Paris

June 26, 2020

1 BATTLE OF NEIGHBOURHOOD - PARIS

1.1 Exploring Paris

1.1.1 Let's import all the necessary libraries

```
[1]: !pip install geocoder
     !pip install bs4
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
     import urllib
     from urllib.request import urlopen
     from bs4 import BeautifulSoup
```

```
Collecting geocoder
  Downloading https://files.pythonhosted.org/packages/4f/6b/13166c909ad2f2
d76b929a4227c952630ebaf0d729f6317eb09cbceccbab/geocoder-1.38.1-py2.py3-none-
any.whl (98kB)
     |                    | 102kB 4.3MB/s ta 0:00:011
Collecting click (from geocoder)
  Downloading https://files.pythonhosted.org/packages/d2/3d/fa76db83bf75c4
f8d338c2fd15c8d33fdd7ad23a9b5e57eb6c5de26b430e/click-7.1.2-py2.py3-none-any.whl
(82kB)
     |                    | 92kB 14.9MB/s eta 0:00:01
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(1.15.0)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(2.23.0)
Collecting ratelim (from geocoder)
  Downloading https://files.pythonhosted.org/packages/f2/98/7e6d147fd16a10a5f821
db6e25f192265d6ecca3d82957a4fdd592cad49c/ratelim-0.1.6-py2.py3-none-any.whl
Collecting future (from geocoder)
  Downloading https://files.pythonhosted.org/packages/45/0b/38b06fd9b92dc2
b68d58b75f900e97884c45bedd2ff83203d933cf5851c9/future-0.18.2.tar.gz (829kB)
```

```
                    |                     | 829kB 19.5MB/s eta 0:00:01
Requirement already satisfied: chardet<4,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (2020.4.5.2)
Requirement already satisfied: idna<3,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (2.9)
Requirement already satisfied: decorator in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ratelim->geocoder) (4.4.2)
Building wheels for collected packages: future
  Building wheel for future (setup.py) … done
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/8b/99/a0/81daf51
dcd359a9377b110a8a886b3895921802d2fc1b2397e
Successfully built future
Installing collected packages: click, ratelim, future, geocoder
Successfully installed click-7.1.2 future-0.18.2 geocoder-1.38.1 ratelim-0.1.6
Collecting bs4
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f45617413
9ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting beautifulsoup4 (from bs4)
  Downloading https://files.pythonhosted.org/packages/66/25/ff030e24372656
16a1e9b25ccc864e0371a0bc3adb7c5a404fd661c6f4f6/beautifulsoup4-4.9.1-py3-none-
any.whl (115kB)
      |                     | 112kB 5.9MB/s eta 0:00:01
|                 | 122kB 5.9MB/s
Collecting soupsieve>1.2 (from beautifulsoup4->bs4)
  Downloading https://files.pythonhosted.org/packages/6f/8f/457f4a5390eeae1cc3ae
ab89deb7724c965be841ffca6cfca9197482e470/soupsieve-2.0.1-py3-none-any.whl
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) … done
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/a0/b0/b2/4f80b94
56b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: soupsieve, beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.9.1 bs4-0.0.1 soupsieve-2.0.1
```

