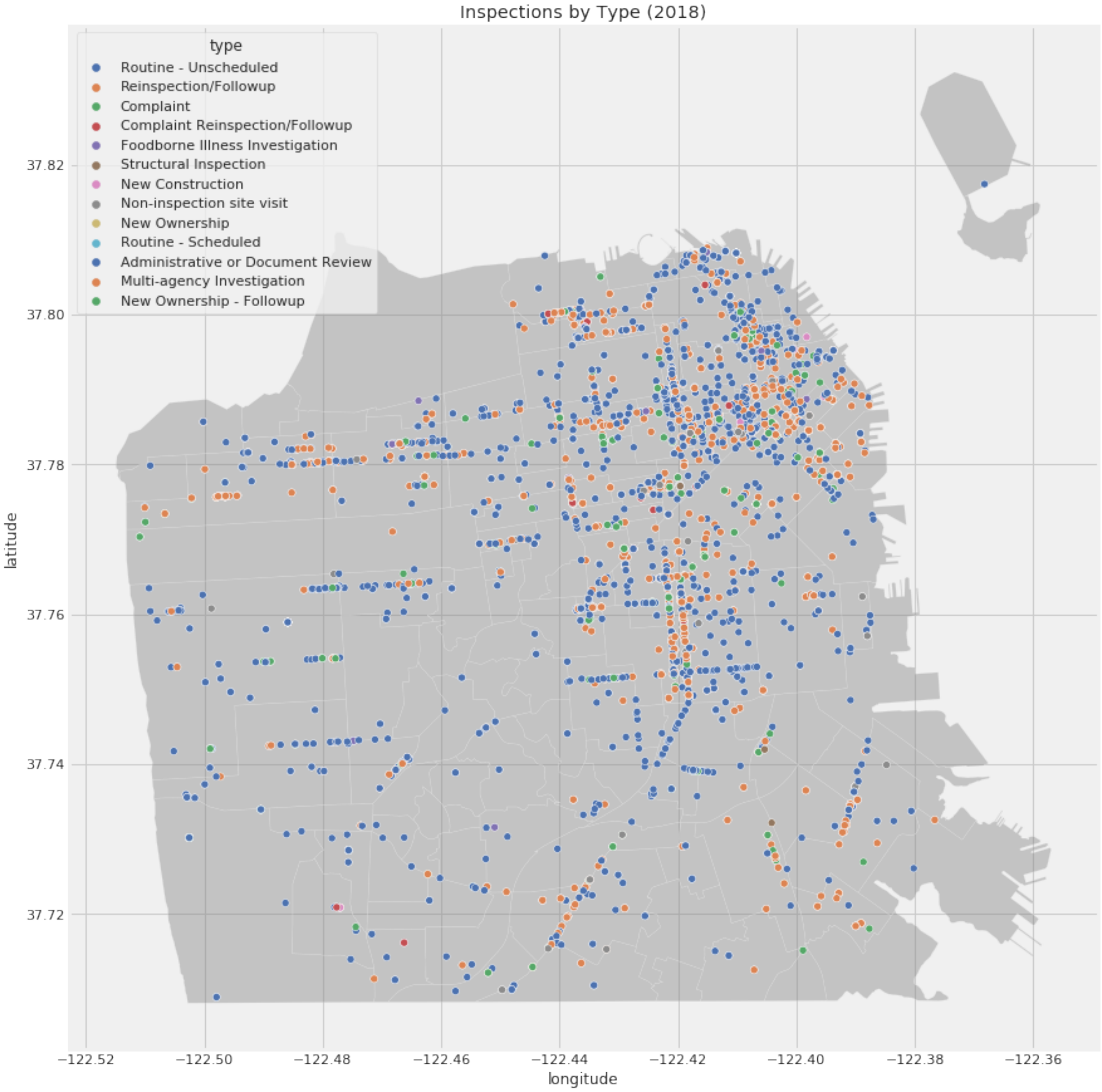


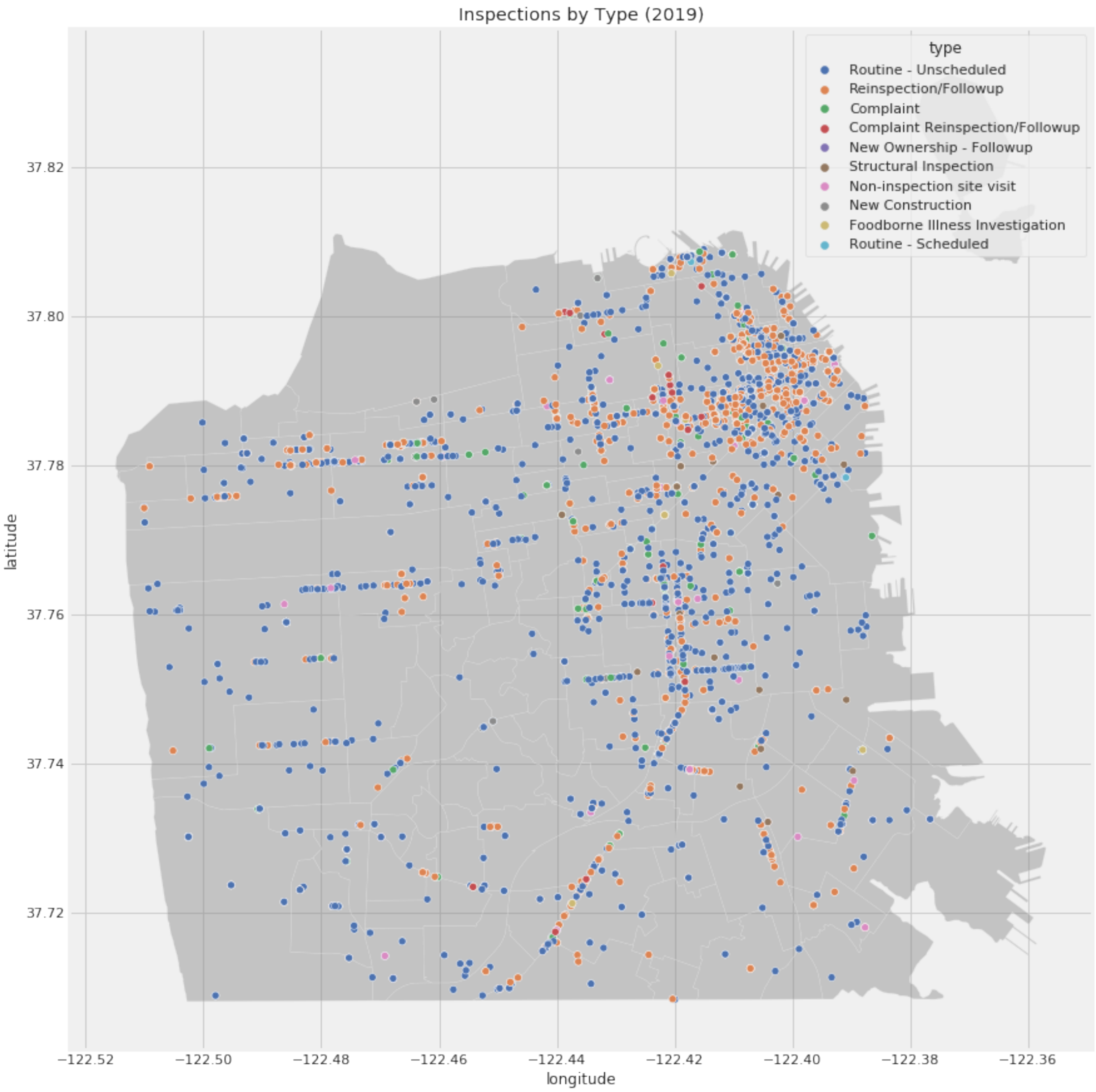
OBSERVATION: A lot of the inspections that were New Construction did not have longitude and latitude points. Thus, we are missing a lot of data on where a lot of New Construction are located.

1.9.3 Inspections types location (2018 and 2019)

[202]:
```
fig, axs = plt.subplots(figsize = (15,15))
streets_fig = street_map.plot(ax = axs, alpha = 0.4, color = 'grey')

newcon_fig = sns.scatterplot(ax = axs,
                             data = clean_type[clean_type['year'] == 2018],
                             x = 'longitude',
                             y = 'latitude',
                             hue = 'type',
                             palette = 'deep',)
plt.title("Inspections by Type (2018)")

fig, axs = plt.subplots(figsize = (15,15))
streets_fig = street_map.plot(ax = axs, alpha = 0.4, color = 'grey')

newcon_fig = sns.scatterplot(ax = axs,
                             data = clean_type[clean_type['year'] == 2019],
                             x = 'longitude',
                             y = 'latitude',
                             hue = 'type',
                             palette = 'deep',)
plt.title("Inspections by Type (2019)")
```

```
[202]: Text(0.5, 1, 'Inspections by Type (2019)')
```


Inspections by Type (2018)
type
Routine - Unscheduled
Reinspection/Followup
Complaint
Complaint Reinspection/Followup
Foodborne Illness Investigation
Structural Inspection
New Construction
Non-inspection site visit
New Ownership
Routine - Scheduled
Administrative or Document Review
Multi-agency Investigation
New Ownership - Followup
37.82
37.80
37.78
37.76
37.74
37.72
latitude
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
longitude



