Citi Bike Availability Tool

User Guide

DS-GA 1002 Programming for Data Science
Final Project

Nora Barry (neb330)
Laura Buchanan (lcb402)
Alex Simonoff (ams889)

Background

Getting Started

Method

Introduction to Project

This project uses the NYC CitiBike Data and NOAA daily weather data to give the user an idea of Citi Bike availability at their nearby Citi Bike station. The nearby stations are determined by zip code, entered by the user. Past weather data indicating rain or snow during a given hour was excluded, since we believe that 1) the user will often not want to bike in the rain or snow, 2) that not correcting for when it rained or snowed in the past would predict more available bikes then there likely are on a sunny day, and 3) the user would likely not have trouble finding a bike when it is raining or snowing.

This application will be particularly helpful to larger groups, like tour groups or office-wide bike clubs, to determine whether enough bikes will be available for their group bike ride. If the requested number of bikes is not available at the closest station, the application lists all the stations within a zip code, so another station should be in walking distance.

Datasets

Two open datasets were used in the development of this application. The first is the Citi Bike data available via the NYC open data website:

https://nycopendata.socrata.com/

Specifically, the Citi Bike Data was found here:

https://www.citibikenyc.com/system-data

The data we included was Citi Bike Trip Histories and Station Feed.

The weather data was requested at:

http://www.ncdc.noaa.gov/cdo-web/search?datasetid=GHCND

Installation

The project can be found on: https://github.com/ds-ga-1007/final_project under the Pull Requests of neb330,lcb402, and ams889. Clone the project onto you local computer.

ATTENTION: To run this program, you need matplotlib's basemap module. Please follow the download and installation directions here:

http://matplotlib.org/basemap/

Instructions

In the terminal window, enter the directory where the program has been cloned from GitHub. Run:

$ python bikeFinder.py

The program will prompt the user with several questions to determine the appropriate search for the user. Questions include 1) which day of the week they would like the ride, 2) the month they would like to ride, 3) the time they would like to start riding (in hours), and 4) their zip code. Additionally, the user is asked if they would like to ride “now” and, if so, the time information is taken from the local computer's clock.

Analyses Summary

First, we downloaded the Citi Bike use data. This data was stored across multiple files, which had to be unzipped and combined. Separately, a file containing information about the Bike Station locations was downloaded. Second, the net change in bikes over each hour of data was computed for all stations. This net change was accounted for at each station with a rolling sum that was reset every 24 hours. This gave us the number of bikes available at each station for a given hour.

We then grouped the number of bikes available data by hour, day of the week, and month at a given station and averaged. If it rained or snowed on a given day, we excluded that data from the average. This was to prevent overestimating the average number of bikes that are likely available when we assume our user will be riding a bike, i.e., on a sunny day.

Finally, we search over this averaged data and output the stations in a given zip code, the number of bikes likely to be available, and a map of the stations in that borough for the time that the user requested.

Python

Our program was developed using python.

pandas

The pandas module was heavily used to store and manipulate our data.

NumPy

NumPy was used for data analysis.

matplotlib

The matplotlib was used for displaying the output of our program. In the first plot, a bar graph displays the likely number of bikes available at a given time at the stations located in a given zip code. The second plot plots the bike stations in the borough of the zip code given.