Brielle Broder

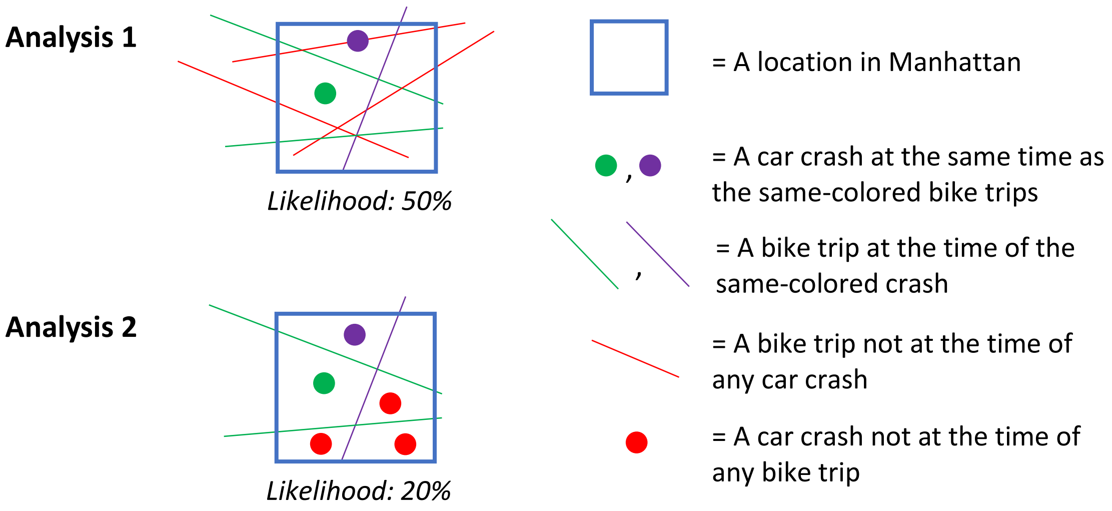
Analysis of Manhattan’s Vehicle Collision and Citi Bike Data

Submitted as Final Project for Applied Linux Programming and Scripting, Fall 2017

Introduction

In this study, I determined the locations in Manhattan where, as a fraction of the number of bikes at that location, there is a high likelihood of bikers being near car crashes. This analysis is based on the Citi Bike Trip Data which contains data about bike rentals in lower Manhattan and NYPD Motor Vehicle Collision records, both from January 2014. Additionally, I used this data to study if those locations correlate to the likelihood of a car crash occurring near at least one biker.

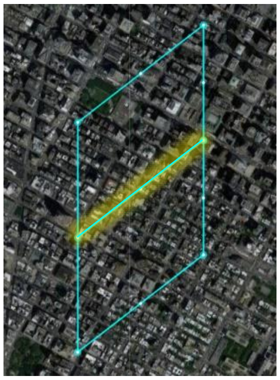
The images below demonstrate the difference between these two analyses. In the first example, the likelihood of a biker being near a car crash at a given location is determined in by taking the percent of bike trips near a car crash out of the total number of bike trips passing through that location. In the second example, the likelihood of a location is the percentage of car crashes near at least one bike out of all car crashes occurring at that location.


Analysis 1
Likelihood: 50%
Analysis 2
Likelihood: 20%
= A location in Manhattan
● , ● = A car crash at the same time as the same-colored bike trips
, = A bike trip at the time of the same-colored crash
= A bike trip not at the time of any car crash
● = A car crash not at the time of any bike trip

Code Walkthrough

From the bike data used, I read in each line of trip data and outputted the beginning and end locations (lat,long), the date of the trip, and the start and end times of the trip (hour:minute) to a file (“bikeEndpoints”). With this output, I checked each line to see if both the beginning and end locations occurred within the “Manhattan polygon.” The Manhattan polygon is an approximated shape which I created by establishing vertices around the edges of the island of Manhattan. I used the latitude and longitude of these points as the “y” and “x” coordinates of the polygon, respectively. If it was determined that the bike trip fell within the limits of the Manhattan polygon, the start and end coordinates for this bike trip were turned into “bike bubbles.”

