Instructions for reproducing

City of Haverhill QAlert System Analysis

Team Members: Xiaochen Xue, Jing Yu, Zhitong Su, Kaijia You

This document is to illustrate how to reproduce results by client's new data. The codes of data cleaning, feature creation, and exploratory data analysis are written in R with tidyverse and sf packages installed; The codes of interactive maps are written in Jupyter Notebook, under Python 3 environment, with folium and geopandas libraries installed.

File Introduction

CSV Data

Haverhill.csv: This is the reduced version of the original csv. We simply deleted all rows with "informational" data types, and the rows with no geographical coordinates. Besides, it contains the additional columns including "Ward", "Ward_Precinct", "Polling_Station_Name", "CDBG".

Department.csv: This is a modified version of the "Haverhill.csv" file, where only 9 attributes are kept for visualization purposes, and the data are sorted by "Department" attribute.

Trash.csv: This is the file containing only trash-related requests modified from "department.csv". The 480 rows of data are sorted by "Request_Type" and are all important to trash collecting routes.

Trash: This is a document containing three csv files: "illegal_dumping_sites.csv", "recycling_missed_pickups.csv", "trash_missed_pickups.csv", which are the categorization of illegal dumping sites, recycling _missed_pickups, and trash_missed_pickups separately. All three csv files contain two columns, where the first column is address and the second column is frequency.

JSON Data

The GeoJSON data provided by the client was projected through the Pseudo-Mercator Projection type (EPSG: 3857), which is not suitable for the World GPS coordinates. We re-projected suitable JSON files to use, and the process is explained in the coding section.

Code Introduction

"Haverhill_Data_Cleaning.Rmd" : This is a R markdown document that contains text, codes, and results. This piece of codes create a new csv file of requests that filters out all "informational" request types, as well as the ones lacking coordinates. This document of codes can be reused to clean any future request data stored in the haverhill-request.csv file.

Procedure:

(1) Download R and R Studio
(2) Type install_packages("tidyverse") in the console
(3) Store "Haverhill_Data_Cleaning.Rmd" and "haverhill-request.csv" in the same directory"
(4) Open the directory through "Open Project"
(5) Run all chunks of codes
(6) A new csv file with all irrelevant rows removed is created in the same directory.

"Haverhill_Ward&Precinct.Rmd" : This is a R markdown document that contains text, codes, and results. This piece of codes contains two components: mapping all requests onto the map of Haverhill with the borders of every ward/precinct, creating three new columns "Ward", "Ward_Precinct", "Polling_Station_Name" in the original "haverhill-request.csv" file. This document of codes can be reused to process any future request data stored in the haverhill-request.csv file.

Procedure:

(1) Download R and R Studio
(2) Type install_packages("tidyverse") and install_packages("sf") in the console
(3) Store "Haverhill_Ward&Precinct.Rmd" and "haverhill-request.csv" in the same directory"
(4) Open the directory through "Open Project"
(5) Run all chunks of codes
(6) A new csv file with additional columns is created in the same directory. Although the codes are reusable with proper operation, we recommend the client to record the ward/precinct information in the process of data collection.

"Haverhill_CDBG.Rmd" : This is a R markdown document that contains text, codes, and results. This piece of codes contains two components: mapping all requests onto the map of Haverhill with

the borders of CDBG, creating a new column “CDBG” in the original “haverhill-request.csv” file. This document of codes can be reused to process any future request data stored in the haverhill-request.csv file.

Procedure:

(1) Download R and R Studio
(2) Type install_packages(“tidyverse”) and install_packages(“sf”) in the console
(3) Store “Haverhill_CDBG.Rmd” and “haverhill-request.csv” in the same directory”
(4) Open the directory through “Open Project”
(5) Run all chunks of codes
(6) A new csv file with an additional column is created in the same directory. Although the codes are reusable with proper operation, we recommend the client to record the CDBG information in the process of data collection.

“Haverhill_EDA.Rmd” : This is a R markdown document that contains text, codes, and results. This piece of codes shows all of the exploratory analysis and static visualizations. By clicking the button “Knit” on the top of R studio, it will automatically generate a pdf version with only text and graphics in the same directory.

Procedure:

(1) Download R and R Studio
(2) Type install_packages(“tidyverse”) and install_packages(“sf”) in the console
(3) Store “Haverhill_EDA” and “haverhill-request.csv” in the same directory”
(4) Open the directory through “Open Project”
(5) Click the button “Knit” on the top of R Studio
(6) A new pdf file with only text and graphics is created in the same directory.

Notes:

1. Storing all R codes (Rmd files) and the CSV file (haverhill-request.csv) in a same folder is recommended.
2. The Procedures “Download R and R Studio” and “Type install_packages(“tidyverse”) and install_packages(“sf”) in the console” only need to be done once, especially when it is the first time using R.

3. To make sure the read_csv() work, please manually rename all variable names in a way such that connect all words by "_" in Excel.