

Thomas McLinden and Victor Liu

12/3/24

CMPSC 463, Section 1

Professor Janghoon Yang

Project 2: Crime Tracker

Goal

The goal of this project was to utilize the skills and knowledge of this course to create an application to help solve a real-world problem. In this case, we created a crime tracker application, where users could enter any zip code in Philadelphia County to view its crime statistics.

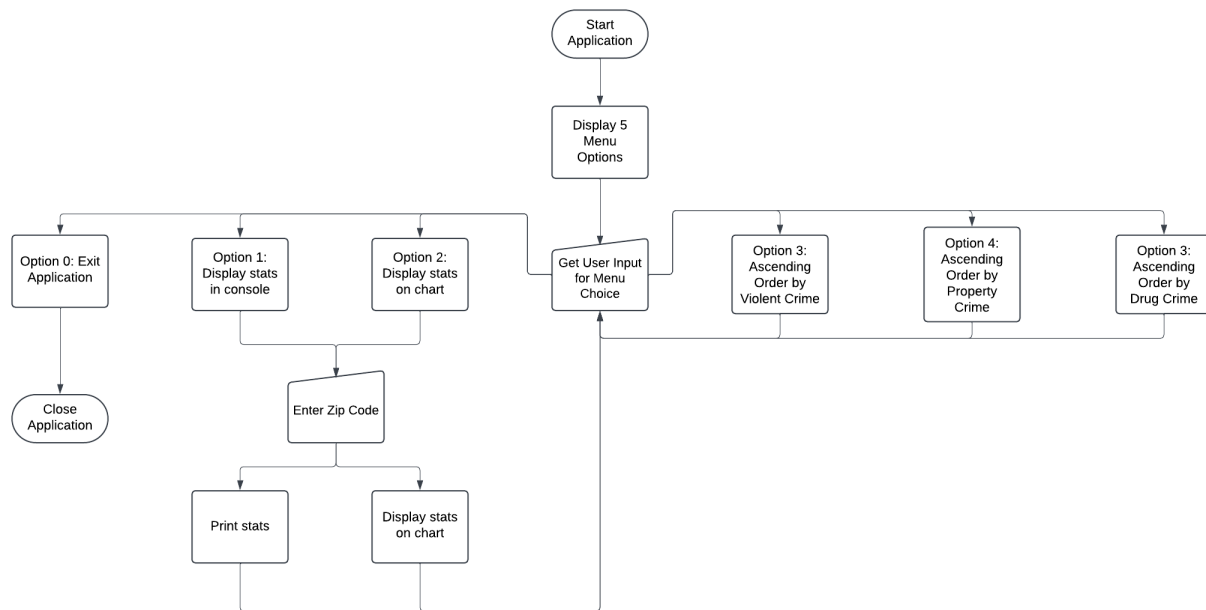
Significance

This application is primarily intended for residents, students, and employees of Philadelphia. By allowing users to obtain the crime statistics of any Philadelphia zip code, users can make decisions on areas they deem unsafe and plan accordingly. This is especially important for Philadelphia residents, tourists, students, and employees.

Installation and Instruction to Use:

Installation is quick and easy. The latest version of python will need to be installed and the main.py file is available on the GitHub repository.

Structure of the Code:



The application begins by initializing the lists containing the zip codes, town names, and their crime statistics. The options are printed for the user on the console and a while loop is activated, where users select options on what functions to run. The user also has the option to close the program, which will break the loop.

List of Functionalities and Verification Results:

- The console prints the options available to the user

```
C:\Users\SF_00\PycharmProjects\CrimeTracker\.venv\Scripts\python.exe C:\Users\SF_00\PycharmProjects\CrimeTracker\main.py
--- Crime Data Options ---
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
0. Quit
Enter your choice:
```

- Print statistics in the console:

This function prints out the crime statistics of the zip code entered by the user.

```

153     if choice == 1:
154         zip_code = int(input("Enter ZIP code: "))
155         for neighborhood, zip_code_mapping in neighborhood_zip_mapping.items():
156             if zip_code == zip_code_mapping:
157                 crime_rates = crime_data.get(neighborhood, None)
158                 if crime_rates:
159                     print(f"\nCrime Data for {neighborhood} (ZIP: {zip_code}):")
160                     print(f"Violent Crime Rate: {crime_rates[0]} per 1,000 residents")
161                     print(f"Property Crime Rate: {crime_rates[1]} per 1,000 residents")
162                     print(f"Drug Crime Rate: {crime_rates[2]} per 1,000 residents")
163                 break
164             else:
165                 print("No neighborhood found for that ZIP code.")

