This is my github link <https://github.com/li312/IS452-final-project>

My final project is about investigating the safety problem of the city of Rockford. The final one is a little bit different with my original final proposal. I cut some task. I didn’t analyze the time series of the crimes and the average day for a crime cause I think they are not useful tasks for analysis. I did the partial part for the areas that crimes occurred using location.py file because I couldn’t do all. The Geopy API has the constrain that I can only find 2500 address a day, so, I just have less than 3000 places. You can run the location.py to verify that my code works. Also, I didn’t predict the crime rate for next year because I don’t know how to do that.

Notice:

1. If you run the final.py for the first time, you should type d before e and f, that means you should run location function first because the following two function need the two text files that generated in the location. If you run the code for the second time, the problem doesn’t matter.
2. You may encounter some error like runtime error, service timed out. That are not because of my code. Geopy API is the reason that cause the error. I don’t know why it cause these type of error. All you need to do is turn the program off and reopen it and run again.

Now it is the time to walk through the project (You can also see the code comment). The project will need you to use two csv files which are offensive.csv and 72hr.csv as dependencies and one python package called geopy. The two csv files contain the crime in Rockford from 2011 to 2016 and the recent 72 hours police call in Rockford. Remember to install geopy through your terminal.

1.First download the offensive.csv and 72hr.csv to your computer, then download the final.py and location.py to your computer. Put them into a same file. Next, open your terminal, type the following command to install geopy.

pip install geopy

2. Open the final.py file, run the code. The console will show as following

What do you want to see?:

a.number of crimes

b.number of crime each year

c.crime type

d.top 10 dangerous place

e.distance between police station and the place

f.if the dangerous places changes comparing to recent police phone call

This is the results from the main function. It will just let users to decide which function they want to use. First wrote moreFunction==”yes”. Then, I wrote a while loop: while moreFuncyion[0]==”yes”, means if user choose yes, the program will continue. Under the while loop, I wrote an input statement to let users to choose the functions following a, b,c,etc. For example, if the input is a, then we take the getNumCrime function. There is another input statement: moreFunction=input(“Do you want to continue?”), if you choose yes or y, the program will continue. If you choose another one, the program will stop.

3. If you type a, you will run the getNumCrime() function.

It just computes how many crimes occurred from 2011 to 2016. The first thing is to read the csv file to a variable called offense. Then, I used next() method to avoid reading the header. Then I initialized 0 to a count variable. Then, I iterate the offense and add 1 to count under the for loop. The final count should be the number of the crimes. After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

4. If you type b, you will run the CrimeEachYear() function

The function CrimeEachYear() shows the number of crime each year. The first thing is to read the csv file to a variable called offense. Then, I used next() method to avoid reading the header. I assigned an empty list. Then I iterated every row in the csv file. And for each row, I append the year to the list by finding the third column and slicing the first element to the fourth (if you look at the csv file, the third column is occurred date, and the first 4 number is the year, so row[2][0:4] would extract the year). After appending the years to the list. I used counter(list) to find the number of each year, and that would be the number of crimes for each year. Then I print out the result. After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

5. If you type c, you will run the crimeType() function.

The function crimeType() shows the crime type of in the city of Rockford. First, I assigned an empty list and read the csv file to a variable called offense. Then, I used next() method to avoid reading the header. Then I iterated every row in the csv file, for each row, I append the crime type to the list. We can find the crime type in the second column of the file, and we slice the string from the position 6 to the end. So, row[1][6:] would extract the crime type. Next, I wrote dict(Counter(list)) to have a dictionary with the crime types as keys and their frequencies as their occurred times. After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

6. If you type d, you will run the location() function.

The function location() would show the top 10 dangerous places in the city of Rockford based on the offense csv file. Same as before, I first assigned an empty coordlist and and read the csv file and omit the header. Then, I iterated every row in the file. Under the for loop, I first assigned an empty list for storing the current coordinate. I try to append the latitude and longitude in the list; the latitude is in the seventh column and the longitude is in the sixth column in the offensive.csv file, then, I append the current list (and convert it to tuple) to the coordlist. The exception will happen, because the last row doesn’t have coordinate value for some reasons, and if we go to the last row, I will break the loop.

I used Counter(cordlist) to count their frequency. Now, I open two files, one is coordinate file the other one is address file for future use. Then, I wrote for coord in coordinate\_count.most\_common(10) to have the coordinate with top 10 frequencies. Every coord is a tuple with coordinate and their frequencies, so, coord[0][0] is latitude and cood[0][1] is longitude, and I concatenate latitude and longitude with ‘,’ and assigned it to a variable called coordinate. I have to do this because the geo API only accept strings to convert it to actual address. I used geocoder API to convert the coordinate to the address, and wrote the coordinate to the coordinate file and wrote address to the address file. Finally, I just print out the top 10 address.

After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

7. if you type e, you will run isLocationChange() function

The function isLocationChange shows if the top 10 dangerous place in offense file changed comparing to 72 hour recent police call. First, I open the 72hr csv file and omit the header. I assigned an empty location list and then wrote a for loop to find all locations which is in column 4 and append it to the location list. Then, I open the previous address file I generated in location() functin and append the top 10 address to an empty address list. Now, I iterate the new top 10 frequent location using location\_count.most\_common(10). And I wrote a nested for loop under the previous one to iterate the address in the old top 10. The location[0].split(‘ ’)[0] is the house number of the address; the each location looks like (‘1234 xxxx st, IL’,4), So my code will extract 1234. If the house number is substring in the old address, then they are the same address, and I then I printed out that place is still dangerous.

After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

8. If you type f, you will run disPolice() function

The function disPolice compute the distance between police station and the top 10 dangerous places. I google the address of the police station and assigned it to a variable called policestation. I used geopy API to get the coordinate of the policestation and append it to an empty stationlist. Then, I opened the coordfile and the address file. I put the address to the address list. Then, I assigned 0 to a count variable. Then, I wrote for cord in cord file to get the coordinate of top 10 dangerous places. Then, I tried to convert the string coordinate to tuple. First, I split the “,”. Then, I wrote cord=tuple(map(float,cord)) to convert the list coordinate to tuple. The map method will convert the type element in the cord list to float. And tuple() convert the list of float coordinate to tuple. Then, I wrote the vincenty(cord,stationList[0]).miles to compute the distance. The cord is the previous tuple coordinate, the stationlist[0] is the coordinate of the police station. Addresscount+=1 will shows the next address in the address list (addressList[addresscount]).

After this function finished, the console will pop up the question that if you want to continue, if you type yes or y, then continue, if you choose another, then the program closes.

9. open the location.py file to find the area of the crimes occurred.

This code is similar as the location function in final.py. The difference is the last few lines. In this program, the code iterate all place coordinate from 2011 to 2016 and use the API to convert them to actual addresses. Then, it writes all the address to a file called location.txt.

You will have an error in some point because of the API. I recommend you stop and don’t run again. Because if you run again, you may start at the beginning and overwrite the location.txt and will have another error in some point. That is pretty annoying. In order to get over 2500 results, I rerun the code multiple times and change the code that let it start parsing the coordinate at the line where it have error and stop at previous time, and store them in several different files. I need to create a new file as I rerun one time. That is a horrible work, so, I do not recommend you do the same as I did. Just run the code and see I can do that, but constrains didn’t allow me to finish the work.