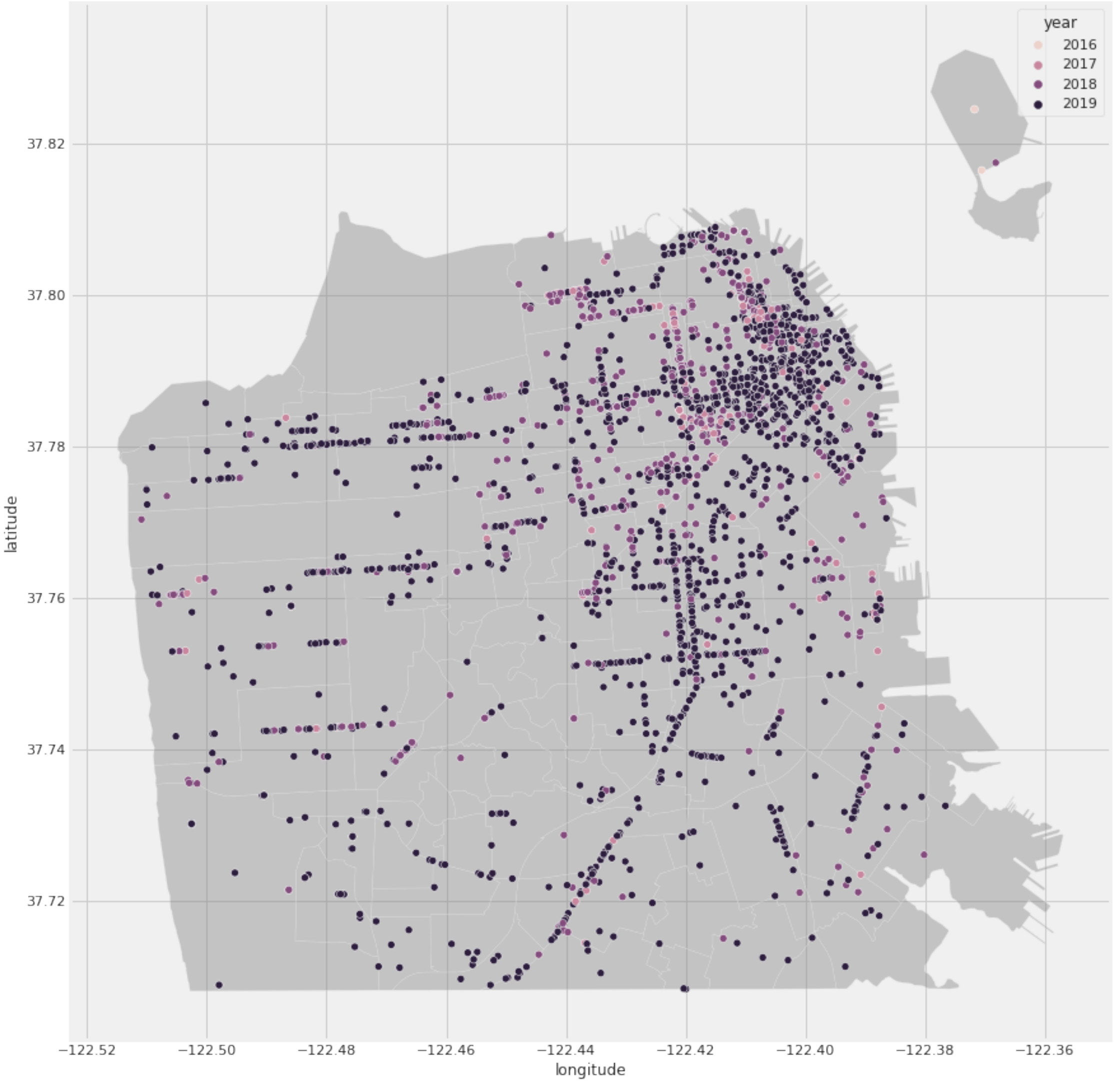
1.10 Inspections Locations per Year

I looked next at where inspections took place over the years. For each business, I chose to look at the their latest inspection.

```
[192]: recent_ins = clean_type.sort_values("year", ascending = False).
        ↪groupby(['bid','longitude', 'latitude']).first().reset_index()
```

```
[193]: fig, axs = plt.subplots(figsize = (15,15))
       streets_fig = street_map.plot(ax = axs, alpha = 0.4, color = 'grey')
```

```
newcon_fig = sns.scatterplot(ax = axs,
                             data = recent_ins,
                             x = 'longitude',
                             y = 'latitude',
                             hue = 'year',)
```



```
[ ]:
```

1.11 III. Visualization: Median Score per Zip Code (2019)

```
[207]: valid_zips = pd.read_json('data/sf_zipcodes.json',dtype= str)['zip_codes']

       valid_postal5 = bus[bus['postal5'].isin(valid_zips)]
       valid_postal5
```

```
[207]:        bid                        name                 address  \
       0      1000          HEUNG YUEN RESTAURANT            3279 22nd St
       1    100010          ILLY CAFFE SF_PIER 39         PIER 39  K-106-B
       2    100017  AMICI'S EAST COAST PIZZERIA             475 06th St
       3    100026                 LOCAL CATERING        1566 CARROLL AVE
       4    100030               OUI OUI! MACARON  2200 JERROLD AVE STE C
       …       …                          …                       …
       6248  99948              SUSIECAKES BAKERY       3509 CALIFORNIA ST
       6249  99988                  HINODEYA SOMA      303 02nd ST STE 102
       6250  99991                        TON TON            422 GEARY ST
       6251  99992    URBAN EXPRESS KITCHENS LLC             475 06th ST
       6252  99993             THE BRIXTON SOUTH             701 02nd St

                      city state postal_code     latitude    longitude phone_number  \
       0     San Francisco    CA       94110    37.755282  -122.420493        -9999
       1     San Francisco    CA       94133 -9999.000000 -9999.000000  14154827284
       2     San Francisco    CA       94103 -9999.000000 -9999.000000  14155279839
       3     San Francisco    CA       94124 -9999.000000 -9999.000000  14155860315
       4     San Francisco    CA       94124 -9999.000000 -9999.000000  14159702675
       …               …   …         …          …            …           …
       6248  San Francisco    CA       94118 -9999.000000 -9999.000000  14150452253
       6249  San Francisco    CA       94107 -9999.000000 -9999.000000        -9999
       6250  San Francisco    CA       94102 -9999.000000 -9999.000000  14155531280
       6251  San Francisco    CA       94103 -9999.000000 -9999.000000  14150368085
       6252  San Francisco    CA       94102 -9999.000000 -9999.000000  14158315871

            postal5
       0      94110
       1      94133
       2      94103
       3      94124
       4      94124
       …        …
       6248   94118
       6249   94107
       6250   94102
       6251   94103
       6252   94102

       [6032 rows x 10 columns]
```

```
[208]: bus_clean_zip = bus[["bid", "postal5"]]
       bus_clean_zip
```

```
[208]:          bid postal5
       0       1000   94110
       1     100010   94133
       2     100017   94103
       3     100026   94124
       4     100030   94124
       …        …      …
       6248   99948   94118
       6249   99988   94107
       6250   99991   94102
       6251   99992   94103
       6252   99993   94102

       [6253 rows x 2 columns]
```

```
[215]: ins_zip_score = ins_named[(ins_named['Missing Score'] == False) &␣
        ↪(ins_named['year'] == 2019)].groupby(
           'postal5',
           as_index = False)['score'].median().rename(columns = {'score':
        ↪'median_score'})
       ins_zip_score.head()
```

```
[215]:   postal5  median_score
       0   94102          92.0
       1   94103          91.0
       2   94104          90.0
       3   94105          90.0
       4   94107          96.0
```

```
[216]: data = 'SanFrancisco.Neighborhoods.json'
       gdf = gpd.read_file(data)
       gdf.head()
```

```
[216]:       id neighborhood                                          geometry
       0  94105  Rincon Hill  GEOMETRYCOLLECTION (POLYGON ((-122.39170 37.79…
       1  94107  South Beach  GEOMETRYCOLLECTION (POLYGON ((-122.38777 37.78…
       2  94108    Chinatown  GEOMETRYCOLLECTION (POLYGON ((-122.40496 37.79…
       3  94109     Nob Hill  GEOMETRYCOLLECTION (POLYGON ((-122.42043 37.80…
       4  94112    Ingleside  GEOMETRYCOLLECTION (POLYGON ((-122.42070 37.73…
```

```
[217]: ins_zip_score['id'] = ins_zip_score['postal5'].astype(str)
```

```
[218]: merge = gdf.merge(ins_zip_score, how='left', on='id')
```

```python
merged_json = json.loads(merge.to_json())
json_data = json.dumps(merged_json)
json_data
```

```
[218]: '{"type": "FeatureCollection", "features": [{"id": "0", "type": "Feature",
"properties": {"id": "94105", "median_score": 90.0, "neighborhood": "Rincon
Hill", "postal5": "94105"}, "geometry": {"type": "GeometryCollection",
"geometries": [{"type": "Polygon", "coordinates": [[[-122.391701, 37.794113],
[-122.39198, 37.793906], [-122.391614, 37.793571], [-122.391714, 37.793459],
[-122.388816, 37.791005], [-122.388932, 37.790919], [-122.388616, 37.790348],
[-122.388076, 37.790518], [-122.388375, 37.790334], [-122.388225, 37.790032],
[-122.385852, 37.790951], [-122.385496, 37.790559], [-122.387589, 37.789838],
[-122.387289, 37.789347], [-122.385303, 37.789838], [-122.38512, 37.789313],
[-122.387174, 37.788807], [-122.387032, 37.788255], [-122.385261, 37.788537],
[-122.385136, 37.788156], [-122.38739, 37.787736], [-122.387415, 37.787269],
[-122.3845, 37.787437], [-122.384342, 37.785728], [-122.387577, 37.785485],
[-122.38765, 37.784929], [-122.385546, 37.785009], [-122.385455, 37.784711],
[-122.387477, 37.784493], [-122.387691, 37.784412], [-122.387722, 37.783928],
[-122.385271, 37.784049], [-122.38525, 37.78379], [-122.38572, 37.783774],
[-122.38573, 37.78354], [-122.386088, 37.783435], [-122.387732, 37.783282],
[-122.387773, 37.782911], [-122.388426, 37.781801], [-122.388189, 37.784771],
[-122.388504, 37.785348], [-122.389694, 37.786243], [-122.39141, 37.785103],
[-122.39146, 37.7855], [-122.393512, 37.784564], [-122.394567, 37.783783],
[-122.396711, 37.785549], [-122.398941, 37.783785], [-122.400955, 37.785389],
[-122.399523, 37.786631], [-122.39999, 37.787004], [-122.401497, 37.785824],
[-122.403374, 37.787401], [-122.403427, 37.787676], [-122.393875, 37.795248],
[-122.392429, 37.793835], [-122.391701, 37.794113]]]}]}}, {"id": "1", "type":
"Feature", "properties": {"id": "94107", "median_score": 96.0, "neighborhood":
"South Beach", "postal5": "94107"}, "geometry": {"type": "GeometryCollection",
"geometries": [{"type": "Polygon", "coordinates": [[[-122.387773, 37.782911],
[-122.38474, 37.782975], [-122.384693, 37.78256], [-122.387778, 37.782318],
[-122.387757, 37.781825], [-122.385561, 37.781946], [-122.385521, 37.781607],
[-122.387737, 37.781454], [-122.38743, 37.778402], [-122.385306, 37.778443],
[-122.387471, 37.778322], [-122.389642, 37.777478], [-122.390383, 37.777037],
[-122.391116, 37.777571], [-122.393279, 37.775838], [-122.393858, 37.776284],
[-122.396169, 37.774505], [-122.39557, 37.774012], [-122.40004, 37.7705],
[-122.395085, 37.766537], [-122.394825, 37.766551], [-122.394286, 37.766114],
[-122.393456, 37.765089], [-122.393013, 37.764089], [-122.387845, 37.764369],
[-122.386873, 37.765646], [-122.386312, 37.765628], [-122.386523, 37.765339],
[-122.386427, 37.764449], [-122.386725, 37.76414], [-122.386629, 37.76392],
[-122.387081, 37.76357], [-122.386533, 37.763395], [-122.386581, 37.763239],
[-122.386331, 37.763148], [-122.386187, 37.763369], [-122.385513, 37.763308],
[-122.385484, 37.763163], [-122.385186, 37.763369], [-122.385099, 37.762353],
[-122.384723, 37.76214], [-122.38429, 37.762162], [-122.384473, 37.764079],
[-122.384319, 37.764072], [-122.384098, 37.762261], [-122.38326, 37.762322],
[-122.383289, 37.762543], [-122.383771, 37.762573], [-122.383809, 37.764277],
[-122.383289, 37.764148], [-122.383116, 37.762467], [-122.382337, 37.762482],
```

