In [1]:
```
import gmaps
import json
import pandas as pd
import os
import gmaps.geojson_geometries
import gmaps.datasets
from matplotlib.cm import viridis
from matplotlib.colors import to_hex
```

In [2]:
```
gmaps.configure(api_key="")
```

In [3]:
```
public_df = pd.DataFrame(pd.read_csv("SJpublic_color.csv"))
```

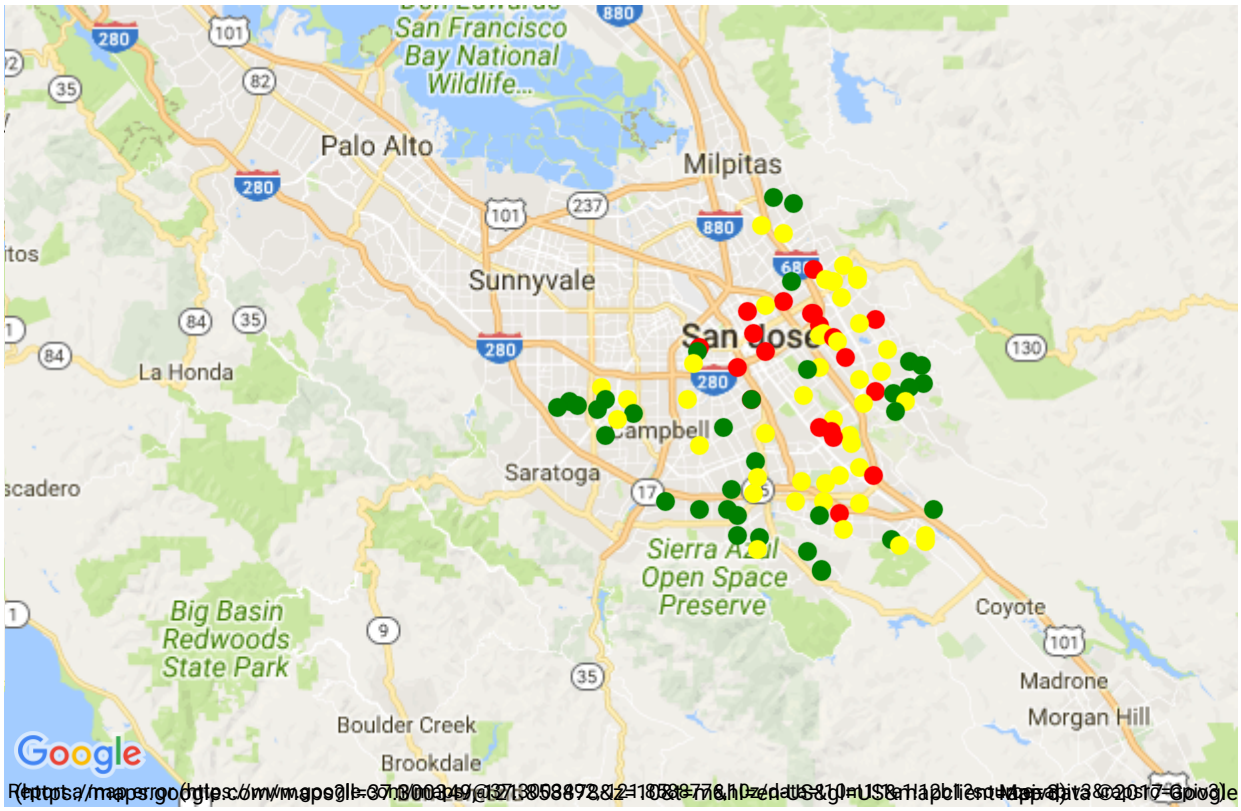
In [4]:
```

plocations = public_df[["lat", "lon"]]
pweights = public_df["gsRating"]
pcolors = []

for rating in public_df["gsRating"]:
    if rating <= 3:
        pcolors.append('red')
    elif rating >3 and rating <7:
        pcolors.append('yellow')
    else:
        pcolors.append('green')

fig = gmaps.figure()
fig.add_layer(gmaps.symbol_layer(plocations, fill_color= pcolors, stroke_co
```