Extract Data from Wikepedia

```python
[2]: url="https://en.wikipedia.org/wiki/Arrondissements_of_Paris"
     page = urllib.request.urlopen(url)
     soup = BeautifulSoup(page)
```

```
all_tables=soup.find_all("table")
right_table=soup.find('table',class_="wikitable sortable")
```

```
[3]:  A=[]
      B=[]
      C=[]
      D=[]


      for row in right_table.findAll('tr'):
          cells=row.findAll('td')
          if len(cells)==8:
              A.append((cells[0].find(text=True)).strip())
              B.append((cells[1].find(text=True)).strip())
              C.append((cells[2].find(text=True)).strip())
              D.append((cells[4].find(text=True)).strip())

      import re
      A2=[]
      subs="("
      for r in A:
          #start=str.find(A[r],"(")
          length=len(r)
          start=str.find(r,"(")
          A2.append(r[0:start-1] + ' Arrondissement')



      #create DF
      df=pd.DataFrame([A2,B,C,D])
      df=df.transpose()
      df.columns=(['Arrondissement','Area','SqKM','Population'])
      df
```

```
[3]:         Arrondissement             Area       SqKM Population
      0      1st Arrondissement          Louvre   1.826 km     17,700
      1      2nd Arrondissement          Bourse   0.992 km     20,700
      2      3rd Arrondissement          Temple   1.171 km     35,100
      3      4th Arrondissement   Hôtel-de-Ville   1.601 km     28,600
      4      5th Arrondissement        Panthéon   2.541 km     60,600
      5      6th Arrondissement      Luxembourg   2.154 km     45,200
      6      7th Arrondissement   Palais-Bourbon   4.088 km     55,400
      7      8th Arrondissement          Élysée   3.881 km     38,700
      8      9th Arrondissement           Opéra   2.179 km     58,500
      9     10th Arrondissement        Entrepôt   2.892 km     88,800
      10    11th Arrondissement      Popincourt   3.666 km    152,500
      11    12th Arrondissement         Reuilly  16.324 km    138,300
      12    13th Arrondissement        Gobelins   7.146 km    181,300
```

```
13   14th Arrondissement              Observatoire    5.621  km     134,700
14   15th Arrondissement                 Vaugirard    8.502  km     232,400
15   16th Arrondissement                     Passy   16.305  km     149,500
16   17th Arrondissement   Batignolles-Monceau        5.669  km     160,300
17   18th Arrondissement       Butte-Montmartre        6.005  km     188,700
18   19th Arrondissement       Buttes-Chaumont         6.786  km     187,200
19   20th Arrondissement          Ménilmontant         5.984  km     191,800
```

1.2   Import Geopy and get Latitude and Longitude for all the Arrondissements / areas

[5]:
```python
!pip install geopy
```

```
Collecting geopy
  Downloading https://files.pythonhosted.org/packages/ab/97/25def417bf5db4
cc6b89b47a56961b893d4ee4fec0c335f5b9476a8ff153/geopy-1.22.0-py2.py3-none-any.whl
(113kB)
     |                        | 122kB 5.4MB/s eta 0:00:01
Collecting geographiclib<2,>=1.49 (from geopy)
  Downloading https://files.pythonhosted.org/packages/8b/62/26ec95a98ba642991631
99e95ad1b0e34ad3f4e176e221c40245f211e425/geographiclib-1.50-py3-none-any.whl
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.50 geopy-1.22.0
```

[6]:
```python
from geopy.geocoders import Nominatim
lat=[]
lon=[]
loc=[]
for r in df.index:
    area=df.iloc[r,0]
    geolocator = Nominatim(user_agent="ny_explorer")
    location = geolocator.geocode(area)
    loc.append(area)
    lat.append(location.latitude)
    lon.append(location.longitude)


df_longlat=pd.DataFrame([loc,lat,lon])
```

[7]:
```python
df_longlat=df_longlat.transpose()
df_longlat.columns=(['Arrondissement','Latitude','Longitude'])
df_paris=pd.merge(df,df_longlat,how='inner')
```