[-122.382519, 37.764647], [-122.382336, 37.764639], [-122.382125, 37.762425],
[-122.381894, 37.762425], [-122.381865, 37.762745], [-122.381682, 37.762737],
[-122.381643, 37.764852], [-122.381153, 37.764951], [-122.380883, 37.764776],
[-122.380874, 37.763308], [-122.380691, 37.763293], [-122.380383, 37.760219],
[-122.379507, 37.76028], [-122.379776, 37.763506], [-122.379526, 37.763513],
[-122.379218, 37.759968], [-122.381288, 37.759787], [-122.381327, 37.759353],
[-122.381202, 37.759322], [-122.381317, 37.75901], [-122.381182, 37.758995],
[-122.381288, 37.75866], [-122.381134, 37.758638], [-122.38125, 37.758318],
[-122.381096, 37.758211], [-122.38124, 37.758211], [-122.3815, 37.75777],
[-122.381481, 37.757549], [-122.381288, 37.755936], [-122.380787, 37.755487],
[-122.380797, 37.755258], [-122.384109, 37.75497], [-122.384109, 37.754719],
[-122.38333, 37.754689], [-122.382868, 37.754384], [-122.382896, 37.753699],
[-122.382723, 37.753418], [-122.381703, 37.753197], [-122.381433, 37.752999],
[-122.380095, 37.752923], [-122.380089, 37.752759], [-122.381482, 37.752782],
[-122.382078, 37.753131], [-122.382933, 37.753184], [-122.387833, 37.752817],
[-122.387666, 37.750267], [-122.391705, 37.750133], [-122.401546, 37.74945],
[-122.403798, 37.749443], [-122.403022, 37.752336], [-122.403364, 37.752366],
[-122.403524, 37.754463], [-122.40315, 37.754488], [-122.403328, 37.756082],
[-122.403585, 37.756818], [-122.403428, 37.756871], [-122.40405, 37.757619],
[-122.405604, 37.758851], [-122.406015, 37.759409], [-122.406167, 37.759315],
[-122.40645, 37.760114], [-122.406451, 37.760865], [-122.40604, 37.761865],
[-122.405859, 37.761815], [-122.405832, 37.762199], [-122.405271, 37.763272],
[-122.405115, 37.764635], [-122.399688, 37.765005], [-122.3998, 37.766807],
[-122.400422, 37.767291], [-122.40001, 37.767614], [-122.400915, 37.768096],
[-122.400881, 37.767518], [-122.401851, 37.767416], [-122.402021, 37.768998],
[-122.401768, 37.769145], [-122.401685, 37.769721], [-122.402091, 37.769716],
[-122.402105, 37.769912], [-122.403477, 37.769842], [-122.404003, 37.77018],
[-122.403723, 37.770182], [-122.399425, 37.773612], [-122.405629, 37.778502],
[-122.396711, 37.785549], [-122.394567, 37.783783], [-122.393512, 37.784564],
[-122.39146, 37.7855], [-122.39141, 37.785103], [-122.389694, 37.786243],
[-122.388504, 37.785348], [-122.388189, 37.784771], [-122.388426, 37.781801],
[-122.387773, 37.782911]]]}]}}, {"id": "2", "type": "Feature", "properties":
{"id": "94108", "median_score": 91.0, "neighborhood": "Chinatown", "postal5":
"94108"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.404959, 37.795337], [-122.404055, 37.790751],
[-122.402694, 37.790937], [-122.402535, 37.790006], [-122.403858, 37.789819],
[-122.403478, 37.787965], [-122.403247, 37.787814], [-122.405883, 37.785718],
[-122.406236, 37.785793], [-122.406602, 37.787583], [-122.408244, 37.787369],
[-122.408655, 37.789244], [-122.411913, 37.788812], [-122.41229, 37.790669],
[-122.413931, 37.79047], [-122.414883, 37.79503], [-122.411553, 37.795409],
[-122.411484, 37.795033], [-122.411024, 37.795145], [-122.411087, 37.795467],
[-122.407091, 37.79597], [-122.407014, 37.795587], [-122.406591, 37.795697],
[-122.406649, 37.796027], [-122.40568, 37.796152], [-122.405526, 37.795292],
[-122.404959, 37.795337]]]}]}}, {"id": "3", "type": "Feature", "properties":
{"id": "94109", "median_score": 91.0, "neighborhood": "Nob Hill", "postal5":
"94109"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.420432, 37.808308], [-122.420342, 37.807746],

```
[-122.420746, 37.807598], [-122.419084, 37.807804], [-122.418879, 37.806877],
[-122.4196, 37.806783], [-122.418789, 37.806249], [-122.418349, 37.804101],
[-122.418468, 37.804109], [-122.418129, 37.803168], [-122.417893, 37.80172],
[-122.419167, 37.801559], [-122.419075, 37.801144], [-122.417814, 37.80132],
[-122.417548, 37.799967], [-122.418252, 37.799884], [-122.418226, 37.799295],
[-122.41741, 37.799405], [-122.417224, 37.798494], [-122.415574, 37.798688],
[-122.413931, 37.79047], [-122.41229, 37.790669], [-122.411712, 37.78787],
[-122.413356, 37.787657], [-122.413165, 37.786726], [-122.414805, 37.786523],
[-122.414256, 37.783728], [-122.427427, 37.782055], [-122.430001, 37.794906],
[-122.423491, 37.795723], [-122.42553, 37.805821], [-122.425374, 37.806107],
[-122.425397, 37.806565], [-122.426352, 37.807528], [-122.425528, 37.806976],
[-122.424292, 37.806622], [-122.423469, 37.80678], [-122.422145, 37.80759],
[-122.421381, 37.808188], [-122.42109, 37.808549], [-122.421123, 37.808812],
[-122.422852, 37.810176], [-122.422752, 37.810251], [-122.421319, 37.809178],
[-122.421194, 37.809232], [-122.420906, 37.808818], [-122.420739, 37.808837],
[-122.420764, 37.809105], [-122.420323, 37.809002], [-122.420639, 37.80884],
[-122.42032, 37.808739], [-122.420514, 37.808645], [-122.420432,
37.808308]]]}]}}, {"id": "4", "type": "Feature", "properties": {"id": "94112",
"median_score": 90.0, "neighborhood": "Ingleside", "postal5": "94112"},
"geometry": {"type": "GeometryCollection", "geometries": [{"type": "Polygon",
"coordinates": [[[-122.420701, 37.731936], [-122.420551, 37.731562],
[-122.420949, 37.731304], [-122.420182, 37.731449], [-122.420306, 37.729926],
[-122.420399, 37.729529], [-122.420599, 37.729374], [-122.422911, 37.729289],
[-122.423689, 37.729515], [-122.426026, 37.729391], [-122.426016, 37.728636],
[-122.423712, 37.728761], [-122.423154, 37.727262], [-122.424157, 37.725984],
[-122.423554, 37.725625], [-122.422873, 37.723983], [-122.424475, 37.723544],
[-122.425246, 37.722277], [-122.424405, 37.721949], [-122.425484, 37.720235],
[-122.423682, 37.719801], [-122.423493, 37.719533], [-122.423541, 37.718906],
[-122.423308, 37.718471], [-122.422477, 37.717882], [-122.423615, 37.71719],
[-122.424422, 37.717063], [-122.425093, 37.717356], [-122.425602, 37.718097],
[-122.42605, 37.718375], [-122.427306, 37.715611], [-122.42836, 37.716053],
[-122.431213, 37.716631], [-122.433751, 37.713118], [-122.42923, 37.712203],
[-122.426175, 37.710959], [-122.427347, 37.710021], [-122.427285, 37.708352],
[-122.433961, 37.708232], [-122.435558, 37.70799], [-122.436272, 37.708244],
[-122.440646, 37.708221], [-122.441103, 37.708333], [-122.442024, 37.708085],
[-122.44211, 37.708251], [-122.44981, 37.708246], [-122.449854, 37.708358],
[-122.450204, 37.708242], [-122.454386, 37.708221], [-122.455966, 37.708386],
[-122.45716, 37.708221], [-122.458838, 37.708437], [-122.459901, 37.708244],
[-122.469004, 37.708187], [-122.46745, 37.709496], [-122.465786, 37.710243],
[-122.464028, 37.710549], [-122.460894, 37.710579], [-122.462503, 37.711207],
[-122.462598, 37.711377], [-122.462562, 37.713806], [-122.462758, 37.714264],
[-122.462626, 37.714469], [-122.46268, 37.720018], [-122.462264, 37.720021],
[-122.46227, 37.72527], [-122.462838, 37.725488], [-122.462587, 37.727158],
[-122.462888, 37.728087], [-122.462066, 37.728095], [-122.460858, 37.728754],
[-122.460039, 37.728777], [-122.459911, 37.729952], [-122.459585, 37.730425],
[-122.459231, 37.730582], [-122.459245, 37.730838], [-122.458191, 37.73088],
[-122.454427, 37.731533], [-122.444271, 37.731596], [-122.444304, 37.728562],
```

