VIEWS2

August 14, 2020

[1]:
```
#first import some python librairies to convert url into a  panda dataframe
import pandas as pd
import numpy as np
import requests
import json
```

[2]:
```
pip install lxml
```

```
Requirement already satisfied: lxml in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (4.5.2)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```
tables = pd.read_html('https://en.wikipedia.org/wiki/
 ↪List_of_postal_codes_of_Canada:_M', header=0)
#Naming the three columns
columns_names = ['Postalcode', 'Borough','Neighbourhood']
```

[4]:
```
type (tables)
```

[4]:
```
list
```

[5]:
```
tables[0].tail
```

[5]:
```
<bound method NDFrame.tail of     Postal Code           Borough  \
0           M1A      Not assigned
1           M2A      Not assigned
2           M3A        North York
3           M4A        North York
4           M5A  Downtown Toronto
..          …               …
175         M5Z      Not assigned
176         M6Z      Not assigned
177         M7Z      Not assigned
178         M8Z         Etobicoke
179         M9Z      Not assigned

                                    Neighbourhood
0                                    Not assigned
```

```
1                                        Not assigned
2                                           Parkwoods
3                                     Victoria Village
4                           Regent Park, Harbourfront
..                                                ...
175                                      Not assigned
176                                      Not assigned
177                                      Not assigned
178  Mimico NW, The Queensway West, South of Bloor,...
179                                      Not assigned

[180 rows x 3 columns]>
```

[6]:
```
# easier to convert the boolean output into string in order to manipulate the␣
 ↪data
for table in tables:
    if(str(np.array_equal(np.array(table.columns),np.
 ↪array(columns_names)))=="True"):
        Toronto_df = pd.DataFrame(table)
    break
```

[7]:
```
#check the data frame to see if i lost data
Toronto_df = pd.DataFrame(table)
print("Shape of Dataframe is - ",Toronto_df.shape)
Toronto_df.head()
```

```
Shape of Dataframe is -  (180, 3)
```

[7]:

	Postal Code	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront

[8]:
```
#geting out cells without assigned borough
Toronto_df = Toronto_df[Toronto_df.Borough!="Not assigned"]
print("Shape of Dataframe is - ",Toronto_df.shape)
Toronto_df.head()
```

```
Shape of Dataframe is -  (103, 3)
```

[8]:

	Postal Code	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights

```
6          M7A  Downtown Toronto  Queen's Park, Ontario Provincial Government
```

[9]:
```
#using a np. where loop to put borough names where column Neighbourhood is NA
```

[10]:
```
Toronto_df['Neighbourhood'] = np.where(Toronto_df['Neighbourhood']=='Not␣
↪assigned',Toronto_df['Borough'],Toronto_df['Neighbourhood'])
Toronto_df.tail(10)
```

[10]:

	Postal Code	Borough	Neighbourhood
151	M8W	Etobicoke	Alderwood, Long Branch
152	M9W	Etobicoke	Northwest, West Humber - Clairville
153	M1X	Scarborough	Upper Rouge
156	M4X	Downtown Toronto	St. James Town, Cabbagetown
157	M5X	Downtown Toronto	First Canadian Place, Underground city
160	M8X	Etobicoke	The Kingsway, Montgomery Road, Old Mill North
165	M4Y	Downtown Toronto	Church and Wellesley
168	M7Y	East Toronto	Business reply mail Processing Centre, South C…
169	M8Y	Etobicoke	Old Mill South, King's Mill Park, Sunnylea, Hu…
178	M8Z	Etobicoke	Mimico NW, The Queensway West, South of Bloor,…

[11]:
```
#combine neighbourhoods that exist for one postal code
```

[12]:
```
Toronto_postal_df = pd.DataFrame(Toronto_df.groupby(['Postal␣
↪Code','Borough'])['Neighbourhood'].apply(','.join).reset_index())
```

