# 205229118

# Mahalakshmi S

### Lab 9. Geocoding the LinkedIn Connections and visualising it on a map

In this lab you need to access your LinkedIn Connections and geocode the companies of the connections and visualise it on a map

### 1. Retrieve the Connections data of your LinkedIn Profile from the previous lab and read it as a csv file using pandas module

In [1]:

```
import os
import csv

# Point this to your 'Connections.csv' file.
CSV_FILE = os.path.join('Connections1.csv')

csvReader = csv.DictReader(open(CSV_FILE), delimiter=',', quotechar='"')
contacts = [row for row in csvReader]
```

In [2]:

```
contacts
```

Out[2]:

```
[{'First Name': 'Micro Media',
  'Last Name': 'Films',
  'Email Address': '',
  'Company': 'Micro Media Films',
  'Position': 'Owner',
  'Connected On': '16-Sep-21'},
 {'First Name': 'JENIFER',
  'Last Name': 'T',
  'Email Address': '',
  'Company': 'TechLife Global',
  'Position': '.NET Developer',
  'Connected On': '16-Sep-21'},
 {'First Name': 'Bhattacharya',
  'Last Name': 'Shreya',
  'Email Address': '',
  'Company': 'The Sparks Foundation',
  'Position': 'Human Resources Executive',
  'Connected On': '16-Sep-21'},
```

### 2.Apply transformation to the Company feature by removing the company related suffixes

In [3]:

```
!pip install prettytable
```

```
Requirement already satisfied: prettytable in c:\users\mahalakshmi\anaconda3
\lib\site-packages (2.2.0)
Requirement already satisfied: wcwidth in c:\users\mahalakshmi\anaconda3\lib
\site-packages (from prettytable) (0.2.5)
```

In [4]:

```
from prettytable import PrettyTable # pip install prettytable
from collections import Counter
from operator import itemgetter

# Define a set of transforms that converts the first item
# to the second item. Here, we're simply handling some
# commonly known abbreviations, stripping off common suffixes,
# etc.

transforms = [(', Inc.', ''), (', Inc', ''), (', LLC', ''), (', LLP', ''),
               (' LLC', ''), (' Inc.', ''), (' Inc', '')]

companies = [c['Company'].strip() for c in contacts if c['Company'].strip() != '']

for i, _ in enumerate(companies):
    for transform in transforms:
        companies[i] = companies[i].replace(*transform)

pt = PrettyTable(field_names=['Company', 'Freq'])
pt.align = 'l'
c = Counter(companies)

[pt.add_row([company, freq]) for (company, freq) in sorted(c.items(), key=itemgetter(1), re

print(pt)
```

```
+---------------------------------------------------+------+
| Company                                           | Freq |
+---------------------------------------------------+------+
| The Sparks Foundation                             | 11   |
| Bishop Heber College, Tiruchirappalli - 620 017.  | 4    |
+---------------------------------------------------+------+
```

In [5]:

```
transforms = [
    ('Sr.', 'Senior'),
    ('Sr', 'Senior'),
    ('Jr.', 'Junior'),
    ('Jr', 'Junior'),
    ('CEO', 'Chief Executive Officer'),
    ('COO', 'Chief Operating Officer'),
    ('CTO', 'Chief Technology Officer'),
    ('CFO', 'Chief Finance Officer'),
    ('VP', 'Vice President'),
    ]

# Read in a list of titles and split apart
# any combined titles like "President/CEO."
# Other variations could be handled as well, such
# as "President & CEO", "President and CEO", etc.

titles = []
for contact in contacts:
    titles.extend([t.strip() for t in contact['Position'].split('/')
                  if contact['Position'].strip() != ''])

# Replace common/known abbreviations

for i, _ in enumerate(titles):
    for transform in transforms:
        titles[i] = titles[i].replace(*transform)

# Print out a table of titles sorted by frequency

pt = PrettyTable(field_names=['Job Title', 'Freq'])
pt.align = 'l'
c = Counter(titles)
[pt.add_row([title, freq])
 for (title, freq) in sorted(c.items(), key=itemgetter(1), reverse=True)
     if freq > 1]
print(pt)

# Print out a table of tokens sorted by frequency

tokens = []
for title in titles:
    tokens.extend([t.strip(',') for t in title.split()])
pt = PrettyTable(field_names=['Token', 'Freq'])
pt.align = 'l'
c = Counter(tokens)
[pt.add_row([token, freq])
 for (token, freq) in sorted(c.items(), key=itemgetter(1), reverse=True)
     if freq > 1 and len(token) > 2]
print(pt)
```

```
+---------------------------------------------+------+
| Job Title                                   | Freq |
+---------------------------------------------+------+
| Data Science and Business Analytics Intern  | 2    |
| Postgraduate Student                        | 2    |
| Data Science and Business Analystics Intern | 2    |
+---------------------------------------------+------+
+--------------+------+
| Token        | Freq |
+--------------+------+
| Data         | 16   |
| Science      | 12   |
| and          | 10   |
| Business     | 8    |
| Intern       | 6    |
| Analytics    | 5    |
| Student      | 3    |
| analytics    | 3    |
| intern       | 3    |
| Human        | 2    |
| Resources    | 2    |
| Executive    | 2    |
| Trainee      | 2    |
| Postgraduate | 2    |
| business     | 2    |
| Analystics   | 2    |
+--------------+------+
```