[8]:
```python
df_paris
```

```
[8]:          Arrondissement                  Area        SqKM Population Latitude  \
    0     1st Arrondissement              Louvre    1.826  km     17,700   48.8646
    1     2nd Arrondissement              Bourse    0.992  km     20,700   48.8687
    2     3rd Arrondissement              Temple    1.171  km     35,100   48.8642
    3     4th Arrondissement      Hôtel-de-Ville    1.601  km     28,600   48.8562
    4     5th Arrondissement            Panthéon    2.541  km     60,600   48.8461
    5     6th Arrondissement          Luxembourg    2.154  km     45,200   48.8504
    6     7th Arrondissement      Palais-Bourbon    4.088  km     55,400    48.857
    7     8th Arrondissement              Élysée    3.881  km     38,700    43.243
    8     9th Arrondissement               Opéra    2.179  km     58,500    48.876
    9    10th Arrondissement            Entrepôt    2.892  km     88,800   48.8761
    10   11th Arrondissement          Popincourt    3.666  km    152,500   48.8584
    11   12th Arrondissement             Reuilly   16.324  km    138,300   48.8396
    12   13th Arrondissement            Gobelins    7.146  km    181,300   48.8324
    13   14th Arrondissement         Observatoire    5.621  km    134,700   48.8331
    14   15th Arrondissement           Vaugirard    8.502  km    232,400   48.8414
    15   16th Arrondissement               Passy   16.305  km    149,500   48.8632
    16   17th Arrondissement  Batignolles-Monceau    5.669  km    160,300   48.8842
    17   18th Arrondissement     Butte-Montmartre    6.005  km    188,700     48.89
    18   19th Arrondissement     Buttes-Chaumont    6.786  km    187,200   48.8893
    19   20th Arrondissement        Ménilmontant    5.984  km    191,800    48.865

        Longitude
    0      2.3344
    1     2.34169
    2     2.36094
    3     2.35562
    4     2.34452
    5     2.33295
    6      2.3202
    7     5.37887
    8     2.33996
    9     2.35991
    10     2.3797
    11    2.39575
    12    2.35558
    13    2.32696
    14    2.30038
    15    2.27576
    16    2.32236
    17    2.34647
    18    2.38436
    19    2.39893
```

```
[9]: df_paris.iloc[7,4]=48.86849652
     df_paris.iloc[7,5]=2.30449878
```

1.3   Get Venue Details from FourSquare API

```
[10]: CLIENT_ID = 'HUGNSJUP0RDQJP2MJ3X5IEORXTEAF5HQFCVGSCJFKYYTJVTC' # your␣
      ↪Foursquare ID
      CLIENT_SECRET = 'HRB5OZH12CJNW3UGWOYMFQAKR1M3GE2OFTKDIQEGT2IY1O1K' # your␣
      ↪Foursquare Secret
      VERSION = '20180605' # Foursquare API version

      print('Your credentails:')
      print('CLIENT_ID: ' + CLIENT_ID)
      print('CLIENT_SECRET:' + CLIENT_SECRET)
      radius=2000
      LIMIT=100
      VERSION = '20180605' # Foursquare API version
```

```
Your credentails:
CLIENT_ID: HUGNSJUP0RDQJP2MJ3X5IEORXTEAF5HQFCVGSCJFKYYTJVTC
CLIENT_SECRET:HRB5OZH12CJNW3UGWOYMFQAKR1M3GE2OFTKDIQEGT2IY1O1K
```

```
[180]: import requests
```

```
[11]: import requests
      neighborhood_latitude = [48.8646] # neighborhood latitude value
      neighborhood_longitude = [2.3344] # neighborhood longitude value
      neighborhood_name = ['1st Arrondissement'] # neighborhood name

      print('Latitude and longitude values of {} are {}, {}.'.
       ↪format(neighborhood_name,

                                                                    ␣
       ↪neighborhood_latitude,

                                                                    ␣
       ↪neighborhood_longitude))

      # get all venues near toronto and create a df

      url = 'https://api.foursquare.com/v2/venues/explore?
       ↪client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.
       ↪format(CLIENT_ID, CLIENT_SECRET, neighborhood_latitude,␣
       ↪neighborhood_longitude, VERSION, radius, LIMIT)
      results = requests.get(url).json()
      results
      print(url)
```

```
Latitude and longitude values of ['1st Arrondissement'] are [48.8646], [2.3344].
https://api.foursquare.com/v2/venues/explore?client_id=HUGNSJUP0RDQJP2MJ3X5IEORX
TEAF5HQFCVGSCJFKYYTJVTC&client_secret=HRB5OZH12CJNW3UGWOYMFQAKR1M3GE2OFTKDIQEGT2
IY1O1K&ll=[48.8646],[2.3344]&v=20180605&radius=2000&limit=100
```

```python
[12]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

          venues_list=[]
          for name, lat, lng in zip(names, latitudes, longitudes):
              print(name)

              # create the API request URL
              url = 'https://api.foursquare.com/v2/venues/explore?
      ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
                  CLIENT_ID,
                  CLIENT_SECRET,
                  VERSION,
                  lat,
                  lng,
                  radius,
                  LIMIT)

              # make the GET request
              results = requests.get(url).json()["response"]['groups'][0]['items']

              # return only relevant information for each nearby venue
              venues_list.append([(
                  name,
                  lat,
                  lng,
                  v['venue']['name'],
                  v['venue']['location']['lat'],
                  v['venue']['location']['lng'],
                  v['venue']['categories'][0]['name']) for v in results])

          nearby_venues = pd.DataFrame([item for venue_list in venues_list for item
      ↪in venue_list])
          nearby_venues.columns = ['Arrondissement',
                        'Arrondissement Latitude',
                        'Arrondissement Longitude',
                        'Venue',
                        'Venue Latitude',
                        'Venue Longitude',
                        'Venue Category']

          return(nearby_venues)
```

```python
[13]: Paris_venues = getNearbyVenues(names=df_paris['Arrondissement'],
                                     latitudes=df_paris['Latitude'],
                                     longitudes=df_paris['Longitude']
                                     )
```

```
1st Arrondissement
2nd Arrondissement
3rd Arrondissement
4th Arrondissement
5th Arrondissement
6th Arrondissement
7th Arrondissement
8th Arrondissement
9th Arrondissement
10th Arrondissement
11th Arrondissement
12th Arrondissement
13th Arrondissement
14th Arrondissement
15th Arrondissement
16th Arrondissement
17th Arrondissement
18th Arrondissement
19th Arrondissement
20th Arrondissement
```