In [5]:
```
fig
```


San Francisco Bay National Wildlife...
Palo Alto
Milpitas
Sunnyvale
San Jose
La Honda
Campbell
Saratoga
Sierra Azul Open Space Preserve
Big Basin Redwoods State Park
Coyote
Madrone
Morgan Hill
Boulder Creek
Brookdale
Google

In [6]:
```
with open("tl_2010_06_zcta5100.geojson") as f:
    geometry = json.load(f)

file_one = os.path.join('Santa_Clara_Home_Appreciation.csv')
file_one_df = pd.read_csv(file_one, encoding="utf-8")
home_apprec_df = file_one_df[['RegionID','Apperciation']]
home_apprec_df.head()

subset = home_apprec_df
rows = [tuple(x) for x in subset.values]
zip_values = dict(rows)
print(zip_values)
```

```
{95123.0: 82.544040269999996, 95035.0: 81.731368070000002, 95051.0: 95.35
4177710000002, 95014.0: 78.438880710000007, 94087.0: 89.630951049999993,
95125.0: 69.936993700000002, 95112.0: 99.796924860000004, 94086.0: 101.71
35325, 95008.0: 77.849427700000007, 95020.0: 71.837464859999997, 95124.0:
78.679793419999996, 95037.0: 60.17942146, 95127.0: 88.089043079999996, 95
136.0: 72.854715380000002, 95111.0: 88.337762029999993, 95050.0: 89.43446
6950000001, 95126.0: 74.757894739999998, 94040.0: 92.433641820000005, 943
03.0: 99.941217960000003, 95129.0: 79.407665510000001, 95128.0: 80.499500
499999996, 95134.0: 87.193656900000008, 94043.0: 112.22721969999999, 9511
6.0: 98.195437519999999, 95120.0: 59.402298850000001, 95122.0: 95.4223081
89999995, 95070.0: 58.658108910000003, 95132.0: 82.477807799999994, 9430
6.0: 105.74790490000001, 95148.0: 74.036885249999997, 95118.0: 73.3570492
39999995, 95032.0: 64.782563350000004, 95117.0: 82.670796760000002, 9512
1.0: 81.891551070000006, 95131.0: 78.724297099999987, 94085.0: 131.794990
7, 95054.0: 84.984604239999996, 94024.0: 75.937559489999998, 94089.0: 12
3.98641000000001, 95133.0: 102.33918129999999, 94301.0: 97.58353567000000
3, 94022.0: 88.048925049999994, 95135.0: 72.170418010000006, 94041.0: 95.
575011829999994, 95110.0: 104.84693879999999, 95138.0: 63.32335329, 9503
0.0: 62.069435919999997, 95130.0: 88.766996289999994, 94305.0: 70.7353467
8, 95033.0: 46.539110860000001, 95119.0: 77.707285060000004, 95046.0: 58.
316666669999996, 95139.0: 82.494783209999994}
```

In [7]:
```
min_zip = min(zip_values.values())
max_zip = max(zip_values.values())
zip_range = max_zip - min_zip

def calculate_color(zipcode):
    normalized_zip = (zipcode - min_zip) / zip_range
#    inverse_zip = 1.0 - normalized_zip
    mpl_color = viridis(normalized_zip)
    gmaps_color = to_hex(mpl_color, keep_alpha=False)

    return gmaps_color
```

In [8]:
```
#print(geometry['features'][0]['properties']['ZCTA5CE10'])

colors = []
for feature in geometry['features']:
    zip_name = feature['properties']['ZCTA5CE10']
    try:
        zip_code = zip_values[int(zip_name)]
        color = calculate_color(zip_code)
    except KeyError:
        color = (0, 0, 0, 0.3)
    colors.append(color)
print(colors)
```

```
[(0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0,
0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.
3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0,
0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0,
0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3),
(0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0,
0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.
3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0,
0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0,
0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3),
(0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0,
0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.
3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0,
0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0,
0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3),
(0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0,
0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.
3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0,
0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0,
0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3), (0, 0, 0, 0.3),
```

In [9]:
```
geojson_layer = gmaps.geojson_layer(geometry, fill_color=colors, stroke_colo
fig.add_layer(geojson_layer)
fig
```