## 3. Geocode the companies and get the latitude longitude values of the organisations using opencage module

In [6]:

```
!pip install opencage
```

```
Requirement already satisfied: opencage in c:\users\mahalakshmi\anaconda3\li
b\site-packages (2.0.0)
Requirement already satisfied: backoff>=1.10.0 in c:\users\mahalakshmi\anaco
nda3\lib\site-packages (from opencage) (1.11.1)
Requirement already satisfied: Requests>=2.2.0 in c:\users\mahalakshmi\anaco
nda3\lib\site-packages (from opencage) (2.25.1)
Requirement already satisfied: pyopenssl>=0.15.1 in c:\users\mahalakshmi\ana
conda3\lib\site-packages (from opencage) (20.0.1)
Requirement already satisfied: six>=1.5.2 in c:\users\mahalakshmi\anaconda3
\lib\site-packages (from pyopenssl>=0.15.1->opencage) (1.15.0)
Requirement already satisfied: cryptography>=3.2 in c:\users\mahalakshmi\ana
conda3\lib\site-packages (from pyopenssl>=0.15.1->opencage) (3.4.7)
Requirement already satisfied: cffi>=1.12 in c:\users\mahalakshmi\anaconda3
\lib\site-packages (from cryptography>=3.2->pyopenssl>=0.15.1->opencage) (1.
14.5)
Requirement already satisfied: pycparser in c:\users\mahalakshmi\anaconda3\l
ib\site-packages (from cffi>=1.12->cryptography>=3.2->pyopenssl>=0.15.1->ope
ncage) (2.20)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\mahalakshmi
\anaconda3\lib\site-packages (from Requests>=2.2.0->opencage) (1.26.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\mahalakshmi\anaconda
3\lib\site-packages (from Requests>=2.2.0->opencage) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mahalakshmi\an
aconda3\lib\site-packages (from Requests>=2.2.0->opencage) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\mahalakshmi\ana
conda3\lib\site-packages (from Requests>=2.2.0->opencage) (4.0.0)
```