[14]: `Paris_venues.shape`

[14]: (1508, 7)

2 Summary of 1508 Venue by Arrondissement

[15]: `Paris_venues.groupby('Arrondissement').count()`

[15]:

Arrondissement	Arrondissement Latitude	Arrondissement Longitude	Venue \
10th Arrondissement	100	100	100
11th Arrondissement	81	81	81
12th Arrondissement	42	42	42
13th Arrondissement	53	53	53
14th Arrondissement	47	47	47
15th Arrondissement	49	49	49
16th Arrondissement	25	25	25
17th Arrondissement	100	100	100
18th Arrondissement	69	69	69
19th Arrondissement	44	44	44
1st Arrondissement	100	100	100
20th Arrondissement	55	55	55
2nd Arrondissement	100	100	100
3rd Arrondissement	100	100	100
4th Arrondissement	100	100	100

```
5th Arrondissement                           100                    100     100
6th Arrondissement                            61                     61      61
7th Arrondissement                            82                     82      82
8th Arrondissement                           100                    100     100
9th Arrondissement                           100                    100     100


                        Venue Latitude   Venue Longitude   Venue Category
Arrondissement
10th Arrondissement               100               100              100
11th Arrondissement                81                81               81
12th Arrondissement                42                42               42
13th Arrondissement                53                53               53
14th Arrondissement                47                47               47
15th Arrondissement                49                49               49
16th Arrondissement                25                25               25
17th Arrondissement               100               100              100
18th Arrondissement                69                69               69
19th Arrondissement                44                44               44
1st Arrondissement                100               100              100
20th Arrondissement                55                55               55
2nd Arrondissement                100               100              100
3rd Arrondissement                100               100              100
4th Arrondissement                100               100              100
5th Arrondissement                100               100              100
6th Arrondissement                 61                61               61
7th Arrondissement                 82                82               82
8th Arrondissement                100               100              100
9th Arrondissement                100               100              100
```

```python
[16]:  # one hot encoding
       paris_onehot = pd.get_dummies(Paris_venues[['Venue Category']], prefix="",␣
        →prefix_sep="")

       # add neighborhood column back to dataframe
       paris_onehot['Arrondissement'] = Paris_venues['Arrondissement']
```