```

- Display statistics on a chart:

```

167     elif choice == 2:
168         zip_code = int(input("Enter ZIP code to view chart: "))
169         for neighborhood, zip_code_mapping in neighborhood_zip_mapping.items():
170             if zip_code == zip_code_mapping:
171                 crime_rates = crime_data.get(neighborhood, None)
172                 if crime_rates:
173                     plt.bar(x=['Violent Crime', 'Property Crime', 'Drug Crime'], crime_rates)
174                     plt.title(f"Crime Rates for {neighborhood} (ZIP: {zip_code})")
175                     plt.ylabel("Crimes per 1,000 residents")
176                     plt.show()
177                 break
178             else:
179                 print("No neighborhood found for that ZIP code.")

```

- Sort by Violent Crime, Property Crime, or Drug Crime:

```

135     # Function to sort neighborhoods by a specific crime rate
136     def sort_by_crime_rate(criterion): 3 usages  ⚡ thomasmclinden
137         sorted_df = df.sort_values(by=criterion, ascending=True)
138         print(f"\nNeighborhoods sorted by {criterion}:")
139         print(sorted_df)

```

```

180     elif choice == 3:
181         sort_by_crime_rate('Violent Crime Rate')
182     elif choice == 4:
183         sort_by_crime_rate('Property Crime Rate')
184     elif choice == 5:
185         sort_by_crime_rate('Drug Crime Rate')

```

- Exit the application:

```

186         elif choice == 0:
187             print("Exiting the program.")
188             break
189         else:
190             print("Invalid choice, please try again.")

```

Showcasing the Achievement of Project Goals:

- Print statistics in the console:

```

Enter your choice: 1
Enter ZIP code: 19116

Crime Data for Byberry (ZIP: 19116):
Violent Crime Rate: 0.05 per 1,000 residents
Property Crime Rate: 1.5 per 1,000 residents
Drug Crime Rate: 0.0 per 1,000 residents

--- Crime Data Options ---
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
0. Quit
Enter your choice:

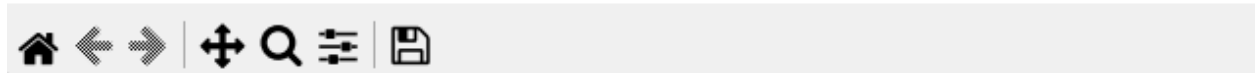
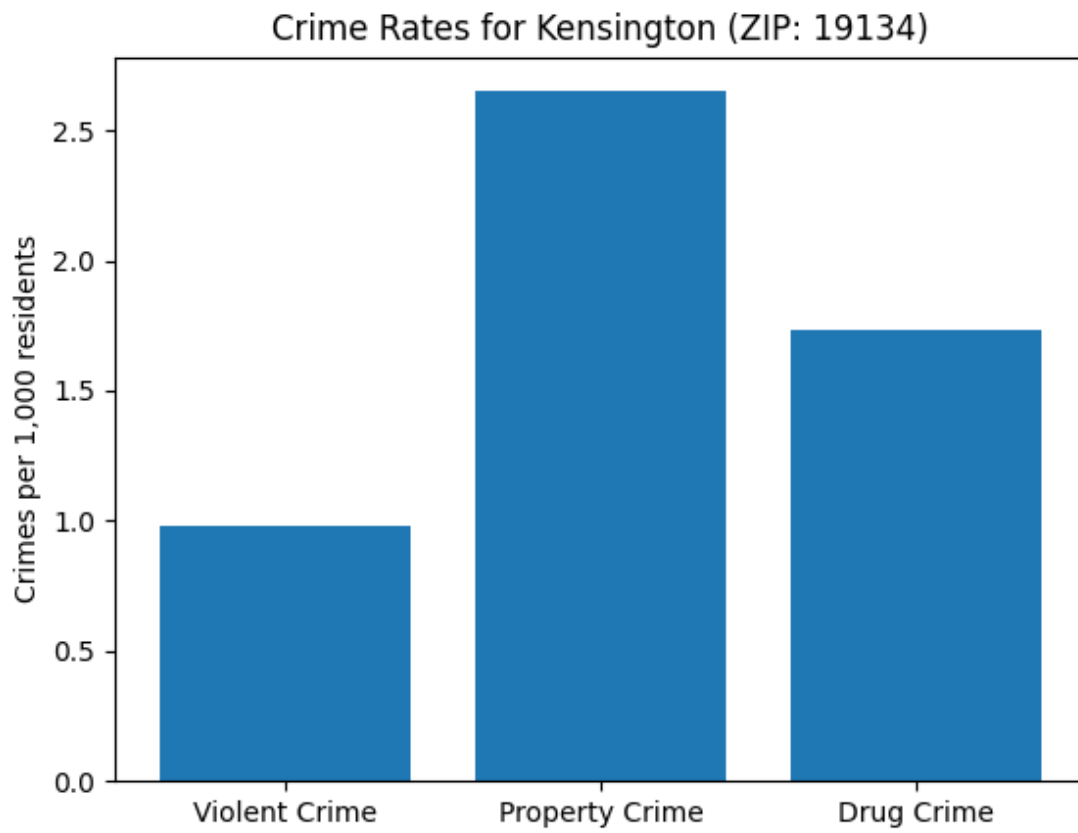
```