Manhattan Polygon



Example of a "Bike Bubble"

A simplified approach[1] was used in these analyses where a "bike bubble" is the shape created by taking the y coordinate (latitude) of the start and end locations of each bike trip in the file and creating a bounding box that stretches from .005 degrees above the bike's y coordinate to .005 below the bike's y coordinate. This was done because it is not always the case that a biker will take the most direct route to get to his end location, especially in Manhattan with its grid-like streets and many driving regulations (such as one-way streets). Therefore, bikers may take less direct paths to reach their destinations; the broadened box of the bike route nearly ensures that the biker will have only ridden within the box on the way to the destination. (The x coordinate was not broadened because it is assumed that the biker will not ride past his destination.) This bike bubble was combined with the rest of the route information (date, start time, end time) and saved to a file ("bikeBlobs").

The crashes from the Motor Vehicle Collision data were handled in a similar, albeit simpler, manner. First, each car crash recorded in the file was checked to see if it occurred in January 2014. If so, the crash was then checked against the Manhattan polygon to limit the crash locations to the same area as the bike routes being considered. The crashes that were found to fall within the Manhattan were saved to a file ("manhatCrash").

In order to calculate the locations in Manhattan for which I determine percentages, I created small, overlapping boxes that, together, cover Manhattan. Each box is .005°x.005° wide/tall and each box is shifted over from its nearest left-hand neighbor by .001°. These boxes are created using a bounding rectangle that surrounds Manhattan but only those boxes that have all four corners fit within the Manhattan polygon are saved into a separate file ("boxCoords").

The "bikeBlobs" and "manhatCrash" files are then both read into another program. Each of the crashes is appended to a list called "noBikeAtCrash" which presumes that there was no bike near the crash at the time that it occurred. The bike routes are then looped through and each crash in "manhatCrash" is checked to see if 1) the date of the crash is the same date as the bike route, 2) the time of the crash is between the starting and ending times of the bike route, and 3) the location (latitude, longitude) of the crash falls within the bike bubble of the route. If the crash fulfills all three of these conditions, it is appended to a list of all crashes that occurred during the bike route. If, at this time, this crash was still in the "noBikeAtCrash" list, it is removed. After all crashes within a bike route have been appended, the bike route becomes a dictionary ("crashInRoute") key and the list of crashes is set as the value.

Separately, the values from "boxCoords" are read in line by line and are established as dictionary ("boxes") keys whose values are all initially set to a list of four elements: empty list, 0, empty list, 0.

Element	**Type**	**Category**
0	list	All car crashes that occurred near at least one bike in this box
1	int	All bike routes that had at least one car crash occur within its bubble in this box
2	list	All car crashes that did not occur near any bikes in this box
3	int	All bikes that did not have any car crashes occur within its bubble in this box

[1] A more enhanced approach would create an ellipsoid between the start and end locations to demonstrate the most likely paths a biker took to reach his destination.

Once the “boxes” dictionary is created, each key of the “crashInRoute” dictionary is looped through. For each key (ie bike route), every box (each key of “boxes”) is looped through and checked to see if some part of the bike bubble overlaps with the box. If so, all the crashes for each bike route are checked against each box’s dimensions and are appended to the list in the 0th spot of that box’s value (if they fall within bounds and have not already been appended). If the number elements appended to the 0th place is greater than 0, then the 1st spot in the box’s value is increased by 1, thereby counting the number of bikes that were near at least one car crash. Otherwise, the 3rd place is increased by one, the place which tallies the number of bikes not near any crashes in that box.

After all of the bike routes have been looped through, a similar process is done with the crashes that remain in the “noBikesAtCrash” list. In this case, however, the crashes that fall within the box (and are not already in the list) are appended to the 2nd place in the box value list which contains all of the crashes within the box that were not near any bikers.

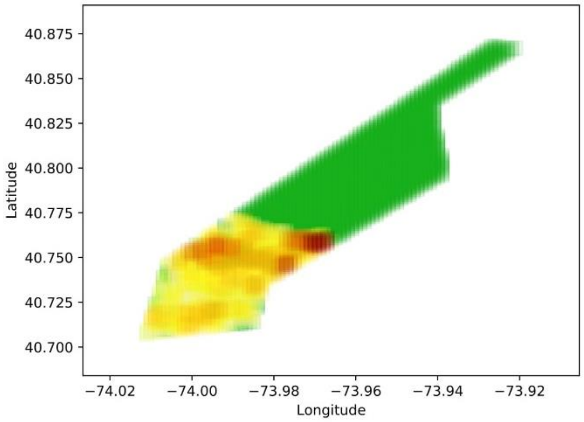
These filled boxes are then written out to a separate file. In the process of this writing, each box’s value elements 0 and 2 are changed into ints that represent the lengths of their lists of crashes, thereby summing the number of crashes that occurred of each category. This new file is then taken and used to calculate for each box:

1) The percentage of bikes near crashes out of the total number of bikes in that box and
2) The percentage of crashes near bikes out of the total number of crashes in that box.

