

battle2-part22222

August 3, 2019

0.0.1 New York

Download and Explore Dataset Neighborhood has a total of 5 boroughs and 306 neighborhoods. In order to segment the neighborhoods and explore them, we will essentially need a dataset that contains the 5 boroughs and the neighborhoods that exist in each borough as well as the latitude and longitude coordinates of each neighborhood.

The link to the dataset: https://geo.nyu.edu/catalog/nyu_2451_34572

```
In [185]: # load the data
          !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
          print('Data downloaded!')
```

Data downloaded!

```
In [186]: with open('newyork_data.json') as json_data:
          newyork_data = json.load(json_data)
```

```
In [187]: neighborhoods_data = newyork_data['features']
```

```
In [188]: # define the dataframe columns
          column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

          # instantiate the dataframe
          neighborhoods = pd.DataFrame(columns=column_names)
```

```
In [189]: neighborhoods
```

```
Out[189]: Empty DataFrame
          Columns: [Borough, Neighborhood, Latitude, Longitude]
          Index: []
```

```
In [190]: # let's loop through the data and fill the dataframe one row at a time.
          for data in neighborhoods_data:
              borough = neighborhood_name = data['properties']['borough']
              neighborhood_name = data['properties']['name']

              neighborhood_latlon = data['geometry']['coordinates']
              neighborhood_lat = neighborhood_latlon[1]
```

```

neighborhood_lon = neighborhood_latlon[0]

neighborhoods = neighborhoods.append({'Borough': borough,
                                      'Neighborhood': neighborhood_name,
                                      'Latitude': neighborhood_lat,
                                      'Longitude': neighborhood_lon}, ignore_index=True)

In [191]: neighborhoods.head()

Out[191]:
   Borough Neighborhood  Latitude  Longitude
0   Bronx   Wakefield   40.894705 -73.847201
1   Bronx   Co-op City   40.874294 -73.829939
2   Bronx   Eastchester  40.887556 -73.827806
3   Bronx   Fieldston   40.895437 -73.905643
4   Bronx   Riverdale   40.890834 -73.912585

In [192]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(
            len(neighborhoods['Borough'].unique()),
            neighborhoods.shape[0])
          )

```

The dataframe has 5 boroughs and 306 neighborhoods.

Use geopy library to get the latitude and longitude values of New York City. In order to define an instance of the geocoder, we need to define a user_agent. We will name our agent ny_explorer, as shown below.

```

In [193]: address = 'New York City, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of New York City are {}, {}'.format(latitude, longitude))

```

The geograpical coordinate of New York City are 40.7127281, -74.0060152.

Create a map of New York with neighborhoods superimposed on top.

```

In [194]: # create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoods['Longitude'],
                                           neighborhoods['Borough'], neighborhoods['Neighborhood']):
    label = '{} {}, {}'.format(neighborhood, borough, lat)
    popup = folium.Popup(label, parse_html=True)
    marker = folium.Marker([lat, lng], popup=popup)
    marker.add_to(map_newyork)

```

```

folium.CircleMarker(
    [lat, lng],
    radius=5,
    popup=label,
    color='blue',
    fill=True,
    fill_color='#3186cc',
    fill_opacity=0.7,
    parse_html=False).add_to(map_newyork)

```

```
map_newyork
```

```
Out[194]: <folium.folium.Map at 0x7f1b744dcb38>
```

0.1 Methodology

0.1.1 Data Exploration

Create a new dataframe of the borough Kensington and Chelsea.

```

In [195]: new_data = df_london_allpart[df_london_allpart['borough'] == 'Kensington and Chelsea']
new_data = new_data.drop_duplicates()
new_data.head()

```

```

Out[195]:
   neighborhoods Dialcode borough posttown postcode \
0      Brompton      020 Kensington and Chelsea  LONDON      SW3
1      Chelsea      020 Kensington and Chelsea  LONDON      SW3
2  Earls Court      020 Kensington and Chelsea  LONDON      SW5
3      Kensington      020 Kensington and Chelsea  LONDON      SW7
4 South Kensington      020 Kensington and Chelsea  LONDON      SW7

   Latitude  Longitude
0  51.49014  -0.16248
1  51.49014  -0.16248
2  51.49004  -0.18971
3  51.49807  -0.17404
4  51.49807  -0.17404

```

```

In [196]: address = 'Kensington and Chelsea, uk'
#address = 'City of London, uk'
#address = 'Islington, uk'

geolocator = Nominatim(user_agent="uk_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of London, uk {}, {}'.format(latitude, longitude))

```

The geograpical coordinate of London, uk 51.4989948, -0.1991229.

```
In [197]: new_data.head()
```

```
Out[197]:
```

	neighborhoods	Dialcode	borough	posttown	postcode	\
0	Brompton	020	Kensington and Chelsea	LONDON	SW3	
1	Chelsea	020	Kensington and Chelsea	LONDON	SW3	
2	Earls Court	020	Kensington and Chelsea	LONDON	SW5	
3	Kensington	020	Kensington and Chelsea	LONDON	SW7	
4	South Kensington	020	Kensington and Chelsea	LONDON	SW7	

	Latitude	Longitude
0	51.49014	-0.16248
1	51.49014	-0.16248
2	51.49004	-0.18971
3	51.49807	-0.17404
4	51.49807	-0.17404

```
In [64]: #create map of North York using latitude and longitude values #new_data['neighborhoods']
map_london_borough = folium.Map(location=[latitude, longitude], zoom_start=12)
```

```
# add markers to map
for lat, lng, label in zip(new_data['Latitude'], new_data['Longitude'], new_data['boroughs']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_london_borough)

map_london_borough
```

```
Out[64]: <folium.folium.Map at 0x7f1b72689d30>
```

Use geopy library to get the latitude and longitude values borough Manhattan.

```
In [198]: address = 'Manhattan, usa'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Manhattan are {}, {}'.format(latitude, longitude))
```

The geograpical coordinate of Manhattan are 40.7900869, -73.9598295.

```
In [199]: ny_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].reset_index(drop=True)
ny_data.head()
```

```
Out[199]:
```

	Borough	Neighborhood	Latitude	Longitude
0	Manhattan	Marble Hill	40.876551	-73.910660
1	Manhattan	Chinatown	40.715618	-73.994279
2	Manhattan	Washington Heights	40.851903	-73.936900
3	Manhattan	Inwood	40.867684	-73.921210
4	Manhattan	Hamilton Heights	40.823604	-73.949688

Let's visualizat Manhattan

```
In [67]: # create map of Manhattan using latitude and longitude values
map_nyc_m = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(ny_data['Latitude'], ny_data['Longitude'], ny_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_nyc_m)

map_nyc_m
```

```
Out[67]: <folium.folium.Map at 0x7f1b726524a8>
```