- Display Statistics on the Chart:

```

--- Crime Data Options ---
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
0. Quit
Enter your choice: 2
Enter ZIP code to view chart: 19134

```



- Sort by Violent Crime:

Enter your choice: 3

Neighborhoods sorted by Violent Crime Rate:

Neighborhood	Violent Crime Rate	Property Crime Rate	Drug Crime Rate
Torresdale	0.04	1.47	0.00
Byberry	0.05	1.50	0.00
Schuylkill Southwest	0.08	3.26	0.00
Roxborough	0.11	3.82	0.00
Riverfront	0.12	1.50	0.00
Girard Estates	0.12	2.21	0.00
Marconi Plaza-Packer Park	0.12	2.12	0.00
Poplar-Ludlow-Yorktowne	0.12	2.29	0.00
Fox Chase	0.14	2.24	0.00
Brewerytown	0.14	3.20	0.00
Mount Airy	0.14	1.64	0.00
Somerton	0.16	2.73	0.00
Bustleton	0.17	2.33	0.00
Cedar Brook	0.21	1.91	0.06
Pennsport-Whitman-Queen	0.22	3.08	0.00
Manayunk	0.25	3.58	0.00
Chestnut Hill	0.28	3.88	0.09

- Sort by Property Crime:

	Neighborhoods sorted by Property Crime Rate:		
	Violent Crime Rate	Property Crime Rate	Drug Crime Rate
Neighborhood			
Bridesburg	0.32	1.27	0.00
Torresdale	0.04	1.47	0.00
Riverfront	0.12	1.50	0.00
Byberry	0.05	1.50	0.00
Mount Airy	0.14	1.64	0.00
Cobbs Creek	0.93	1.65	0.00
Fairhill	1.58	1.88	0.94
Cedar Brook	0.21	1.91	0.06
South Philadelphia	0.65	1.98	0.00
Logan-Fern Rock	0.77	2.00	0.15
Oak Lane	0.28	2.09	0.00
Olney	0.56	2.11	0.05
Marconi Plaza-Packer Park	0.12	2.12	0.00
Morris Park	0.39	2.12	0.00
Girard Estates	0.12	2.21	0.00
Fox Chase	0.14	2.24	0.00
Poplar-Ludlow-Yorktowne	0.12	2.29	0.00
Bustleton	0.17	2.33	0.00
Bella Vista/Southwark	0.50	2.48	0.00
Hartranft	0.89	2.50	0.10

- Sort by Drug Crime:

Neighborhoods sorted by Drug Crime Rate:			
Neighborhood	Violent Crime Rate	Property Crime Rate	Drug Crime Rate
Allegheny West	1.82	4.52	0.00
Bella Vista/Southwark	0.50	2.48	0.00
Bridesburg	0.32	1.27	0.00
Brewerytown	0.14	3.20	0.00
Bustleton	0.17	2.33	0.00
Byberry	0.05	1.50	0.00
Cobbs Creek	0.93	1.65	0.00
East Falls	0.41	3.45	0.00
Elmwood	1.45	2.60	0.00
Marconi Plaza-Packer Park	0.12	2.12	0.00
Manayunk	0.25	3.58	0.00
Harrowgate	0.50	3.00	0.00
Girard Estates	0.12	2.21	0.00
Grays Ferry	0.61	3.81	0.00
Haddington-Carroll Park	0.70	2.55	0.00
Fox Chase	0.14	2.24	0.00
Poplar-Ludlow-Yorktowne	0.12	2.29	0.00
Schuylkill Southwest	0.08	3.26	0.00
Roxborough	0.11	3.82	0.00
Wynnefield	0.93	3.14	0.00

- Exit the Application:

```

--- Crime Data Options ---
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
0. Quit
Enter your choice: 0
Exiting the program.

Process finished with exit code 0

```

Discussion and Conclusions:

While the application serves its main purpose, by providing users with crime-related data in an easily accessible manner, it is not without its limitations. The application does not

have access to a database to update these statistics; Each value is hard coded into the application itself, which would require developers to manually edit these values for them to be updated. In terms of classroom materials implemented into this project, the application utilizes *matplotlibs*, a library frequently used during homework assignments. Furthermore, the *pandas* library utilizes the TImsort algorithm, a hybrid between merge sort and insertion sort. This is especially useful when sorting real-world data.