IBM Data Science Capstone Project
Does post-secondary education affect the rate of crime in cities?

An investigation of the relation between the increase in the number of academic centers and libraries and a decrease in the rate of crime in American cities.

Iman Soltani

2. Data

2.1. Crime rate in American cities

To determine the rate of crime in American cities, a Wikipedia article on the list of a selected United States cities by crime rates, was used. In this article, there is a table containing about 90 most populous American cities along with their crime rates per 100,000 people based on Federal Bureau of Investigation (FBI) Uniform Crime Reports (UCR) statistics from 2017. Both violent crime data including murder and nonnegligent manslaughter, rape, robbery, aggravated assault, as well as property crime including burglary, larceny-theft, motor vehicle theft, and arson were included. The FBI has used the 2009 population estimate in this table (1).

We imported Pandas and NumPy libraries, and we used read_html function from Pandas library to read the webpage table data into a data frame. We performed data cleaning steps on the table like dropping the rows where the total crime values were missing using the dropna method.

```
Cdf1.dropna(subset=['Total Violent Crime','Total Property crime'], axis=0)
```

The numbers in the cities were removed using str.replace method as the following.

```
Cdf2['City'] = Cdf2['City'].str.replace('\d+', '')
```

2.2. Geographic coordinates of the selected most populous American cities.

In order to determine the geographic coordinates of the selected most populous American cities, we used the geopy.geocoder library to convert the name of the cities into their corresponding coordinates.
First, we added a column with the name of both city and state to avoid any mistake in case of the cities with similar names.

```
Cdf3['City, State']=Cdf3[['City','State']].apply(lambda x: ', '.join(x[x.notnull()]), axis = 1)
```

Then, we used geopy.geocoder and imported the Nominatim function to do the geocoding of the cities' names into geographic coordinates. To define an instance of the geocoder, we need to define a user_agent. We will name our agent us_explorer, as shown below.

```
geolocator = Nominatim(user_agent="us_explorer")
for i in range(Cdf3.shape[0]):
    location = geolocator.geocode(Cdf3.loc[i, 'City, State'])
    Cdf3.loc[i, 'Latitude'],  Cdf3.loc[i, 'Longitude'] = location.latitude, location.longitude
```

2.3. Creating a map of American cities using Folium library.

To plot the map we used matplotlib library with cm and colors modules. Also, it was needed to install and import the folium graphical map rendering library. Again geopy geocoder was used to determine the geographic coordinate of the United States.
Then, a map of the United States with the selected cities superimposed on top was created.

```
# create a map of United States using latitude and longitude values
map_USA = folium.Map(location=[latitude, longitude], zoom_start=3)

# add markers to map
for lat, lng, city, state in zip(Cdf3['Latitude'], Cdf3['Longitude'], Cdf3['City'], Cdf3['State']):
    label = '{}, {}'.format(City, State)
    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_USA)

map_USA
```

2.3. Number of post-secondary schools in American cities

To get detailed information about the cities post-secondary schools we used Foursquare, which is a location data provider website. So, we constructed some URL to make Foursquare API calls to search the cities for universities and colleges. Then, we received the json file of the corresponding detailed information upon calling get request.

The Foursquare API calls was used to explore the selected American cities and cluster them. Initially, we registered with Foursquare and received the client ID and Secret needed to call Foursquare API.

```
CLIENT_ID =     # your Foursquare ID
CLIENT_SECRET =   # your Foursquare Secret
VERSION = '20201227' # Foursquare API version
LIMIT = 1000 # A default Foursquare API limit value is 100
```

Then, we constructed the appropriate URL to make a Foursquare API call, upon which we could explore the universities and colleges in the selected American cities. Then a get request was called to receive a json file containing detailed information about the selected cities academic institutions. We began this process for only one sample city Mobile, Alabama to practice its viability and how to interpret the received json file data, which is in the form of a dictionary, with different keys and their corresponding values. So, we picked the appropriate data in the json file and pursued data cleaning to make an appropriate data-frame containing the required data. Accordingly, we picked the relevant data columns. Since the relevant information, which is the type (Category) of the venues, was in the name key of the categories column. So, we defined a function to extract the value of the name from the column. Finally, we chose the venue categories to exclude the venues that are not in fact universities. After that we felt satisfied with the acquired and cleaned data, we do the same procedure for all of the selected cities, using a for loop to automatize it, like the following.

```
def getSchools(names, latitudes, longitudes, radius=5000):

    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/search?&query=University+College&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']['venues']

        # return only relevant information for each nearby venue
        for v in results:
            try:
```

```
            venues_list.append([(
                name, lat, lng,
                v['name'],
                v['location']['lat'],
                v['location']['lng'],
                v['categories'][0]['name'])])
        except:
            pass

    city_colleges = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    city_colleges.columns = ['City',
                'City Latitude',
                'City Longitude',
                'College',
                'College Latitude',
                'College Longitude',
                'College Category']

    return(city_colleges)
```

Then we wrote a code to run the above function on each city and create a new dataframe called colleges.

```
colleges = getSchools(names=Cdf3['City'],
                              latitudes=Cdf3['Latitude'],
                              longitudes=Cdf3['Longitude']
                             )
```

References

1. List of United States cities by crime rate - Wikipedia: https://en.wikipedia.org/wiki/List_of_United_States_cities_by_crime_rate