In [7]:

```python
from opencage.geocoder import OpenCageGeocode #pip install opencage
key = '081afd6c6215402cb8625e6a3c722f74'
geocoder = OpenCageGeocode(key)
query = "O'Reilly Media"
result = geocoder.geocode(query)
print(result)
```

```
[{'annotations': {'DMS': {'lat': "39° 55' 0.40800'' N", 'lng': "75° 23' 15.6
8400'' W"}, 'FIPS': {'county': '42045', 'state': '42'}, 'MGRS': '18SVK668661
8592', 'Maidenhead': 'FM29hw30la', 'Mercator': {'x': -8392119.263, 'y': 4826
433.122}, 'OSM': {'note_url': 'https://www.openstreetmap.org/note/new#map=1
7/39.91678/-75.38769&layers=N', 'url': 'https://www.openstreetmap.org/?mlat=
39.91678&mlon=-75.38769#map=17/39.91678/-75.38769'}, 'UN_M49': {'regions':
{'AMERICAS': '019', 'NORTHERN_AMERICA': '021', 'US': '840', 'WORLD': '001'},
'statistical_groupings': ['MEDC']}, 'callingcode': 1, 'currency': {'alternat
e_symbols': ['US$'], 'decimal_mark': '.', 'disambiguate_symbol': 'US$', 'htm
l_entity': '$', 'iso_code': 'USD', 'iso_numeric': '840', 'name': 'United Sta
tes Dollar', 'smallest_denomination': 1, 'subunit': 'Cent', 'subunit_to_uni
t': 100, 'symbol': '$', 'symbol_first': 1, 'thousands_separator': ','}, 'fla
g': 'us', 'geohash': 'dr47hd8k5e059dershtq', 'qibla': 57.59, 'roadinfo': {'dr
ive_on': 'right', 'speed_in': 'mph'}, 'sun': {'rise': {'apparent': 163204848
0, 'astronomical': 1632043020, 'civil': 1632046860, 'nautical': 1632044940},
'set': {'apparent': 1632092520, 'astronomical': 1632011520, 'civil': 1632094
140, 'nautical': 1632009600}}, 'timezone': {'name': 'America/New_York', 'now
_in_dst': 1, 'offset_sec': -14400, 'offset_string': '-0400', 'short_name':
'EDT'}, 'what3words': {'words': 'vision.sunset.muddy'}}, 'bounds': {'northea
st': {'lat': 39.929567, 'lng': -75.378087}, 'southwest': {'lat': 39.911963,
'lng': -75.399685}}, 'components': {'ISO_3166-1_alpha-2': 'US', 'ISO_3166-1_
alpha-3': 'USA', '_category': 'place', '_type': 'city', 'continent': 'North
America', 'country': 'United States of America', 'country_code': 'us', 'coun
ty': 'Delaware County', 'state': 'Pennsylvania', 'state_code': 'PA', 'town':
'Media'}, 'confidence': 8, 'formatted': 'Media, Pennsylvania, United States
of America', 'geometry': {'lat': 39.91678, 'lng': -75.38769}}]
```

In [8]:

```
result
```

Out[8]:

```
[{'annotations': {'DMS': {'lat': "39° 55' 0.40800'' N",
    'lng': "75° 23' 15.68400'' W"},
   'FIPS': {'county': '42045', 'state': '42'},
   'MGRS': '18SVK6686618592',
   'Maidenhead': 'FM29hw30la',
   'Mercator': {'x': -8392119.263, 'y': 4826433.122},
   'OSM': {'note_url': 'https://www.openstreetmap.org/note/new#map=17/39.916
78/-75.38769&layers=N',
    'url': 'https://www.openstreetmap.org/?mlat=39.91678&mlon=-75.38769#map=
17/39.91678/-75.38769'},
   'UN_M49': {'regions': {'AMERICAS': '019',
     'NORTHERN_AMERICA': '021',
     'US': '840',
     'WORLD': '001'},
    'statistical_groupings': ['MEDC']},
   'callingcode': 1,
   'currency': {'alternate_symbols': ['US$'],
    'decimal_mark': '.',
    'disambiguate_symbol': 'US$',
    'html_entity': '$',
    'iso_code': 'USD',
    'iso_numeric': '840',
    'name': 'United States Dollar',
    'smallest_denomination': 1,
    'subunit': 'Cent',
    'subunit_to_unit': 100,
    'symbol': '$',
    'symbol_first': 1,
    'thousands_separator': ','},
   'flag': 'us',
   'geohash': 'dr47hd8k5e059dershtq',
   'qibla': 57.59,
   'roadinfo': {'drive_on': 'right', 'speed_in': 'mph'},
   'sun': {'rise': {'apparent': 1632048480,
     'astronomical': 1632043020,
     'civil': 1632046860,
     'nautical': 1632044940},
    'set': {'apparent': 1632092520,
     'astronomical': 1632011520,
     'civil': 1632094140,
     'nautical': 1632009600}},
   'timezone': {'name': 'America/New_York',
    'now_in_dst': 1,
    'offset_sec': -14400,
    'offset_string': '-0400',
    'short_name': 'EDT'},
   'what3words': {'words': 'vision.sunset.muddy'}},
  'bounds': {'northeast': {'lat': 39.929567, 'lng': -75.378087},
   'southwest': {'lat': 39.911963, 'lng': -75.399685}},
  'components': {'ISO_3166-1_alpha-2': 'US',
   'ISO_3166-1_alpha-3': 'USA',
   '_category': 'place',
   '_type': 'city',
   'continent': 'North America',
   'country': 'United States of America',
   'country_code': 'us',
   'county': 'Delaware County',
   'state': 'Pennsylvania',
   'state_code': 'PA',
   'town': 'Media'},
  'confidence': 8,
  'formatted': 'Media, Pennsylvania, United States of America',
  'geometry': {'lat': 39.91678, 'lng': -75.38769}}]
```