[-122.444115, 37.7283], [-122.443605, 37.728313], [-122.439982, 37.729929],
[-122.435069, 37.731505], [-122.433944, 37.732408], [-122.428578, 37.735082],
[-122.427849, 37.734718], [-122.426983, 37.735455], [-122.422133, 37.735162],
[-122.422243, 37.734846], [-122.422901, 37.734388], [-122.421827, 37.732917],
[-122.421889, 37.732533], [-122.422997, 37.732224], [-122.423493, 37.732317],
[-122.4237, 37.732172], [-122.425256, 37.732125], [-122.425944, 37.731698],
[-122.423718, 37.73155], [-122.420701, 37.731936]]]}]}}, {"id": "5", "type":
"Feature", "properties": {"id": "94114", "median_score": 94.0, "neighborhood":
"Castro", "postal5": "94114"}, "geometry": {"type": "GeometryCollection",
"geometries": [{"type": "Polygon", "coordinates": [[[-122.42828, 37.769641],
[-122.428262, 37.769541], [-122.426242, 37.769644], [-122.426896, 37.769009],
[-122.424928, 37.748628], [-122.438168, 37.747841], [-122.438259, 37.748644],
[-122.441209, 37.748622], [-122.441851, 37.748537], [-122.442512, 37.748218],
[-122.44399, 37.746777], [-122.444318, 37.747004], [-122.444685, 37.746982],
[-122.444081, 37.747453], [-122.44411, 37.74842], [-122.44345, 37.749502],
[-122.44273, 37.752373], [-122.442509, 37.75242], [-122.441838, 37.753439],
[-122.441256, 37.753916], [-122.440108, 37.755715], [-122.440154, 37.756433],
[-122.441013, 37.756239], [-122.441426, 37.756562], [-122.442224, 37.756707],
[-122.441394, 37.756449], [-122.441139, 37.756211], [-122.442884, 37.755494],
[-122.443334, 37.75639], [-122.444617, 37.757198], [-122.445648, 37.757231],
[-122.446474, 37.757533], [-122.446912, 37.757892], [-122.447093, 37.757671],
[-122.447003, 37.75746], [-122.445905, 37.756932], [-122.44605, 37.756429],
[-122.445121, 37.756601], [-122.444679, 37.756524], [-122.444373, 37.756212],
[-122.44423, 37.755714], [-122.44423, 37.755536], [-122.444746, 37.755293],
[-122.444639, 37.754422], [-122.445121, 37.75366], [-122.445427, 37.75266],
[-122.445085, 37.751871], [-122.445069, 37.751444], [-122.445288, 37.750918],
[-122.445771, 37.750465], [-122.445935, 37.750721], [-122.445661, 37.751657],
[-122.446088, 37.752524], [-122.446427, 37.752725], [-122.446108, 37.753046],
[-122.445997, 37.753716], [-122.447214, 37.754064], [-122.447519, 37.753835],
[-122.44743, 37.754333], [-122.448641, 37.754793], [-122.449441, 37.755474],
[-122.449611, 37.756321], [-122.450168, 37.756772], [-122.452095, 37.756675],
[-122.452084, 37.756971], [-122.452354, 37.757054], [-122.453279, 37.7567],
[-122.45409, 37.757114], [-122.453759, 37.757568], [-122.453158, 37.757986],
[-122.451348, 37.75859], [-122.451531, 37.759473], [-122.448095, 37.759665],
[-122.447637, 37.759222], [-122.447468, 37.759318], [-122.447146, 37.75969],
[-122.446481, 37.76113], [-122.44679, 37.761792], [-122.445302, 37.761876],
[-122.443776, 37.763352], [-122.443017, 37.76351], [-122.442944, 37.764111],
[-122.443207, 37.764522], [-122.443242, 37.765229], [-122.443426, 37.765368],
[-122.44122, 37.765314], [-122.439489, 37.766572], [-122.438575, 37.766771],
[-122.438094, 37.766709], [-122.43819, 37.767203], [-122.436477, 37.767293],
[-122.436632, 37.769031], [-122.429137, 37.769452], [-122.429208, 37.770372],
[-122.42828, 37.769641]]]}]}}, {"id": "6", "type": "Feature", "properties":
{"id": "94122", "median_score": 90.0, "neighborhood": "Sunset", "postal5":
"94122"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.456648, 37.763844], [-122.46085, 37.762616],
[-122.46065, 37.760577], [-122.46179, 37.760505], [-122.461766, 37.760113],
[-122.461619, 37.759868], [-122.461341, 37.759789], [-122.46069, 37.760089],

[-122.460528, 37.759536], [-122.462483, 37.759342], [-122.462935, 37.75906],
[-122.463213, 37.75861], [-122.462755, 37.756882], [-122.463648, 37.756636],
[-122.463559, 37.755159], [-122.463826, 37.753998], [-122.463714, 37.753509],
[-122.463247, 37.75286], [-122.467947, 37.752765], [-122.46821, 37.752963],
[-122.468791, 37.754241], [-122.469991, 37.754686], [-122.470471, 37.753431],
[-122.469909, 37.752784], [-122.470711, 37.752359], [-122.470866, 37.752536],
[-122.473168, 37.752564], [-122.508181, 37.750987], [-122.507941, 37.749124],
[-122.50849, 37.749138], [-122.509105, 37.749743], [-122.509928, 37.749936],
[-122.509922, 37.7502], [-122.510421, 37.750956], [-122.510753, 37.752401],
[-122.510919, 37.752598], [-122.511003, 37.754535], [-122.51121, 37.755586],
[-122.510919, 37.756636], [-122.511875, 37.760314], [-122.511958, 37.762154],
[-122.512332, 37.763007], [-122.512194, 37.764109], [-122.512489, 37.764633],
[-122.513273, 37.771077], [-122.511322, 37.771173], [-122.511398, 37.771486],
[-122.510968, 37.771331], [-122.47172, 37.773141], [-122.471785, 37.772681],
[-122.472237, 37.77236], [-122.474834, 37.772518], [-122.478552, 37.772254],
[-122.479118, 37.772006], [-122.479409, 37.771509], [-122.479165, 37.770749],
[-122.476909, 37.771533], [-122.475216, 37.771681], [-122.47378, 37.771182],
[-122.474249, 37.771029], [-122.475898, 37.771087], [-122.476249, 37.770754],
[-122.476205, 37.770261], [-122.475251, 37.770068], [-122.47402, 37.769507],
[-122.471528, 37.769845], [-122.471438, 37.769604], [-122.472672, 37.768601],
[-122.472844, 37.767756], [-122.472697, 37.767444], [-122.472231, 37.768606],
[-122.471411, 37.769112], [-122.470265, 37.769228], [-122.469282, 37.769118],
[-122.466956, 37.768001], [-122.464536, 37.769638], [-122.461874, 37.770266],
[-122.460096, 37.770396], [-122.459502, 37.770711], [-122.459148, 37.771316],
[-122.462072, 37.771779], [-122.464749, 37.772486], [-122.465845, 37.772545],
[-122.465887, 37.773401], [-122.454676, 37.774758], [-122.452974, 37.766367],
[-122.456862, 37.765856], [-122.457805, 37.765978], [-122.457707, 37.764705],
[-122.457575, 37.764847], [-122.456778, 37.764938], [-122.456648,
37.763844]]]}]}}, {"id": "7", "type": "Feature", "properties": {"id": "94124",
"median_score": 92.0, "neighborhood": "Bayview", "postal5": "94124"},
"geometry": {"type": "GeometryCollection", "geometries": [{"type": "Polygon",
"coordinates": [[[-122.380089, 37.752759], [-122.380062, 37.752064],
[-122.376159, 37.752283], [-122.375723, 37.74872], [-122.376358, 37.748638],
[-122.376356, 37.747159], [-122.375057, 37.746841], [-122.374811, 37.746423],
[-122.374965, 37.745818], [-122.374157, 37.745048], [-122.373929, 37.745014],
[-122.373374, 37.745414], [-122.372797, 37.745573], [-122.367633, 37.740199],
[-122.368216, 37.739935], [-122.373216, 37.73973], [-122.373243, 37.739554],
[-122.373234, 37.739297], [-122.372406, 37.739007], [-122.36792, 37.739216],
[-122.372423, 37.738953], [-122.373303, 37.739284], [-122.373379, 37.739718],
[-122.374008, 37.739622], [-122.374066, 37.739462], [-122.373496, 37.738645],
[-122.372685, 37.738317], [-122.367541, 37.73834], [-122.367579, 37.738218],
[-122.370108, 37.737729], [-122.371392, 37.737134], [-122.372328, 37.737065],
[-122.373574, 37.737233], [-122.374046, 37.737409], [-122.374481, 37.737844],
[-122.375282, 37.738105], [-122.375371, 37.736427], [-122.376029, 37.735769],
[-122.376156, 37.735123], [-122.375315, 37.735416], [-122.374946, 37.734928],
[-122.375011, 37.734747], [-122.375334, 37.734633], [-122.375053, 37.734363],
[-122.375048, 37.734148], [-122.375549, 37.733124], [-122.374324, 37.732569],

```
[-122.373714, 37.732963], [-122.372437, 37.734205], [-122.37215, 37.734273],
[-122.370831, 37.733231], [-122.370126, 37.733662], [-122.370067, 37.733579],
[-122.370758, 37.733162], [-122.369578, 37.732404], [-122.369375, 37.732034],
[-122.368745, 37.731968], [-122.368337, 37.732254], [-122.367411, 37.731931],
[-122.366864, 37.732334], [-122.366683, 37.732285], [-122.36527, 37.73386],
[-122.365183, 37.73375], [-122.366567, 37.732159], [-122.365638, 37.731617],
[-122.364783, 37.7326], [-122.365096, 37.732096], [-122.363719, 37.731331],
[-122.362922, 37.732232], [-122.362608, 37.732045], [-122.363442, 37.731154],
[-122.362856, 37.730839], [-122.362059, 37.731724], [-122.361774, 37.731576],
[-122.362569, 37.730661], [-122.36196, 37.730325], [-122.362054, 37.730155],
[-122.360898, 37.729983], [-122.36075, 37.730276], [-122.36, 37.730379],
[-122.35887, 37.729814], [-122.358962, 37.72958], [-122.362149, 37.728811],
[-122.36226, 37.728532], [-122.361816, 37.728518], [-122.357767, 37.72947],
[-122.357503, 37.729286], [-122.357846, 37.729008], [-122.36006, 37.728502],
[-122.360449, 37.728261], [-122.359967, 37.728224], [-122.357272, 37.728832],
[-122.357031, 37.728693], [-122.357735, 37.726304], [-122.358027, 37.726114],
[-122.361677, 37.725257], [-122.361297, 37.724209], [-122.357944, 37.724993],
[-122.357849, 37.724678], [-122.361434, 37.723926], [-122.364857, 37.726094],
[-122.365156, 37.725974], [-122.36501, 37.725725], [-122.362287, 37.723914],
[-122.362707, 37.723498], [-122.360005, 37.721674], [-122.360273, 37.721388],
[-122.363043, 37.72303], [-122.363836, 37.722158], [-122.359153, 37.719359],
[-122.359931, 37.718502], [-122.36272, 37.720107], [-122.363655, 37.719096],
[-122.362576, 37.718466], [-122.362678, 37.718312], [-122.358611, 37.715909],
[-122.35875, 37.715777], [-122.362668, 37.718056], [-122.363859, 37.716766],
[-122.359959, 37.71448], [-122.360097, 37.71437], [-122.363988, 37.716612],
[-122.365229, 37.715388], [-122.365229, 37.715249], [-122.361115, 37.712882],
[-122.361226, 37.712794], [-122.365443, 37.715293], [-122.364475, 37.716341],
[-122.364642, 37.716517], [-122.364892, 37.716451], [-122.365336, 37.71585],
[-122.365513, 37.715821], [-122.366008, 37.716085], [-122.366426, 37.716799],
[-122.367697, 37.717613], [-122.368134, 37.717782], [-122.36953, 37.717858],
[-122.369955, 37.71812], [-122.370353, 37.718688], [-122.371784, 37.719011],
[-122.374191, 37.718934], [-122.374317, 37.719602], [-122.374531, 37.719878],
[-122.375775, 37.720586], [-122.376094, 37.721673], [-122.376073, 37.722751],
[-122.376755, 37.723458], [-122.376818, 37.724031], [-122.377564, 37.724132],
[-122.378863, 37.723846], [-122.379928, 37.722313], [-122.380731, 37.72173],
[-122.382591, 37.72241], [-122.383777, 37.723126], [-122.383999, 37.723489],
[-122.385167, 37.72411], [-122.386436, 37.724479], [-122.386835, 37.724339],
[-122.386911, 37.724148], [-122.382843, 37.721921], [-122.382765, 37.721547],
[-122.383381, 37.720627], [-122.383253, 37.720135], [-122.381696, 37.71932],
[-122.381353, 37.718785], [-122.38058, 37.718802], [-122.379724, 37.71832],
[-122.379915, 37.71797], [-122.379807, 37.717843], [-122.379217, 37.717818],
[-122.376187, 37.716103], [-122.377399, 37.714411], [-122.378896, 37.713077],
[-122.379684, 37.711801], [-122.380004, 37.711586], [-122.380035, 37.710681],
[-122.378821, 37.710024], [-122.3783, 37.709279], [-122.3783, 37.708891],
[-122.378012, 37.708681], [-122.376979, 37.70895], [-122.376436, 37.7097],
[-122.376085, 37.70975], [-122.375571, 37.709661], [-122.374625, 37.70906],
[-122.374427, 37.708743], [-122.374662, 37.708562], [-122.375646, 37.708409],
```