Each of these boxes gets plotted onto two graphs, one that measures bike percentages (range: 0-1.3%) and one that measures crash percentages (range: 0-100%). This plot is essentially a scatter plot that plots a “box” shape in various colors corresponding to the box’s percentage in a given category (similar to a heat-map). Different colors are used in the bike-percentage plot and the crash-percentage plot to highlight the difference in range between the two graphs. Before plotting, the boxes are sorted so that those with the lowest percentages are plotted first. This ensures that boxes with higher percentages and, therefore, darker colors will be visible on top of the lighter colored boxes. The boxes graphed are not entirely opaque, allowing viewers to see the depth of the boxes plotted. An additional two plots were added – total bikes (range: 0-3611) and total crashes (range: 0-54) – for clarity.

Results

Bike Percentages[2]

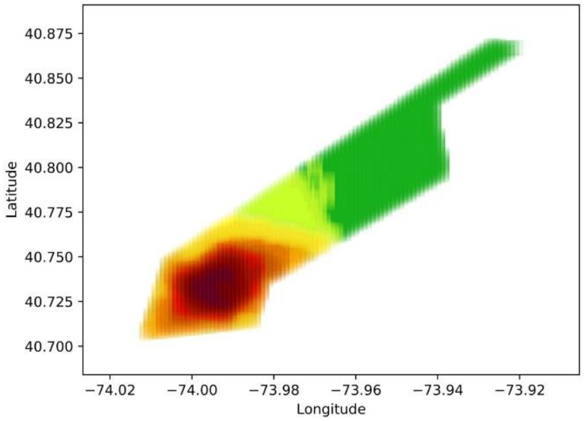


Max: ~1.2% Scale: .1%

These results show that the places in Manhattan with the greatest percentage of its bikers being near car crashes are the exits/entrances to bridges and tunnels leading into and out of Manhattan.

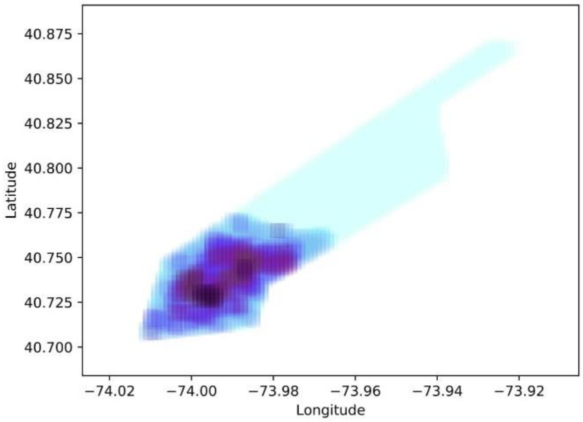
The three darkest points on this graph depict the areas surrounding the exits of the Queensboro Bridge, the Midtown Tunnel, and the Lincoln Tunnel[3].

Total Bikes[2]



Max: ~3600 Bikes Scale: 250 Bikes

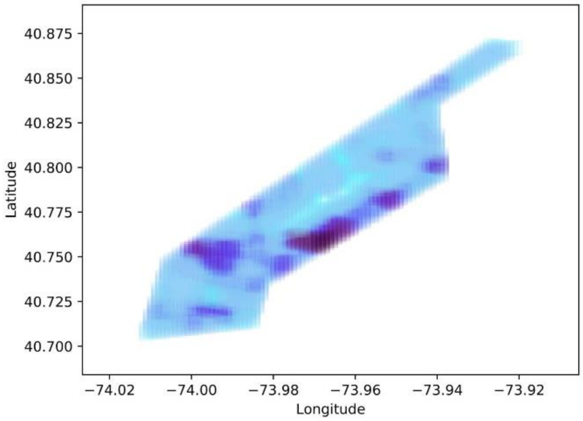
Crash Percentages[2]



Max: 100% Scale 10%

These results demonstrate that the places in Manhattan where there is the highest percentage of car crashes near bikes (100%) is highly correlated with the number of bikes travelling through that location but not with the likelihood (percentage) of those bikes being near a car crash because the sheer volume of bikes travelling through the area completely overwhelms the percentage of those bikes that are near crashes.

Total Crashes



Max: ~50 Crashes Scale: 5 Crashes

[2] Recall that the bike data only includes bike trips from lower Manhattan

[3] Matching up the latitudes and longitudes with Manhattan locations was done though Google Maps