## 4. Visualise the connections on a map and observe your network

In [9]:

```
from opencage.geocoder import OpenCageGeocode
key = "081afd6c6215402cb8625e6a3c722f74"
geocoder = OpenCageGeocode(key)
address='First Name'
result = geocoder.geocode(address, no_annotations="1")
result[0]['geometry']
```

Out[9]:

```
{'lat': 52.5477127, 'lng': -1.935611}
```

In [10]:

```
import pandas as pd
data=pd.read_csv('Connections1.csv')
data.head(10)
```

Out[10]:

| | First Name | Last Name | Email Address | Company | Position | Connected On |
|---|---|---|---|---|---|---|
| **0** | Micro Media | Films | NaN | Micro Media Films | Owner | 16-Sep-21 |
| **1** | JENIFER | T | NaN | TechLife Global | .NET Developer | 16-Sep-21 |
| **2** | Bhattacharya | Shreya | NaN | The Sparks Foundation | Human Resources Executive | 16-Sep-21 |
| **3** | Iswariya | T | NaN | NaN | NaN | 16-Sep-21 |
| **4** | PRIYAA | LAKSHMI A S | NaN | NaN | NaN | 16-Sep-21 |
| **5** | Yuvasri | Matathammal | NaN | Geno Consulting Services Pvt Ltd. | Senior Human Resources Recruiter | 6-Jul-21 |
| **6** | Eagle Eye | IT Solutions | NaN | NaN | NaN | 30-Jun-21 |
| **7** | Yuvashree | M | NaN | NaN | NaN | 30-Jun-21 |
| **8** | Franklin | Solomon | NaN | NaN | NaN | 24-Jun-21 |
| **9** | Jeniffer | R | NaN | NaN | NaN | 24-Jun-21 |

In [11]:

```
addresses = data["Connected On"].values.tolist()
key = "081afd6c6215402cb8625e6a3c722f74"
geocoder = OpenCageGeocode(key)
latitudes = []
longitudes = []
for address in addresses:
    result = geocoder.geocode(address, no_annotations="1")

    if result and len(result):
        longitude = result[0]["geometry"]["lng"]
        latitude = result[0]["geometry"]["lat"]
    else:
        longitude = "N/A"
        latitude = "N/A"

    latitudes.append(latitude)
    longitudes.append(longitude)
```

In [12]:

```
data["latitudes"] = latitudes
data["longitudes"] = longitudes
data.head(5)
```

Out[12]:

| | First Name | Last Name | Email Address | Company | Position | Connected On | latitudes | longitudes |
|---|---|---|---|---|---|---|---|---|
| **0** | Micro Media | Films | NaN | Micro Media Films | Owner | 16-Sep-21 | 51.017593 | 20.221634 |
| **1** | JENIFER | T | NaN | TechLife Global | .NET Developer | 16-Sep-21 | 51.017593 | 20.221634 |
| **2** | Bhattacharya | Shreya | NaN | The Sparks Foundation | Human Resources Executive | 16-Sep-21 | 51.017593 | 20.221634 |
| **3** | Iswariya | T | NaN | NaN | NaN | 16-Sep-21 | 51.017593 | 20.221634 |
| **4** | PRIYAA | LAKSHMI A S | NaN | NaN | NaN | 16-Sep-21 | 51.017593 | 20.221634 |