[-122.379472, 37.708427], [-122.380428, 37.708795], [-122.384507, 37.709281],
[-122.385965, 37.709993], [-122.386977, 37.709655], [-122.388155, 37.708672],
[-122.389693, 37.709556], [-122.38991, 37.70958], [-122.390654, 37.709186],
[-122.391782, 37.709808], [-122.39094, 37.710318], [-122.38874, 37.710489],
[-122.386724, 37.711066], [-122.386375, 37.711033], [-122.387684, 37.712178],
[-122.387994, 37.713004], [-122.388774, 37.71228], [-122.390439, 37.712109],
[-122.390961, 37.712322], [-122.391224, 37.713259], [-122.393041, 37.713609],
[-122.394738, 37.713611], [-122.396527, 37.714716], [-122.396489, 37.714972],
[-122.398027, 37.715493], [-122.398393, 37.716337], [-122.398555, 37.717783],
[-122.399202, 37.71992], [-122.401262, 37.724794], [-122.402107, 37.727849],
[-122.404946, 37.733588], [-122.406506, 37.735623], [-122.407024, 37.738016],
[-122.408009, 37.737734], [-122.408136, 37.73964], [-122.407045, 37.739587],
[-122.406936, 37.740554], [-122.406652, 37.741241], [-122.404524, 37.744285],
[-122.404865, 37.744385], [-122.405493, 37.744084], [-122.405614, 37.7442],
[-122.405091, 37.745281], [-122.403798, 37.749443], [-122.401546, 37.74945],
[-122.391705, 37.750133], [-122.387666, 37.750267], [-122.387833, 37.752817],
[-122.382933, 37.753184], [-122.382078, 37.753131], [-122.381482, 37.752782],
[-122.380089, 37.752759]]]}]}}, {"id": "8", "type": "Feature", "properties":
{"id": "94127", "median_score": 92.0, "neighborhood": "Portola", "postal5":
"94127"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.444271, 37.731596], [-122.454427, 37.731533],
[-122.458191, 37.73088], [-122.459245, 37.730838], [-122.459231, 37.730582],
[-122.459585, 37.730425], [-122.459911, 37.729952], [-122.460039, 37.728777],
[-122.460858, 37.728754], [-122.462066, 37.728095], [-122.462888, 37.728087],
[-122.462587, 37.727158], [-122.462838, 37.725488], [-122.46227, 37.72527],
[-122.462297, 37.723037], [-122.464104, 37.722246], [-122.464475, 37.721682],
[-122.471815, 37.721661], [-122.471677, 37.720951], [-122.471654, 37.71788],
[-122.472453, 37.717877], [-122.47226, 37.721627], [-122.472667, 37.721562],
[-122.472676, 37.722124], [-122.472602, 37.724834], [-122.472364, 37.726819],
[-122.472829, 37.726836], [-122.47269, 37.728843], [-122.472283, 37.728807],
[-122.472046, 37.733756], [-122.471751, 37.733878], [-122.47168, 37.734642],
[-122.471464, 37.734653], [-122.471543, 37.7351], [-122.470664, 37.736636],
[-122.470972, 37.737018], [-122.471224, 37.741374], [-122.468572, 37.741502],
[-122.467798, 37.743412], [-122.463662, 37.743771], [-122.461372, 37.745591],
[-122.460766, 37.745108], [-122.458701, 37.74661], [-122.458952, 37.746923],
[-122.455093, 37.746263], [-122.453776, 37.745686], [-122.451731, 37.745527],
[-122.449493, 37.743019], [-122.448814, 37.742758], [-122.446663, 37.742452],
[-122.446266, 37.742106], [-122.4461, 37.741372], [-122.445874, 37.741102],
[-122.444651, 37.740775], [-122.443888, 37.740074], [-122.442865, 37.739757],
[-122.44256, 37.739472], [-122.442635, 37.738458], [-122.442409, 37.737629],
[-122.442603, 37.737183], [-122.443662, 37.736503], [-122.444158, 37.736536],
[-122.444385, 37.736182], [-122.444704, 37.736429], [-122.444598, 37.737092],
[-122.445914, 37.737074], [-122.445737, 37.736525], [-122.446381, 37.73578],
[-122.44633, 37.735245], [-122.445366, 37.734517], [-122.445397, 37.734267],
[-122.44573, 37.733938], [-122.445652, 37.733835], [-122.444319, 37.733831],
[-122.444271, 37.731596]]]}]}}, {"id": "9", "type": "Feature", "properties":
{"id": "94132", "median_score": 96.0, "neighborhood": "Lake Merced", "postal5":

"94132"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"MultiPolygon", "coordinates": [[[[-122.486279, 37.736821], [-122.485731,
37.736825], [-122.484755, 37.737207], [-122.482882, 37.737513], [-122.481446,
37.737249], [-122.47097, 37.737652], [-122.470945, 37.736905], [-122.470664,
37.736636], [-122.471543, 37.7351], [-122.471464, 37.734653], [-122.47168,
37.734642], [-122.471751, 37.733878], [-122.472046, 37.733756], [-122.472283,
37.728807], [-122.47269, 37.728843], [-122.472829, 37.726836], [-122.472364,
37.726819], [-122.472602, 37.724834], [-122.472676, 37.722124], [-122.472667,
37.721562], [-122.47226, 37.721627], [-122.472453, 37.717877], [-122.471654,
37.71788], [-122.471677, 37.720951], [-122.471815, 37.721661], [-122.464475,
37.721682], [-122.464104, 37.722246], [-122.462297, 37.723037], [-122.462264,
37.720021], [-122.46268, 37.720018], [-122.462626, 37.714469], [-122.462758,
37.714264], [-122.462562, 37.713806], [-122.462598, 37.711377], [-122.462503,
37.711207], [-122.460894, 37.710579], [-122.464028, 37.710549], [-122.465786,
37.710243], [-122.46745, 37.709496], [-122.469004, 37.708187], [-122.471987,
37.708406], [-122.485555, 37.70826], [-122.485383, 37.708507], [-122.485717,
37.70826], [-122.50278, 37.70805], [-122.503255, 37.709186], [-122.503881,
37.712656], [-122.504632, 37.714564], [-122.504809, 37.715626], [-122.505174,
37.716546], [-122.505394, 37.718032], [-122.506133, 37.720154], [-122.506222,
37.721235], [-122.506473, 37.721616], [-122.506687, 37.723115], [-122.507043,
37.72379], [-122.507151, 37.72527], [-122.507551, 37.72704], [-122.507223,
37.72862], [-122.508116, 37.731333], [-122.507951, 37.732438], [-122.508526,
37.733837], [-122.508172, 37.73443], [-122.508396, 37.734631], [-122.50842,
37.735401], [-122.508573, 37.735638], [-122.507884, 37.735942], [-122.506807,
37.735587], [-122.502009, 37.735517], [-122.500874, 37.735256], [-122.498765,
37.734264], [-122.497435, 37.733934], [-122.493786, 37.733991], [-122.49343,
37.734328], [-122.492622, 37.734039], [-122.491528, 37.734094], [-122.491263,
37.734165], [-122.491357, 37.735345], [-122.490312, 37.735638], [-122.490714,
37.736565], [-122.491319, 37.737325], [-122.490655, 37.737821], [-122.489989,
37.737975], [-122.488812, 37.73713], [-122.486279, 37.736821]]]]}]}}, {"id":
"10", "type": "Feature", "properties": {"id": "94131", "median_score": 96.0,
"neighborhood": "Twin Peaks", "postal5": "94131"}, "geometry": {"type":
"GeometryCollection", "geometries": [{"type": "Polygon", "coordinates":
[[[-122.452936, 37.7581], [-122.453759, 37.757568], [-122.45409, 37.757114],
[-122.453279, 37.7567], [-122.452354, 37.757054], [-122.452084, 37.756971],
[-122.452095, 37.756675], [-122.450168, 37.756772], [-122.449611, 37.756321],
[-122.449441, 37.755474], [-122.448641, 37.754793], [-122.44743, 37.754333],
[-122.447519, 37.753835], [-122.447214, 37.754064], [-122.445997, 37.753716],
[-122.446108, 37.753046], [-122.446427, 37.752725], [-122.446088, 37.752524],
[-122.445661, 37.751657], [-122.445935, 37.750721], [-122.445771, 37.750465],
[-122.445288, 37.750918], [-122.445069, 37.751444], [-122.445085, 37.751871],
[-122.445427, 37.75266], [-122.445121, 37.75366], [-122.444639, 37.754422],
[-122.444746, 37.755293], [-122.44423, 37.755536], [-122.44423, 37.755714],
[-122.444373, 37.756212], [-122.444679, 37.756524], [-122.445121, 37.756601],
[-122.44605, 37.756429], [-122.445905, 37.756932], [-122.447003, 37.75746],
[-122.447093, 37.757671], [-122.446912, 37.757892], [-122.446474, 37.757533],
[-122.445648, 37.757231], [-122.444617, 37.757198], [-122.443334, 37.75639],

