In [213]:
```
import pandas as pd
import geopy
import re
import time
from geopy.geocoders import Nominatim
from geopy.exc import GeocoderTimedOut
from geopy.extra.rate_limiter import RateLimiter
import matplotlib.pyplot as plt
import folium
from folium.plugins import FastMarkerCluster
```

In [214]:
```
locator = Nominatim(user_agent="myGeocoder")
location = locator.geocode("10858 Remmet Ave, Chatsworth, CA 91311")
```

In [215]:
```
print(location.address)
print("Latitude = {}, Longitude = {}".format(location.latitude, location.longi
tude))
```

```
10858, Remmet Avenue, Chatsworth, Los Angeles, Los Angeles County, Californi
a, 91311, United States
Latitude = 34.26812293939393, Longitude = -118.60029364646465
```

In [216]:
```
l=locator.reverse("34.26812293939393,-118.60029364646465").raw['address']['cou
nty']
```

In [217]:
```
l
```

Out[217]:
```
'Los Angeles County'
```

Geocoder Dry Code:

```
Load .csv file/connect to SQL Server
## This part can not be done in SQL Server. This will require Python
        Check for Valid Columns:
        Check for Address, Street Name, City, State, Zipcode
        Must have atleast the Zip
        Where there is no zip, throw an error and also the rowid

        Remove All punctuation and special characters from Street Addresses
        Create a new address field by combining all Address Fields, and add a comma
in between the combined fields
        Create two fields each for Lattitude/Longitude. One for street Geocoding, t
he other for Postal Code Geocoding
        Apply the Geocoding function to each row
```

In [218]:
```
## Location of the SQL FILE
d = 'C:\\Users\\sali\\CSCE5214\\'
# d = 'data/'
df = pd.read_csv('%sGeoAdd.csv'%d)
```

In [220]:
```
# Convert all Column Names to Lower Case and Look for Address Fields
df.columns = map(str.lower, df.columns)

add_fields = ['street', 'streetname', 'city', 'state', 'statecode', 'zip', 'zi
pcode' , 'postalcode' , 'cntrycode']
found = []
not_found = []
for f in add_fields:
    if f in df.columns:
        found.append(f)
    else:
        not_found.append(f)

print ('Fields Found' , found)
print ('Fields not Found', not_found)
```

```
Fields Found ['streetname', 'city', 'statecode', 'postalcode', 'cntrycode']
Fields not Found ['street', 'state', 'zip', 'zipcode']
```

In [221]:
```
# Check and Clean the Fields that were Found
null_fields=[]
## Check for null
for f in found:
    if (df[f].isnull().sum(axis=0)) > 0:
        print (f, df[f].isnull().sum(axis=0))
        null_fields.append ([f, df[f].isnull().sum(axis=0)])
```

In [222]:
```
# Fix the Null fields
for f in null_fields:
    df[f[0]].fillna('No ' + f[0], inplace = True)
```

In [223]:
```
df
```

Out[223]:

	rowid	streetname	city	statecode	postalcode	cntrycode	occtype	bldgclass	numfloc
0	NaN	1500 W Cypress Creek Rd	Fort Lauderdale	FL	33309	US	1	1	
1	NaN	851 Gulf Shore Blvd. North	Naples	FL	34102	US	2	2	
2	NaN	852 Gulf Shore Blvd. North	Naples	FL	34102	US	1	3	
3	NaN	825 South Golf Dr.	Naples	FL	34102	US	1	1	
4	NaN	485 South Golf Drive	Naples	FL	34102	US	2	2	
5	NaN	525 South Flagler Dr.	West Palm Beach	FL	33401	US	2	1	
6	NaN	525 South Flagler Dr	West Palm Beach	FL	33401	US	2	2	

In [224]:
```
#Remove all Special Characters
df['streetname']=df['streetname'].str.translate({ord(c): None for c in '?!#@#$,.;-@!%^&*)('})
df['city']=df['city'].str.translate({ord(c): None for c in '?!@#$,.;-@!#%^&*)('})
df['statecode']=df['statecode'].str.translate({ord(c): None for c in '?#!@#$,.;-@!%^&*)('})
```

In [225]:
```
# Check Nulls
df[found].isnull().sum(axis= 0)
```

Out[225]:
```
streetname    0
city          0
statecode     0
postalcode    0
cntrycode     0
dtype: int64
```

In [226]:
```
#pad postal codes with zero to the left where only 4 digits
df['postalcode']=df['postalcode'].astype(str).str.pad(5, side='left', fillchar='0')
```