In [13]:

```
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\mahalakshmi\anaconda3\lib
\site-packages (0.12.1)
Requirement already satisfied: branca>=0.3.0 in c:\users\mahalakshmi\anacond
a3\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: requests in c:\users\mahalakshmi\anaconda3\li
b\site-packages (from folium) (2.25.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\mahalakshmi\anaconda3
\lib\site-packages (from folium) (2.11.3)
Requirement already satisfied: numpy in c:\users\mahalakshmi\anaconda3\lib\s
ite-packages (from folium) (1.19.5)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\mahalakshmi\anac
onda3\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\mahalakshmi\ana
conda3\lib\site-packages (from requests->folium) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mahalakshmi\an
aconda3\lib\site-packages (from requests->folium) (2020.12.5)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\mahalakshmi
\anaconda3\lib\site-packages (from requests->folium) (1.26.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\mahalakshmi\anaconda
3\lib\site-packages (from requests->folium) (2.10)
```

In [14]:

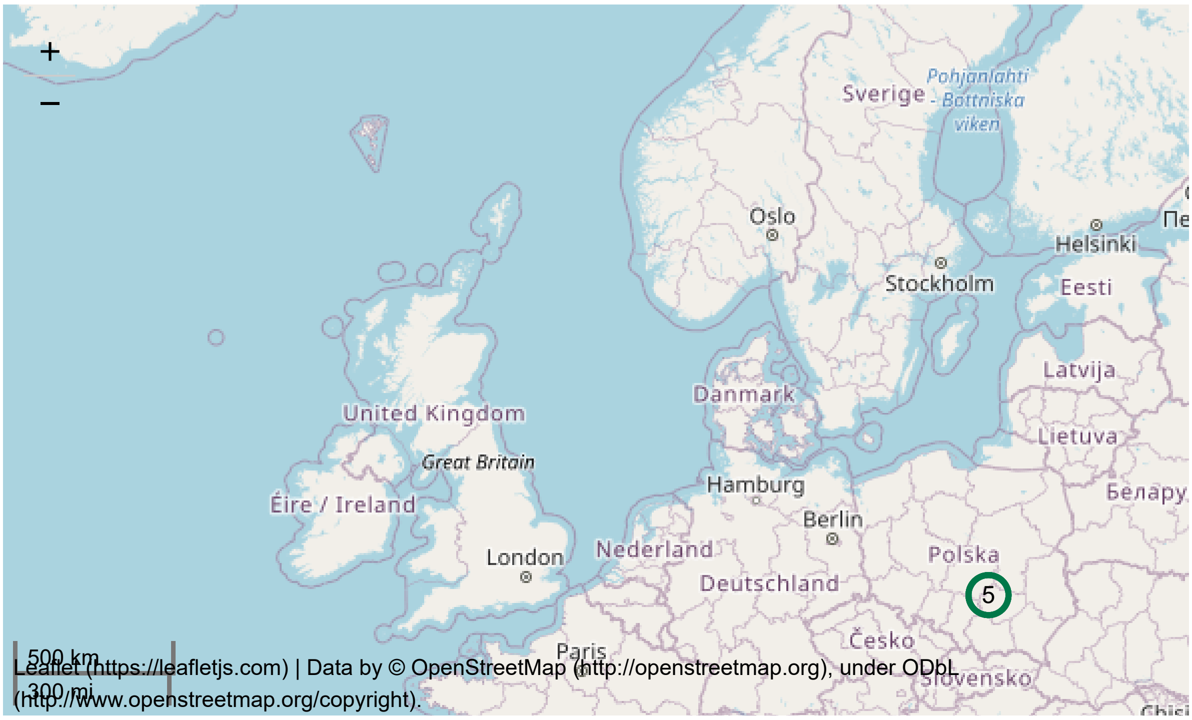
```
import folium
from folium.plugins import FastMarkerCluster
```

In [16]:

```
folium_map= folium.Map(location=[51.017593,20.221634],zoom_start=4.4,control_scale=True)
FastMarkerCluster(data[['latitudes', 'longitudes']].values.tolist()).add_to(folium_map)
folium.LayerControl().add_to(folium_map)
for row in data.iterrows():
    row=row[1]
    folium.CircleMarker(location=(row["latitudes"],
                                  row["longitudes"]),
                        radius= 10,
                        color="#007849",
                        popup=row['Company'],
                        fill=False).add_to(folium_map)

folium_map
```

Out[16]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [ ]:

```

```