[-122.442884, 37.755494], [-122.441139, 37.756211], [-122.441394, 37.756449],
[-122.442072, 37.756597], [-122.442224, 37.756707], [-122.441426, 37.756562],
[-122.441013, 37.756239], [-122.440154, 37.756433], [-122.440108, 37.755715],
[-122.441256, 37.753916], [-122.441838, 37.753439], [-122.442509, 37.75242],
[-122.44273, 37.752373], [-122.44345, 37.749502], [-122.44411, 37.74842],
[-122.444081, 37.747453], [-122.444685, 37.746982], [-122.444318, 37.747004],
[-122.44399, 37.746777], [-122.442512, 37.748218], [-122.441851, 37.748537],
[-122.441092, 37.748632], [-122.438259, 37.748644], [-122.438168, 37.747841],
[-122.424928, 37.748628], [-122.424168, 37.740805], [-122.424394, 37.740405],
[-122.42427, 37.739867], [-122.42452, 37.739868], [-122.42551, 37.737774],
[-122.428454, 37.735882], [-122.42801, 37.735441], [-122.433944, 37.732408],
[-122.435069, 37.731505], [-122.439982, 37.729929], [-122.443605, 37.728313],
[-122.444115, 37.7283], [-122.444304, 37.728562], [-122.444319, 37.733831],
[-122.445652, 37.733835], [-122.44573, 37.733938], [-122.445397, 37.734267],
[-122.445366, 37.734517], [-122.44633, 37.735245], [-122.446387, 37.735748],
[-122.445737, 37.736525], [-122.445914, 37.737074], [-122.444598, 37.737092],
[-122.444704, 37.736429], [-122.444385, 37.736182], [-122.444158, 37.736536],
[-122.443662, 37.736503], [-122.442523, 37.737263], [-122.442409, 37.737629],
[-122.442628, 37.738387], [-122.442523, 37.739283], [-122.442631, 37.739589],
[-122.443888, 37.740074], [-122.444651, 37.740775], [-122.445874, 37.741102],
[-122.4461, 37.741372], [-122.446266, 37.742106], [-122.446663, 37.742452],
[-122.448814, 37.742758], [-122.449493, 37.743019], [-122.451731, 37.745527],
[-122.453776, 37.745686], [-122.455093, 37.746263], [-122.458952, 37.746923],
[-122.45919, 37.747199], [-122.458708, 37.747608], [-122.458717, 37.748113],
[-122.460528, 37.749935], [-122.461227, 37.751194], [-122.463247, 37.75286],
[-122.46379, 37.753707], [-122.463554, 37.75526], [-122.463648, 37.756636],
[-122.462755, 37.756882], [-122.463213, 37.75861], [-122.462935, 37.75906],
[-122.462483, 37.759342], [-122.460528, 37.759536], [-122.46069, 37.760089],
[-122.461341, 37.759789], [-122.461619, 37.759868], [-122.461766, 37.760113],
[-122.46179, 37.760505], [-122.46065, 37.760577], [-122.46085, 37.762616],
[-122.455731, 37.764099], [-122.455465, 37.760271], [-122.453594, 37.761617],
[-122.452405, 37.761578], [-122.45182, 37.758743], [-122.452997, 37.758202],
[-122.452936, 37.7581]]]]}]}}, {"id": "11", "type": "Feature", "properties":
{"id": "94133", "median_score": 92.0, "neighborhood": "North Beach", "postal5":
"94133"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"MultiPolygon", "coordinates": [[[[-122.419083, 37.811572], [-122.415375,
37.809166], [-122.415194, 37.809226], [-122.415403, 37.809424], [-122.415085,
37.809231], [-122.414794, 37.809331], [-122.415315, 37.809767], [-122.415212,
37.809848], [-122.413444, 37.809247], [-122.413371, 37.809391], [-122.414276,
37.809863], [-122.414024, 37.81014], [-122.413789, 37.809946], [-122.413868,
37.809874], [-122.412638, 37.809089], [-122.41239, 37.809118], [-122.411911,
37.809384], [-122.412856, 37.810523], [-122.412717, 37.810604], [-122.411402,
37.809035], [-122.41056, 37.808843], [-122.410475, 37.80937], [-122.411323,
37.810958], [-122.410669, 37.811379], [-122.410203, 37.811311], [-122.409973,
37.810966], [-122.410179, 37.810095], [-122.409611, 37.809141], [-122.409242,
37.809049], [-122.409109, 37.808848], [-122.409089, 37.808138], [-122.405774,
37.806775], [-122.405899, 37.806399], [-122.405462, 37.806107], [-122.405175,

37.804795], [-122.40596, 37.80469], [-122.405752, 37.803725], [-122.404953,
37.803822], [-122.405067, 37.802885], [-122.403175, 37.803098], [-122.402791,
37.801265], [-122.4036, 37.801195], [-122.403549, 37.800548], [-122.404359,
37.800576], [-122.404329, 37.800423], [-122.404265, 37.80013], [-122.403422,
37.800209], [-122.403765, 37.799746], [-122.403656, 37.799212], [-122.402423,
37.799369], [-122.401864, 37.796638], [-122.405152, 37.796213], [-122.404959,
37.795337], [-122.405526, 37.795292], [-122.40568, 37.796152], [-122.406649,
37.796027], [-122.406591, 37.795697], [-122.407014, 37.795587], [-122.407091,
37.79597], [-122.411087, 37.795467], [-122.411024, 37.795145], [-122.411484,
37.795033], [-122.411553, 37.795409], [-122.414883, 37.79503], [-122.415574,
37.798688], [-122.417224, 37.798494], [-122.41741, 37.799405], [-122.418226,
37.799295], [-122.418252, 37.799884], [-122.417548, 37.799967], [-122.417814,
37.80132], [-122.419075, 37.801144], [-122.419167, 37.801559], [-122.417893,
37.80172], [-122.418129, 37.803168], [-122.418468, 37.804109], [-122.418349,
37.804101], [-122.418789, 37.806249], [-122.4196, 37.806783], [-122.418879,
37.806877], [-122.419084, 37.807804], [-122.420746, 37.807598], [-122.420342,
37.807746], [-122.420432, 37.808308], [-122.417911, 37.808629], [-122.417992,
37.808864], [-122.419957, 37.808915], [-122.41998, 37.809133], [-122.418052,
37.809139], [-122.417879, 37.808993], [-122.417733, 37.808362], [-122.417454,
37.80838], [-122.417364, 37.808135], [-122.4164, 37.808303], [-122.416315,
37.808478], [-122.4172, 37.808528], [-122.416291, 37.808576], [-122.416241,
37.808862], [-122.417477, 37.808903], [-122.417448, 37.809198], [-122.420591,
37.811315], [-122.420379, 37.811535], [-122.418938, 37.810817], [-122.419471,
37.811444], [-122.419083, 37.811572]]]]}]}}, {"id": "12", "type": "Feature",
"properties": {"id": "94102", "median_score": 92.0, "neighborhood":
"Tenderloin", "postal5": "94102"}, "geometry": {"type": "GeometryCollection",
"geometries": [{"type": "Polygon", "coordinates": [[[-122.405883, 37.785718],
[-122.426242, 37.769644], [-122.428262, 37.769541], [-122.429573, 37.776045],
[-122.4294, 37.776267], [-122.429593, 37.776526], [-122.429983, 37.778864],
[-122.426829, 37.779265], [-122.427427, 37.782055], [-122.414256, 37.783728],
[-122.414805, 37.786523], [-122.413165, 37.786726], [-122.413356, 37.787657],
[-122.411712, 37.78787], [-122.411913, 37.788812], [-122.408655, 37.789244],
[-122.408244, 37.787369], [-122.406602, 37.787583], [-122.406236, 37.785793],
[-122.405883, 37.785718]]]}]}}, {"id": "13", "type": "Feature", "properties":
{"id": "94103", "median_score": 91.0, "neighborhood": "SoMa", "postal5":
"94103"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.398941, 37.783785], [-122.405629, 37.778502],
[-122.399425, 37.773612], [-122.403723, 37.770182], [-122.404003, 37.77018],
[-122.403477, 37.769842], [-122.402105, 37.769912], [-122.402091, 37.769716],
[-122.401685, 37.769721], [-122.401768, 37.769145], [-122.402021, 37.768998],
[-122.401851, 37.767416], [-122.400881, 37.767518], [-122.400915, 37.768096],
[-122.40001, 37.767614], [-122.400422, 37.767291], [-122.3998, 37.766807],
[-122.399688, 37.765005], [-122.407431, 37.764497], [-122.407553, 37.765798],
[-122.419459, 37.765089], [-122.421248, 37.764947], [-122.421031, 37.76335],
[-122.42173, 37.763304], [-122.421885, 37.764908], [-122.426453, 37.764634],
[-122.426897, 37.769107], [-122.42245, 37.772761], [-122.403427, 37.787676],
[-122.40333, 37.787336], [-122.401497, 37.785824], [-122.39999, 37.787004],