[13]:
```
print(Toronto_postal_df.head())
print("\n The Shape of the dataframe is  - ",Toronto_postal_df.shape)
```

```
  Postal Code      Borough                                Neighbourhood
0         M1B  Scarborough                               Malvern, Rouge
1         M1C  Scarborough  Rouge Hill, Port Union, Highland Creek
2         M1E  Scarborough       Guildwood, Morningside, West Hill
3         M1G  Scarborough                                  Woburn
4         M1H  Scarborough                               Cedarbrae
```

```
The Shape of the dataframe is  -  (103, 3)
```

[14]:
```
#the shape still the same ie there was not several neighbourhoods for one postal code
```

[15]:
```
#PartII
```

[16]:
```
!conda install -c conda-forge geocoder --yes
print("Installation Done!")
import geocoder # import geocoder
print("Geo Coder imported!")
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.3
  latest version: 4.8.4

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.

Installation Done!
Geo Coder imported!
```

[17]:
```
def get_geocoder(postal_code_from_df):
    # initialize  variable to None
    lat_lng_coords = None
    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Toronto, Ontario'.format(postal_code_from_df.strip()))
        lat_lng_coords = g.latlng
        latitude = lat_lng_coords[0]
        longitude = lat_lng_coords[1]
    return latitude,longitude
```

[18]:
```
#adding geoloc to each postal code
```

[19]:
```
Toronto_postal_df['Latitude'], Toronto_postal_df['Longitude'] = zip(*Toronto_postal_df['Postal Code'].apply(get_geocoder))
```

```
Toronto_postal_df.head(11)
```

[19]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough	Malvern, Rouge	43.81153	-79.19552
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.78564	-79.15871
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.76575	-79.17520
3	M1G	Scarborough	Woburn	43.76820	-79.21761
4	M1H	Scarborough	Cedarbrae	43.76969	-79.23944
5	M1J	Scarborough	Scarborough Village	43.74309	-79.23526
6	M1K	Scarborough	Kennedy Park, Ionview, East Birchmount Park	43.72861	-79.26367
7	M1L	Scarborough	Golden Mile, Clairlea, Oakridge	43.71406	-79.28412
8	M1M	Scarborough	Cliffside, Cliffcrest, Scarborough Village West	43.72360	-79.23496
9	M1N	Scarborough	Birch Cliff, Cliffside West	43.69539	-79.26194
10	M1P	Scarborough	Dorset Park, Wexford Heights, Scarborough Town…	43.75998	-79.26837

[20]:
```
#need to install geopy in order to use geopy.geocoders
```

[21]:
```
pip install geopy
```

```
Requirement already satisfied: geopy in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (2.0.0)
Requirement already satisfied: geographiclib<2,>=1.49 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geopy)
(1.50)
Note: you may need to restart the kernel to use updated packages.
```

[22]:
```
#needed to import Nominatim for using geolocator
```

[23]:
```
from geopy.geocoders import Nominatim
address = 'Toronto, Ontario'
geolocator = Nominatim(user_agent="toronto_ontario")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
```

```
print('The geograpical coordinates of Toronto, Ontario are {}, {}.'.
 ↪format(latitude, longitude))
```

```
The geograpical coordinates of Toronto, Ontario are 43.6534817, -79.3839347.
```

[24]:
```
#Import Folium for maping Toronto
```

[25]:
```
import folium
Toronto_map = folium.Map(location=[latitude, longitude], zoom_start=11)

for lat, long, post, borough, neigh in zip(Toronto_postal_df['Latitude'],␣
 ↪Toronto_postal_df['Longitude'],Toronto_postal_df['Postal Code'],␣
 ↪Toronto_postal_df['Borough'], Toronto_postal_df['Neighbourhood']):
    label = "{} ({}): {}".format(borough, post, neigh)
    popup = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, long],
        radius=5,
        popup=popup,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(Toronto_map)

Toronto_map
```

[25]:
```
<folium.folium.Map at 0x7f328b9e3c18>
```