2.1 Take the mean of all the arrondissements by venue

```python
[17]:  paris_grouped = paris_onehot.groupby('Arrondissement').mean().reset_index()
       paris_grouped
```

```
[17]:         Arrondissement   Accessories Store   Afghan Restaurant  \
       0   10th Arrondissement              0.00            0.000000
       1   11th Arrondissement              0.00            0.012346
       2   12th Arrondissement              0.00            0.000000
       3   13th Arrondissement              0.00            0.000000
```

```
4   14th Arrondissement                 0.00           0.000000
5   15th Arrondissement                 0.00           0.000000
6   16th Arrondissement                 0.00           0.000000
7   17th Arrondissement                 0.00           0.010000
8   18th Arrondissement                 0.00           0.000000
9   19th Arrondissement                 0.00           0.000000
10   1st Arrondissement                 0.00           0.000000
11  20th Arrondissement                 0.00           0.000000
12   2nd Arrondissement                 0.00           0.000000
13   3rd Arrondissement                 0.00           0.000000
14   4th Arrondissement                 0.00           0.000000
15   5th Arrondissement                 0.00           0.000000
16   6th Arrondissement                 0.00           0.000000
17   7th Arrondissement                 0.00           0.000000
18   8th Arrondissement                 0.02           0.000000
19   9th Arrondissement                 0.00           0.000000

    African Restaurant  American Restaurant  Antique Shop  \
0            0.020000             0.000000          0.00
1            0.012346             0.000000          0.00
2            0.000000             0.000000          0.00
3            0.000000             0.018868          0.00
4            0.000000             0.000000          0.00
5            0.000000             0.000000          0.00
6            0.000000             0.000000          0.00
7            0.000000             0.000000          0.00
8            0.000000             0.000000          0.00
9            0.000000             0.022727          0.00
10           0.000000             0.000000          0.00
11           0.000000             0.000000          0.00
12           0.000000             0.000000          0.01
13           0.000000             0.000000          0.00
14           0.000000             0.000000          0.00
15           0.000000             0.000000          0.00
16           0.000000             0.016393          0.00
17           0.000000             0.000000          0.00
18           0.000000             0.000000          0.00
19           0.010000             0.000000          0.01

    Argentinian Restaurant  Art Gallery  Art Museum  Arts & Crafts Store  … \
0                 0.000000     0.000000    0.000000             0.000000  …
1                 0.000000     0.000000    0.012346             0.000000  …
2                 0.000000     0.000000    0.000000             0.000000  …
3                 0.000000     0.000000    0.000000             0.000000  …
4                 0.000000     0.000000    0.000000             0.000000  …
5                 0.000000     0.000000    0.000000             0.020408  …
6                 0.000000     0.000000    0.000000             0.000000  …
```

7	0.000000	0.010000	0.000000	0.000000	…
8	0.014493	0.014493	0.000000	0.000000	…
9	0.000000	0.000000	0.000000	0.000000	…
10	0.000000	0.000000	0.010000	0.000000	…
11	0.000000	0.000000	0.000000	0.000000	…
12	0.000000	0.000000	0.000000	0.000000	…
13	0.010000	0.030000	0.010000	0.010000	…
14	0.000000	0.030000	0.010000	0.010000	…
15	0.000000	0.000000	0.000000	0.000000	…
16	0.000000	0.016393	0.016393	0.000000	…
17	0.000000	0.000000	0.036585	0.000000	…
18	0.000000	0.010000	0.010000	0.000000	…
19	0.010000	0.000000	0.000000	0.000000	…