[-122.399523, 37.786631], [-122.400955, 37.785389], [-122.398941,
37.783785]]]}]}}, {"id": "14", "type": "Feature", "properties": {"id": "94110",
"median_score": 90.0, "neighborhood": "Mission", "postal5": "94110"},
"geometry": {"type": "GeometryCollection", "geometries": [{"type": "Polygon",
"coordinates": [[[-122.405115, 37.764635], [-122.405212, 37.763469],
[-122.405832, 37.762199], [-122.405859, 37.761815], [-122.40604, 37.761865],
[-122.406365, 37.761199], [-122.40645, 37.760114], [-122.406167, 37.759315],
[-122.406015, 37.759409], [-122.405604, 37.758851], [-122.40405, 37.757619],
[-122.403428, 37.756871], [-122.403585, 37.756818], [-122.403328, 37.756082],
[-122.40315, 37.754488], [-122.403524, 37.754463], [-122.403437, 37.753199],
[-122.403364, 37.752366], [-122.403022, 37.752336], [-122.403052, 37.752048],
[-122.405091, 37.745281], [-122.405614, 37.7442], [-122.405493, 37.744084],
[-122.404865, 37.744385], [-122.404524, 37.744285], [-122.406652, 37.741241],
[-122.406936, 37.740554], [-122.407045, 37.739587], [-122.408136, 37.73964],
[-122.408009, 37.737734], [-122.408218, 37.73765], [-122.408284, 37.736846],
[-122.408613, 37.736074], [-122.409287, 37.735258], [-122.410052, 37.734675],
[-122.411978, 37.733732], [-122.412695, 37.733197], [-122.414554, 37.732371],
[-122.416255, 37.732034], [-122.419881, 37.732016], [-122.423718, 37.73155],
[-122.425944, 37.731698], [-122.425256, 37.732125], [-122.422274, 37.732383],
[-122.421889, 37.732533], [-122.421787, 37.732774], [-122.422472, 37.733967],
[-122.422901, 37.734388], [-122.422243, 37.734846], [-122.422133, 37.735162],
[-122.426983, 37.735455], [-122.427849, 37.734718], [-122.428578, 37.735082],
[-122.42801, 37.735441], [-122.428454, 37.735882], [-122.42551, 37.737774],
[-122.42452, 37.739868], [-122.42427, 37.739867], [-122.424394, 37.740405],
[-122.424168, 37.740805], [-122.426453, 37.764634], [-122.421885, 37.764908],
[-122.42173, 37.763304], [-122.421031, 37.76335], [-122.421248, 37.764947],
[-122.407553, 37.765798], [-122.407431, 37.764497], [-122.405115,
37.764635]]]}]}}, {"id": "15", "type": "Feature", "properties": {"id": "94111",
"median_score": 91.0, "neighborhood": "Financial District", "postal5": "94111"},
"geometry": {"type": "GeometryCollection", "geometries": [{"type": "Polygon",
"coordinates": [[[-122.391701, 37.794113], [-122.392429, 37.793835],
[-122.393875, 37.795248], [-122.399156, 37.791051], [-122.399607, 37.791317],
[-122.400146, 37.79415], [-122.404629, 37.79358], [-122.405152, 37.796213],
[-122.401864, 37.796638], [-122.402423, 37.799369], [-122.403656, 37.799212],
[-122.403765, 37.799746], [-122.403422, 37.800209], [-122.404265, 37.80013],
[-122.404329, 37.800423], [-122.404359, 37.800576], [-122.403549, 37.800548],
[-122.4036, 37.801195], [-122.402791, 37.801265], [-122.403175, 37.803098],
[-122.405067, 37.802885], [-122.404953, 37.803822], [-122.405752, 37.803725],
[-122.40596, 37.80469], [-122.405175, 37.804795], [-122.405462, 37.806107],
[-122.405899, 37.806399], [-122.405774, 37.806775], [-122.409089, 37.808138],
[-122.409004, 37.808563], [-122.407936, 37.808146], [-122.407617, 37.808244],
[-122.407477, 37.808085], [-122.406554, 37.807891], [-122.406816, 37.810064],
[-122.406209, 37.810103], [-122.405976, 37.807252], [-122.405313, 37.806883],
[-122.404321, 37.808927], [-122.403972, 37.808829], [-122.404712, 37.807015],
[-122.403955, 37.806592], [-122.402774, 37.808053], [-122.402375, 37.807896],
[-122.403406, 37.806516], [-122.402824, 37.806076], [-122.400944, 37.807679],
[-122.400408, 37.807307], [-122.40097, 37.803579], [-122.400512, 37.8035],

```
[-122.39855, 37.80465], [-122.398217, 37.804315], [-122.399938, 37.803283],
[-122.399597, 37.802882], [-122.397818, 37.803862], [-122.397518, 37.803526],
[-122.399718, 37.802209], [-122.399419, 37.801867], [-122.399203, 37.801906],
[-122.397115, 37.803089], [-122.39613, 37.801975], [-122.3984, 37.80072],
[-122.398001, 37.800267], [-122.395772, 37.801535], [-122.39544, 37.80118],
[-122.397667, 37.7999], [-122.396766, 37.798884], [-122.39658, 37.798852],
[-122.394503, 37.800155], [-122.394306, 37.80013], [-122.394252, 37.799996],
[-122.396516, 37.798686], [-122.395855, 37.797941], [-122.395655, 37.797922],
[-122.393825, 37.798956], [-122.393551, 37.798652], [-122.395468, 37.797463],
[-122.395024, 37.797575], [-122.394882, 37.797427], [-122.39512, 37.797145],
[-122.393219, 37.798276], [-122.392969, 37.797941], [-122.394774, 37.796778],
[-122.394682, 37.796659], [-122.3942, 37.796922], [-122.393041, 37.795634],
[-122.391593, 37.796229], [-122.391211, 37.795822], [-122.392391, 37.794856],
[-122.392292, 37.794737], [-122.391934, 37.794842], [-122.391668, 37.79458],
[-122.391934, 37.794376], [-122.391701, 37.794113]]]}]}}, {"id": "16", "type":
"Feature", "properties": {"id": "94115", "median_score": 90.0, "neighborhood":
"Lower Pacific Heights", "postal5": "94115"}, "geometry": {"type":
"GeometryCollection", "geometries": [{"type": "Polygon", "coordinates":
[[[-122.427427, 37.782055], [-122.426829, 37.779265], [-122.444972, 37.776956],
[-122.445357, 37.778831], [-122.446084, 37.778731], [-122.446307, 37.77984],
[-122.446226, 37.780051], [-122.44544, 37.780142], [-122.445536, 37.781341],
[-122.447476, 37.781067], [-122.447533, 37.781349], [-122.447238, 37.782464],
[-122.44584, 37.782546], [-122.447689, 37.791703], [-122.446306, 37.791871],
[-122.446428, 37.792795], [-122.44153, 37.793428], [-122.441719, 37.794387],
[-122.440875, 37.794492], [-122.440679, 37.793537], [-122.430001, 37.794906],
[-122.427427, 37.782055]]]}]}}, {"id": "17", "type": "Feature", "properties":
{"id": "94116", "median_score": 90.5, "neighborhood": "Taraval", "postal5":
"94116"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.463247, 37.75286], [-122.461338, 37.751312],
[-122.460528, 37.749935], [-122.458717, 37.748113], [-122.458708, 37.747608],
[-122.45919, 37.747199], [-122.458701, 37.74661], [-122.460766, 37.745108],
[-122.461372, 37.745591], [-122.463662, 37.743771], [-122.467798, 37.743412],
[-122.468572, 37.741502], [-122.471224, 37.741374], [-122.47097, 37.737652],
[-122.481446, 37.737249], [-122.482882, 37.737513], [-122.486168, 37.736807],
[-122.488812, 37.73713], [-122.489989, 37.737975], [-122.490655, 37.737821],
[-122.491319, 37.737325], [-122.490714, 37.736565], [-122.490312, 37.735638],
[-122.491357, 37.735345], [-122.491263, 37.734165], [-122.491528, 37.734094],
[-122.492622, 37.734039], [-122.49343, 37.734328], [-122.493786, 37.733991],
[-122.497435, 37.733934], [-122.498765, 37.734264], [-122.500874, 37.735256],
[-122.502009, 37.735517], [-122.506807, 37.735587], [-122.507884, 37.735942],
[-122.508573, 37.735638], [-122.50817, 37.736602], [-122.50842, 37.739587],
[-122.508233, 37.740381], [-122.509089, 37.74456], [-122.508879, 37.74557],
[-122.509589, 37.747897], [-122.50961, 37.748912], [-122.509928, 37.749936],
[-122.509105, 37.749743], [-122.50849, 37.749138], [-122.507941, 37.749124],
[-122.508181, 37.750987], [-122.473168, 37.752564], [-122.470866, 37.752536],
[-122.470711, 37.752359], [-122.469909, 37.752784], [-122.470471, 37.753431],
[-122.469991, 37.754686], [-122.468791, 37.754241], [-122.46821, 37.752963],
[-122.467947, 37.752765], [-122.463247, 37.75286]]]}]}}, {"id": "18", "type":
"Feature", "properties": {"id": "94117", "median_score": 91.5, "neighborhood":
"Panhandle", "postal5": "94117"}, "geometry": {"type": "GeometryCollection",
"geometries": [{"type": "Polygon", "coordinates": [[[-122.42828, 37.769641],
[-122.429208, 37.770372], [-122.429137, 37.769452], [-122.436632, 37.769031],
[-122.436477, 37.767293], [-122.43819, 37.767203], [-122.438094, 37.766709],
[-122.438575, 37.766771], [-122.439489, 37.766572], [-122.44122, 37.765314],
[-122.443426, 37.765368], [-122.443242, 37.765229], [-122.443207, 37.764522],
[-122.442944, 37.764111], [-122.443017, 37.76351], [-122.443776, 37.763352],
[-122.445302, 37.761876], [-122.44679, 37.761792], [-122.446481, 37.76113],
[-122.447146, 37.75969], [-122.447468, 37.759318], [-122.447637, 37.759222],
[-122.448095, 37.759665], [-122.451531, 37.759473], [-122.451348, 37.75859],
[-122.452936, 37.7581], [-122.452997, 37.758202], [-122.45182, 37.758743],
[-122.452405, 37.761578], [-122.453594, 37.761617], [-122.455465, 37.760271],
[-122.455731, 37.764099], [-122.456648, 37.763844], [-122.456778, 37.764938],
[-122.457575, 37.764847], [-122.457707, 37.764705], [-122.457805, 37.765978],
[-122.456862, 37.765856], [-122.452974, 37.766367], [-122.454676, 37.774758],
[-122.452852, 37.774986], [-122.453191, 37.776877], [-122.446859, 37.777678],
[-122.446673, 37.776733], [-122.429983, 37.778864], [-122.429593, 37.776526],
[-122.4294, 37.776267], [-122.429573, 37.776045], [-122.42828, 37.769641]]]}]}},
{"id": "19", "type": "Feature", "properties": {"id": "94118", "median_score":
90.0, "neighborhood": "Inner Richmond", "postal5": "94118"}, "geometry":
{"type": "GeometryCollection", "geometries": [{"type": "Polygon", "coordinates":
[[[-122.444972, 37.776956], [-122.446673, 37.776733], [-122.446859, 37.777678],
[-122.453191, 37.776877], [-122.452852, 37.774986], [-122.465887, 37.773401],
[-122.465845, 37.772545], [-122.464749, 37.772486], [-122.462072, 37.771779],
[-122.459148, 37.771316], [-122.459502, 37.770711], [-122.460096, 37.770396],
[-122.461874, 37.770266], [-122.464536, 37.769638], [-122.466956, 37.768001],
[-122.469006, 37.769046], [-122.470813, 37.769209], [-122.471557, 37.769065],
[-122.472056, 37.768783], [-122.472362, 37.7684], [-122.472697, 37.767444],
[-122.472844, 37.767756], [-122.472672, 37.768601], [-122.471497, 37.76954],
[-122.471476, 37.769812], [-122.47402, 37.769507], [-122.475251, 37.770068],
[-122.476153, 37.770232], [-122.476281, 37.770393], [-122.476217, 37.770837],
[-122.475898, 37.771087], [-122.474249, 37.771029], [-122.47378, 37.771182],
[-122.475395, 37.771703], [-122.477216, 37.77146], [-122.479165, 37.770749],
[-122.479412, 37.771247], [-122.479254, 37.771868], [-122.478613, 37.772245],
[-122.474834, 37.772518], [-122.472407, 37.772329], [-122.471937, 37.772516],
[-122.471729, 37.772828], [-122.47172, 37.773141], [-122.475639, 37.772941],
[-122.476602, 37.78702], [-122.475083, 37.787011], [-122.473321, 37.787239],
[-122.472691, 37.786462], [-122.472522, 37.787124], [-122.471777, 37.787485],
[-122.468463, 37.787804], [-122.449327, 37.791654], [-122.449368, 37.791488],
[-122.447689, 37.791703], [-122.44584, 37.782546], [-122.447238, 37.782464],
[-122.447533, 37.781349], [-122.447476, 37.781067], [-122.445536, 37.781341],
[-122.44544, 37.780142], [-122.446226, 37.780051], [-122.446307, 37.77984],
[-122.446084, 37.778731], [-122.445357, 37.778831], [-122.444972,
37.776956]]]}]}}, {"id": "20", "type": "Feature", "properties": {"id": "94121",
"median_score": 90.0, "neighborhood": "Outer Richmond", "postal5": "94121"},
```

