Documentation for Household Maps Code

Overview

This project includes two Python scripts that generate interactive Folium maps visualizing household locations and routes displaying distinct metadata about each customer. Each household is marked on the map with color-coded icons based on water tank size.

Environment

- Developed in Python and run on Jupyter Notebook
    - You will need Jupyter Notebook to run these scripts, but you can create Python files containing the same code on any IDE to generate the same results and maps.

Dependencies

- Pandas library: used for data cleaning, analytics, and management
- Folium library: used to create interactive maps and HTML file styling and generation

Scripts

Script Name	**Focus**	**Input File**	**Output Files**
plotFullSetCoordinates.ipynb	points in regions	Client Coordinates.xlsx	fullmap.html, Dilkonmap.html, Thoreaumap.html, Navajomap.html
plotSubsets.ipynb	specific routes	Subsets_Afreen.xlsx	Time0.html, Time1.html, Time3.html, NN Time1.html, NN Time4.html

Features

- Interactive Folium maps focusing on specifc regions or routes
- Color-coded (based on water storage) markers representing households with popup and hover features
    - plotFullSetCoordinates.ipynb:
        - hover: household ID
        - pop-up: household size (# people), days since last visit, water storage type (# gallons)
        - legend: marker colors with associated water storage
    - plotSubsets.ipynb
        - hover: preferred ID (customer #)
        - pop-up: household size (# people), capacity (# gallons), delivery amount (# gallons), inventory (# gallons)
        - path segments: numbered segments by chronological order in route

Data Inputs

Each script takes in an excel file to create dataframes that are used for the map generation. Both scripts and their required data files are in the "afreenali" folder in the Git repository.

- Client Coordinates.xlsx → plotFullSetCoordinates.ipynb
    - The code creates a separate csv file of each sheet of the excel file. It only requires this one data file and will generate the rest of the csv and html files for you.
- Subsets_Afreen.xlsx → plotSubsets.ipynb
    - The code creates a separate csv file of each sheet of the excel file. It only requires this one data file and will generate the rest of the csv and html files for you.

Output Files

The following maps are in the "maps" folder in the "afreenali" folder in the Git repository.

- plotFullSetCoordinates.ipynb
    1. map with all coordinates involved in project → fullmap.html
    2. map with coordinates in Dilkon region → Dilkonmap.html
    3. map with coordinates in Thoreau region → Thoreaumap.html
    4. map with coordinates in Navajo region → Navajomap.html

- plotSubsets.ipynb
    1. map with 7 customers points, no route → Time0.html
    2. map with same customers points and route 1 → Time1.html
    3. map with same customers points and route 3 → Time3.html
    4. map with same customers points and 2 different routes → NN Time1.html
    5. map with same customers points and route 4 → NN Time4.html

Instructions to Run Code

1. Ensure that each data file is in the same location (folder) of its associated Jupyter Notebook (ipynb) file on your machine.
    i. Example: Client Coordinates.xlsx and plotFullSetCoordinates.ipynb should be in the same location, i.e. "C:\MapCode\Client Coordinates.xlsx" and "C:\MapCode\plotFullSetCoordinates.ipynb"
2. Make sure all packages are installed in your environment. You must have Pandas and Folium installed in your IDE.
3. On Jupyter Notebook, manually run each code chunk or run all cells from menu to generate all interactive maps.
    i. If you are not using Jupyter Notebook or are using a different IDE that supports Python, paste all the code from each script into a Python file used by the IDE and run.
4. All csv files and html files of the maps will be generated in the same root folder where the code is. These html files can be opened on any browser once generated and can be viewed with the included zoom and movable features. You can open them from the file location.
    i. Example: "C:\MapCode\Thoreaumap.html"
5. To make changes to any of the code or data files, please follow the same conventions I have organized in the current set-up. For example, to add or remove coordinates on the data files, ensure changes are in the same format as other initial data. Both scripts have detailed comments for extra documentation to help explain every part of the code.