	Turkish Restaurant	Udon Restaurant	Vegetarian / Vegan Restaurant	\
0	0.00000	0.00	0.010000	
1	0.00000	0.00	0.024691	
2	0.02381	0.00	0.000000	
3	0.00000	0.00	0.018868	
4	0.00000	0.00	0.021277	
5	0.00000	0.00	0.000000	
6	0.00000	0.00	0.000000	
7	0.01000	0.00	0.010000	
8	0.00000	0.00	0.014493	
9	0.00000	0.00	0.000000	
10	0.00000	0.03	0.000000	
11	0.00000	0.00	0.000000	
12	0.00000	0.01	0.000000	
13	0.00000	0.00	0.020000	
14	0.00000	0.00	0.000000	
15	0.00000	0.00	0.000000	
16	0.00000	0.00	0.000000	
17	0.00000	0.00	0.012195	
18	0.00000	0.00	0.000000	
19	0.01000	0.00	0.030000	

	Venezuelan Restaurant	Video Game Store	Vietnamese Restaurant	Wine Bar	\
0	0.00	0.000000	0.000000	0.010000	
1	0.00	0.012346	0.012346	0.024691	
2	0.00	0.000000	0.000000	0.023810	
3	0.00	0.000000	0.037736	0.000000	
4	0.00	0.000000	0.021277	0.021277	
5	0.00	0.000000	0.020408	0.000000	
6	0.00	0.000000	0.000000	0.000000	
7	0.00	0.000000	0.020000	0.070000	
8	0.00	0.000000	0.000000	0.014493	
9	0.00	0.000000	0.000000	0.000000	

10	0.00	0.000000	0.000000	0.030000
11	0.00	0.000000	0.000000	0.018182
12	0.00	0.000000	0.000000	0.060000
13	0.00	0.000000	0.030000	0.050000
14	0.00	0.000000	0.000000	0.030000
15	0.00	0.000000	0.010000	0.020000
16	0.00	0.000000	0.000000	0.032787
17	0.00	0.000000	0.000000	0.000000
18	0.00	0.000000	0.000000	0.000000
19	0.01	0.000000	0.010000	0.040000

	Wine Shop	Women's Store	Yoga Studio
0	0.020000	0.000000	0.00
1	0.000000	0.012346	0.00
2	0.000000	0.000000	0.00
3	0.000000	0.000000	0.00
4	0.021277	0.000000	0.00
5	0.000000	0.000000	0.00
6	0.000000	0.000000	0.00
7	0.000000	0.000000	0.00
8	0.000000	0.000000	0.00
9	0.000000	0.000000	0.00
10	0.000000	0.010000	0.00
11	0.000000	0.000000	0.00
12	0.010000	0.010000	0.00
13	0.020000	0.000000	0.00
14	0.000000	0.000000	0.00
15	0.000000	0.000000	0.00
16	0.000000	0.000000	0.00
17	0.000000	0.000000	0.00
18	0.000000	0.010000	0.00
19	0.000000	0.000000	0.01

[20 rows x 206 columns]

2.2 Create a list of top 10 most visited venues in Paris by Arondissement

```
[18]: def return_most_common_venues(row, num_top_venues):
          row_categories = row.iloc[1:]
          row_categories_sorted = row_categories.sort_values(ascending=False)

          return row_categories_sorted.index.values[0:num_top_venues]
```

```
[20]: num_top_venues = 10

      indicators = ['st', 'nd', 'rd']
```

```python
# create columns according to number of top venues
columns = ['Arrondissement']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Arrondissement'] = paris_grouped['Arrondissement']

for ind in np.arange(paris_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
 ↪return_most_common_venues(paris_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head(20)
```

[20]: | | Arrondissement | 1st Most Common Venue | 2nd Most Common Venue | \ |
|---|---|---|---|---|
| 0 | 10th Arrondissement | French Restaurant | Bistro | |
| 1 | 11th Arrondissement | French Restaurant | Bar | |
| 2 | 12th Arrondissement | Hotel | French Restaurant | |
| 3 | 13th Arrondissement | Hotel | French Restaurant | |
| 4 | 14th Arrondissement | French Restaurant | Italian Restaurant | |
| 5 | 15th Arrondissement | French Restaurant | Italian Restaurant | |
| 6 | 16th Arrondissement | Bakery | French Restaurant | |
| 7 | 17th Arrondissement | French Restaurant | Bar | |
| 8 | 18th Arrondissement | Bar | French Restaurant | |
| 9 | 19th Arrondissement | Bistro | Bar | |
| 10 | 1st Arrondissement | Japanese Restaurant | French Restaurant | |
| 11 | 20th Arrondissement | Bar | French Restaurant | |
| 12 | 2nd Arrondissement | French Restaurant | Japanese Restaurant | |
| 13 | 3rd Arrondissement | French Restaurant | Japanese Restaurant | |
| 14 | 4th Arrondissement | French Restaurant | Clothing Store | |
| 15 | 5th Arrondissement | French Restaurant | Hotel | |
| 16 | 6th Arrondissement | French Restaurant | Italian Restaurant | |
| 17 | 7th Arrondissement | French Restaurant | Hotel | |
| 18 | 8th Arrondissement | French Restaurant | Hotel | |
| 19 | 9th Arrondissement | French Restaurant | Hotel | |

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	\
0	Hotel	Coffee Shop	Café	
1	Café	Cocktail Bar	Restaurant	
2	Supermarket	Bistro	Bakery	
3	Thai Restaurant	Indian Restaurant	Bar	
4	Bakery	Hotel	Bar	
5	Hotel	Coffee Shop	Korean Restaurant	

6	Restaurant	Hotel	Café
7	Wine Bar	Italian Restaurant	Hotel
8	Italian Restaurant	Middle Eastern Restaurant	Bistro
9	Café	Beer Bar	French Restaurant
10	Hotel	Coffee Shop	Italian Restaurant
11	Bakery	Japanese Restaurant	Café
12	Wine Bar	Hotel	Bakery
13	Wine Bar	Italian Restaurant	Cocktail Bar
14	Cosmetics Shop	Ice Cream Shop	Gourmet Shop
15	Bar	Italian Restaurant	Pub
16	Bakery	Seafood Restaurant	Café
17	Café	Bakery	Italian Restaurant
18	Boutique	Japanese Restaurant	Tailor Shop
19	Wine Bar	Bar	Tea Room

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue \
0	Pizza Place	Bar	Indian Restaurant
1	Bistro	Pastry Shop	Bakery
2	Café	Sushi Restaurant	Ice Cream Shop
3	Bakery	Gym	Italian Restaurant
4	Plaza	Bistro	Sushi Restaurant
5	Japanese Restaurant	Lebanese Restaurant	Bar
6	Garden	Diner	Sandwich Place
7	Thai Restaurant	Japanese Restaurant	Creperie
8	Pizza Place	Sandwich Place	Plaza
9	Italian Restaurant	Brewery	Gastropub
10	Café	Plaza	Udon Restaurant
11	Bistro	Italian Restaurant	Park
12	Cocktail Bar	Bistro	Clothing Store
13	Restaurant	Hotel	Vietnamese Restaurant
14	Gay Bar	Art Gallery	Wine Bar
15	Ice Cream Shop	Indie Movie Theater	Plaza
16	Plaza	Fountain	Chocolate Shop
17	Coffee Shop	Art Museum	Chocolate Shop
18	Clothing Store	Cosmetics Shop	Plaza
19	Pizza Place	Bakery	Vegetarian / Vegan Restaurant

	9th Most Common Venue	10th Most Common Venue
0	Japanese Restaurant	Restaurant
1	Italian Restaurant	Wine Bar
2	Bookstore	Farmers Market
3	Vietnamese Restaurant	Juice Bar
4	Pizza Place	Restaurant
5	Supermarket	Moroccan Restaurant
6	Chinese Restaurant	Supermarket
7	Pizza Place	Pub
8	Coffee Shop	Café

```
9                    Hotel             Concert Hall
10         Cosmetics Shop                 Wine Bar
11                  Plaza              Supermarket
12              Bookstore        Italian Restaurant
13            Coffee Shop              Art Gallery
14            Coffee Shop               Restaurant
15                   Café                 Wine Bar
16                 Bistro                 Wine Bar
17                 Bistro               Restaurant
18                    Spa        Italian Restaurant
19     Japanese Restaurant                   Bistro
```