In [227]:
```
# if no Nulls then combine the address field
df['address'] = df['streetname'] + ', ' + df['city'] + ', ' + df['statecode'] + ', '+ df['cntrycode'] + ', ' + df['postalcode'].astype(str)
```

In [228]:
```
#check there are no null addresses
df['address'].isnull().sum(axis=0)
```

Out[228]:
```
0
```

In [229]:
```
df
```

Out[229]:

	rowid	streetname	city	statecode	postalcode	cntrycode	occtype	bldgclass	numfloc
0	NaN	1500 W Cypress Creek Rd	Fort Lauderdale	FL	33309	US	1	1	
1	NaN	851 Gulf Shore Blvd North	Naples	FL	34102	US	2	2	
2	NaN	852 Gulf Shore Blvd North	Naples	FL	34102	US	1	3	
3	NaN	825 South Golf Dr	Naples	FL	34102	US	1	1	
4	NaN	485 South Golf Drive	Naples	FL	34102	US	2	2	
5	NaN	525 South Flagler Dr	West Palm Beach	FL	33401	US	2	1	
6	NaN	525 South Flagler Dr	West Palm Beach	FL	33401	US	2	2	

In [230]:
```
df['address']=df['address'].fillna(df['postalcode'].astype(str) + ', ' + df['cntrycode'])
```

In [74]:
```
df.to_csv('Test.csv')
```

In [231]:
```
df['address'].head()
```

Out[231]:
```
0    1500 W Cypress Creek Rd, Fort Lauderdale, FL, ...
1     851 Gulf Shore Blvd North, Naples, FL, US, 34102
2     852 Gulf Shore Blvd North, Naples, FL, US, 34102
3              825 South Golf Dr, Naples, FL, US, 34102
4          485 South Golf Drive, Naples, FL, US, 34102
Name: address, dtype: object
```

In [232]:
```
def gcode (address, postalcode, lat):
    if (lat=='Yes'):
        if locator.geocode(address) is None:
            return locator.geocode(postalcode).point[0]
        else:
            return locator.geocode(address).point[0]
    else:
        if locator.geocode(address) is None:
            return locator.geocode(postalcode).point[1]
        else:
            return locator.geocode(address).point[1]


def do_geocode(address, postalcode):
    try:
        if locator.geocode(address) is None:
            return ('Zip', locator.geocode(postalcode).latitude, locator.geoco
de(postalcode).longitude)
        else:
            return ('Street', locator.geocode(address).latitude, locator.geoco
de(address).longitude)

    except GeocoderTimedOut:
        return do_geocode(address, postalcode)
    except GeocoderQuotaExceeded:
        time.sleep(15)
        return do_geocode(address, postalcode)
```

In [ ]:
```
```

In [243]:
```
def return_gran(address, postalcode, point):
    z=do_geocode (address, postalcode)
    if (point==0):
        return (z[0])
    if (point==1):
        return(z[1])
    if (point==2):
        return (z[2])
    if (point==3):
        return ((locator.reverse(str(z[1]) +" , " + str(z[2])).raw['address'][
'county']).replace(' County', ''))
```

In [ ]:
```
```

In [244]:
```
df['Code_Level']= df.apply(lambda row: return_gran (row['address'], row['postalcode'], 0), axis= 1)
df['latP']= df.apply(lambda row: return_gran (row['address'], row['postalcode'], 1), axis= 1)
df['lngP']= df.apply(lambda row: return_gran (row['address'], row['postalcode'], 2), axis= 1)
df['county']= df.apply(lambda row: return_gran (row['address'], row['postalcode'], 3), axis= 1)
```

In [245]:
```
df
```

Out[245]:

	rowid	streetname	city	statecode	postalcode	cntrycode	occtype	bldgclass	numfloc
0	NaN	1500 W Cypress Creek Rd	Fort Lauderdale	FL	33309	US	1	1	
1	NaN	851 Gulf Shore Blvd North	Naples	FL	34102	US	2	2	
2	NaN	852 Gulf Shore Blvd North	Naples	FL	34102	US	1	3	
3	NaN	825 South Golf Dr	Naples	FL	34102	US	1	1	
4	NaN	485 South Golf Drive	Naples	FL	34102	US	2	2	
5	NaN	525 South Flagler Dr	West Palm Beach	FL	33401	US	2	1	
6	NaN	525 South Flagler Dr	West Palm Beach	FL	33401	US	2	2	

In [246]:
```
df.to_csv('%sGeoAdd_Coded.csv'%d)
```

In [ ]:

In [118]:

In [ ]: