Project Report: Relationship Analysis between Parks and Graffiti

Environment Setup

To successfully run this Python project, the following dependencies must be installed:

* Requests: This library is essential for fetching data from the web.
* Matplotlib: Used for generating visual representations of the data.

To install these libraries, execute the following commands in terminal:



Code Highlights

1. Data Acquisition and Processing:

The project employs the ‘requests’ library to access open datasets concerning parks and graffiti. This raw data undergoes stages of parsing and cleaning to ensure quality and usability. The code snippet below illustrates the process from fetching to cleaning:

A black background with red text

Description automatically generated

1. Model Design:

Two main classes, ‘Park’ and ‘Graffiti’, encapsulate the relevant data and behaviors related to parks and graffiti, respectively. The ‘Park’ class, for example, includes methods such as ‘graffitis\_within\_km’ to compute the number of graffiti occurrences within a specified distance:

A screen shot of a computer program

Description automatically generated

1. Interaction Design and Visualization:

The project is interactive, guiding users through data exploration via command-line inputs and providing visual feedback through graphs generated with ‘matplotlib’. The visualization updates dynamically based on user input, enhancing interactivity, such as: 

Future Directions

Given more time, the project could be enhanced in several ways:

* Incorporate Additional Data Sources: Including data on other public amenities to provide a more comprehensive analysis.
* Improve User Interface: Transitioning from a command-line interface to a graphical user interface using libraries like ‘tkinter’ or ‘PyQt’ to make the application more user-friendly.
* Enhanced Data Processing: Introducing advanced data analysis techniques to offer deeper insights.

Reflection

* Learning Outcomes: This project has been a great opportunity to learn about data handling, visualization in Python, and object-oriented programming.
* Challenges: The most challenging aspect was dealing with inconsistent and incomplete data within the datasets.
* Most Valuable Part: Implementing and debugging the data visualization component was particularly rewarding, as it provided insights into the underlying data.
* Changes for Future Iterations: In future iterations, I would place a stronger emphasis on code modularity and reuse from the project’s onset to reduce redundancy.

This project not only enhanced my technical skills but also provided valuable insights into the relationship between urban elements like parks and graffiti, illustrating the potential of data-driven decision-making in urban planning.