2.3 Clustering using KMeans. 4 Clusters

```
[21]: # import k-means from clustering stage
      from sklearn.cluster import KMeans

      # set number of clusters
      kclusters = 4

      paris_grouped_clustering = paris_grouped.drop('Arrondissement', 1)

      # run k-means clustering
      kmeans = KMeans(n_clusters=kclusters, random_state=0).
       →fit(paris_grouped_clustering)

      # check cluster labels generated for each row in the dataframe
      kmeans.labels_[0:10]
```

```
[21]: array([0, 3, 2, 3, 1, 1, 2, 3, 3, 3], dtype=int32)
```

```
[22]: # add clustering labels
      neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

      paris_merged = df_paris

      # merge toronto_grouped with toronto_data to add latitude/longitude for each␣
       →neighborhood
      paris_merged = paris_merged.join(neighborhoods_venues_sorted.
       →set_index('Arrondissement'), on='Arrondissement')
      #paris_merged.head(10) # check the last columns!
      paris_merged.shape
```

```
[22]: (20, 17)
```

```
[23]:  import numpy as np # library to handle data in a vectorized manner

       import pandas as pd # library for data analsysis
       pd.set_option('display.max_columns', None)
       pd.set_option('display.max_rows', None)

       import json # library to handle JSON files

       #!conda install -c conda-forge geopy --yes # uncomment this line if you haven't␣
        ↪completed the Foursquare API lab
       from geopy.geocoders import Nominatim # convert an address into latitude and␣
        ↪longitude values

       import requests # library to handle requests
       from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
        ↪dataframe

       # Matplotlib and associated plotting modules
       import matplotlib.cm as cm
       import matplotlib.colors as colors

       # import k-means from clustering stage
       from sklearn.cluster import KMeans

       #!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you␣
        ↪haven't completed the Foursquare API lab
       import folium # map rendering library

       print('Libraries imported.')
```

```
Libraries imported.
```

2.4   View the clusters on Choropleth Map. use Folium library.

```
[25]:  import folium

       # create map
       map_clusters = folium.Map(location=[lat,lon],zoom_start=12)

       # set color scheme for the clusters
       x = np.arange(kclusters)
       ys = [i + x + (i*x)**2 for i in range(kclusters)]
       colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
       rainbow = [colors.rgb2hex(i) for i in colors_array]

       # add markers to the map
       markers_colors = []
```

```python
for lat, lon, poi, cluster in zip(paris_merged['Latitude'],
 →paris_merged['Longitude'], paris_merged['Arrondissement'],
 →paris_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=10,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

[25]: <folium.folium.Map at 0x7f7a4c007f60>

[26]:
```python
paris_merged.groupby('Cluster Labels').count()
```

[26]:

Cluster Labels	Arrondissement	Area	SqKM	Population	Latitude	Longitude
0	7	7	7	7	7	7
1	5	5	5	5	5	5
2	2	2	2	2	2	2
3	6	6	6	6	6	6

Cluster Labels	1st Most Common Venue	2nd Most Common Venue
0	7	7
1	5	5
2	2	2
3	6	6

Cluster Labels	3rd Most Common Venue	4th Most Common Venue
0	7	7
1	5	5
2	2	2
3	6	6

Cluster Labels	5th Most Common Venue	6th Most Common Venue
0	7	7
1	5	5
2	2	2
3	6	6

```
          7th Most Common Venue  8th Most Common Venue  \
Cluster Labels
0                             7                      7
1                             5                      5
2                             2                      2
3                             6                      6

          9th Most Common Venue  10th Most Common Venue
Cluster Labels
0                             7                       7
1                             5                       5
2                             2                       2
3                             6                       6
```

3   That's all folks!!

```
[ ]:
```