"geometry": {"type": "GeometryCollection", "geometries": [{"type":
"MultiPolygon", "coordinates": [[[[-122.515568, 37.778158], [-122.515522,
37.778054], [-122.516044, 37.778034], [-122.5171, 37.778255], [-122.515972,
37.778931], [-122.515659, 37.778709], [-122.51575, 37.778534], [-122.515568,
37.778158]]], [[[-122.517686, 37.780601], [-122.517402, 37.780516],
[-122.517389, 37.780248], [-122.517109, 37.780174], [-122.517043, 37.779939],
[-122.517474, 37.779998], [-122.517747, 37.78023], [-122.517686, 37.780601]]],
[[[-122.484717, 37.789597], [-122.484583, 37.789242], [-122.484718, 37.788811],
[-122.485334, 37.788309], [-122.485284, 37.787608], [-122.483795, 37.787747],
[-122.483607, 37.78743], [-122.48337, 37.78736], [-122.48138, 37.787401],
[-122.479785, 37.787032], [-122.479129, 37.787142], [-122.477715, 37.786931],
[-122.476602, 37.78702], [-122.475639, 37.772941], [-122.510968, 37.771331],
[-122.511398, 37.771486], [-122.511322, 37.771173], [-122.513273, 37.771077],
[-122.513184, 37.774133], [-122.513026, 37.774949], [-122.513142, 37.775182],
[-122.513194, 37.777452], [-122.514068, 37.778184], [-122.514289, 37.778184],
[-122.514373, 37.77944], [-122.514926, 37.779652], [-122.514884, 37.779893],
[-122.514399, 37.780002], [-122.514178, 37.780542], [-122.514736, 37.781041],
[-122.514698, 37.781217], [-122.514205, 37.781328], [-122.514245, 37.781536],
[-122.513721, 37.781503], [-122.513795, 37.781861], [-122.513426, 37.782119],
[-122.513563, 37.782343], [-122.513395, 37.782526], [-122.51349, 37.782784],
[-122.512869, 37.782851], [-122.512785, 37.783408], [-122.512406, 37.783757],
[-122.51267, 37.784013], [-122.512121, 37.783849], [-122.511811, 37.783945],
[-122.511632, 37.784377], [-122.511411, 37.784286], [-122.511348, 37.784585],
[-122.510833, 37.784452], [-122.509747, 37.784795], [-122.509062, 37.785544],
[-122.508124, 37.785907], [-122.506904, 37.787204], [-122.50621, 37.787597],
[-122.506022, 37.787797], [-122.506092, 37.788194], [-122.505811, 37.78826],
[-122.504845, 37.787919], [-122.504094, 37.788082], [-122.503578, 37.787971],
[-122.502917, 37.788146], [-122.500674, 37.788287], [-122.50006, 37.788535],
[-122.499835, 37.788446], [-122.499703, 37.788669], [-122.499497, 37.788669],
[-122.499037, 37.788298], [-122.498381, 37.788217], [-122.498118, 37.787979],
[-122.497677, 37.787987], [-122.497246, 37.78749], [-122.496326, 37.78732],
[-122.49486, 37.78784], [-122.494599, 37.788153], [-122.49389, 37.787551],
[-122.492867, 37.787922], [-122.492545, 37.787773], [-122.492055, 37.787864],
[-122.491482, 37.788136], [-122.490012, 37.789413], [-122.488188, 37.789339],
[-122.487051, 37.789528], [-122.485863, 37.790665], [-122.485131, 37.790421],
[-122.484717, 37.789597]]]]}]}}, {"id": "21", "type": "Feature", "properties":
{"id": "94123", "median_score": 94.0, "neighborhood": "Marina", "postal5":
"94123"}, "geometry": {"type": "GeometryCollection", "geometries": [{"type":
"Polygon", "coordinates": [[[-122.426352, 37.807528], [-122.425397, 37.806565],
[-122.425374, 37.806107], [-122.42553, 37.805821], [-122.423491, 37.795723],
[-122.440679, 37.793537], [-122.440875, 37.794492], [-122.441719, 37.794387],
[-122.44153, 37.793428], [-122.446428, 37.792795], [-122.447811, 37.80152],
[-122.448525, 37.801634], [-122.449009, 37.801917], [-122.449404, 37.802357],
[-122.449561, 37.802801], [-122.449275, 37.803858], [-122.448409, 37.804394],
[-122.448243, 37.804788], [-122.448481, 37.806062], [-122.448321, 37.806876],
[-122.447295, 37.807455], [-122.443705, 37.807725], [-122.441096, 37.808198],
[-122.44055, 37.80839], [-122.440055, 37.808821], [-122.439998, 37.808663],

[-122.44041, 37.808174], [-122.441003, 37.808088], [-122.441364, 37.80786],
[-122.441493, 37.808019], [-122.442101, 37.807966], [-122.442427, 37.807869],
[-122.442406, 37.807668], [-122.44268, 37.807623], [-122.44274, 37.80782],
[-122.442999, 37.807803], [-122.443683, 37.807592], [-122.443643, 37.807467],
[-122.447471, 37.806461], [-122.44728, 37.805577], [-122.447182, 37.805402],
[-122.442496, 37.806008], [-122.442604, 37.806591], [-122.444053, 37.806469],
[-122.442517, 37.806758], [-122.442471, 37.807123], [-122.442303, 37.806809],
[-122.435826, 37.807598], [-122.435633, 37.806878], [-122.434208, 37.80697],
[-122.433816, 37.805671], [-122.432637, 37.805729], [-122.432535, 37.805917],
[-122.432195, 37.806011], [-122.432646, 37.808535], [-122.432349, 37.808561],
[-122.432062, 37.807246], [-122.431414, 37.807326], [-122.431787, 37.809057],
[-122.431368, 37.809109], [-122.431027, 37.807395], [-122.430333, 37.807473],
[-122.430652, 37.809196], [-122.430212, 37.809257], [-122.429937, 37.807873],
[-122.428901, 37.808021], [-122.428209, 37.808373], [-122.427017, 37.808129],
[-122.427547, 37.808655], [-122.428028, 37.808617], [-122.428059, 37.808747],
[-122.427568, 37.808804], [-122.426866, 37.80813], [-122.426726, 37.808158],
[-122.426817, 37.809302], [-122.426369, 37.810186], [-122.425473, 37.810746],
[-122.424468, 37.810765], [-122.424322, 37.810603], [-122.424449, 37.81045],
[-122.424801, 37.810665], [-122.425364, 37.810603], [-122.425897, 37.810359],
[-122.426387, 37.809852], [-122.426648, 37.808866], [-122.426352,
37.807528]]]}]}}, {"id": "22", "type": "Feature", "properties": {"id": "94134",
"median_score": 88.0, "neighborhood": "Bayshore", "postal5": "94134"},
"geometry": {"type": "GeometryCollection", "geometries": [{"type": "Polygon",
"coordinates": [[[-122.391527, 37.709674], [-122.393396, 37.708262],
[-122.395354, 37.708369], [-122.407491, 37.708181], [-122.408326, 37.708426],
[-122.410318, 37.708314], [-122.410071, 37.708909], [-122.412248, 37.708601],
[-122.413821, 37.709054], [-122.414165, 37.708316], [-122.418198, 37.708311],
[-122.420305, 37.707817], [-122.420709, 37.708291], [-122.426074, 37.708319],
[-122.426415, 37.707763], [-122.427043, 37.708066], [-122.426985, 37.70837],
[-122.427283, 37.708307], [-122.427347, 37.710021], [-122.426175, 37.710959],
[-122.42923, 37.712203], [-122.433751, 37.713118], [-122.431213, 37.716631],
[-122.42836, 37.716053], [-122.427306, 37.715611], [-122.42605, 37.718375],
[-122.425602, 37.718097], [-122.425093, 37.717356], [-122.424422, 37.717063],
[-122.423615, 37.71719], [-122.422477, 37.717882], [-122.423308, 37.718471],
[-122.423541, 37.718906], [-122.423493, 37.719533], [-122.423682, 37.719801],
[-122.425484, 37.720235], [-122.424405, 37.721949], [-122.425246, 37.722277],
[-122.424475, 37.723544], [-122.422873, 37.723983], [-122.423554, 37.725625],
[-122.424157, 37.725984], [-122.423154, 37.727262], [-122.423712, 37.728761],
[-122.426016, 37.728636], [-122.426026, 37.729391], [-122.423689, 37.729515],
[-122.422911, 37.729289], [-122.420599, 37.729374], [-122.420399, 37.729529],
[-122.420306, 37.729926], [-122.420182, 37.731449], [-122.420949, 37.731304],
[-122.420551, 37.731562], [-122.420701, 37.731936], [-122.416255, 37.732034],
[-122.413909, 37.732606], [-122.409647, 37.734951], [-122.408488, 37.736297],
[-122.408218, 37.73765], [-122.407024, 37.738016], [-122.406506, 37.735623],
[-122.404946, 37.733588], [-122.402107, 37.727849], [-122.401262, 37.724794],
[-122.399202, 37.71992], [-122.398555, 37.717783], [-122.398393, 37.716337],
[-122.398027, 37.715493], [-122.396489, 37.714972], [-122.396527, 37.714716],

```
[-122.394738, 37.713611], [-122.393041, 37.713609], [-122.391224, 37.713259],
[-122.390961, 37.712322], [-122.390439, 37.712109], [-122.388774, 37.71228],
[-122.387994, 37.713004], [-122.387684, 37.712178], [-122.386375, 37.711033],
[-122.386724, 37.711066], [-122.38874, 37.710489], [-122.39094, 37.710318],
[-122.391782, 37.709808], [-122.391527, 37.709674]]]}}]}'
```

[219]:

```
geosource = GeoJSONDataSource(geojson = json_data)
#set the color palette
palette = brewer['Blues'][9]
palette = palette[::-1]
color_mapper = LinearColorMapper(palette = palette, low = 88, high = 100, ␣
 ↪nan_color = '#d9d9d9')
color_bar = ColorBar(color_mapper=color_mapper, label_standoff=8,width = 500,␣
 ↪height = 20,
border_line_color='black',location = (0,0), orientation ='horizontal')
#Set the size and title of the graph
p = figure(title = 'Business Inspection Median Scores', plot_height = 700 ,␣
 ↪plot_width = 700, toolbar_location = None,
           tooltips=[
          ("Name", "@neighborhood"),
          ("Zip Code","@id"),
          ("Median Score", "@median_score")])
#Makes it so there are no grid lines
p.xgrid.grid_line_color = None
p.ygrid.grid_line_color = None
p.patches('xs','ys', source = geosource,fill_color = {'field':'median_score',␣
 ↪'transform' : color_mapper},
         line_color = 'black', line_width = 0.25, fill_alpha = 1)
p.add_layout(color_bar, 'below')
output_notebook()
show(p)
```

[ ]:

```
```

[ ]:

```
```