THE BOROUGH OF MANHATTAN COMMUNITY COLLEGE
THE CITY UNIVERSITY OF NEW YORK

Data Analysis on NYC Vehicle Collision

Honors Project

CSC 330 – Data Structure I

Fall 2015

MENTOR: Ching-Song Don Wei

STUDENT: Hengqi Zhu

1. Introduction

According to NYPD Motor Vehicle Collisions dataset (NYC OpenData, 2015), there are hundreds of vehicle collision incidents reported weekly in the five boroughs of New York City. Approximately 12% of the collision incidents result in death or injury. It would be very helpful in reducing collisions if the motorists could be alerted when driving through the dangerous areas or intersections based on how often and how many sever collisions happen in the surrounding area. Although the NYPD Motor Vehicle Collision dataset, which is a huge spreadsheet with more than700,000 lines of records, contains enough information to determine if an area is dangerous in traffic, this flat table is very hard to comprehend and utilize without proper processing and analysis. In order to help us understand how the collision incidents are distributed across the city and identify the dangerous hot spots, the ArcGIS displays collision locations on a map that provides geographical understanding with better visual presentation. ArcGIS (Gorr & Kurland, 2011) is a professional and powerful geographic information system (GIS) published by ESRI. There are many GIS software tools that can be used. The reasons that ArcGis is selected are: firstly, it provides hundreds of tools that can be used to read, edit, optimize, and analyze geographic data or a map; secondly, ArcGIS allows users to export its geoprocessing models into Python scripts; last, ESRI offers free license to academic institutions such as BMCC. Since new collision data is added every week, a Python program was developed to download the data automatically.

By applying the ArcGIS analysis tools, a map of New York City (Department of City Planning, 2015) is marked with a layer of different colors to show the levels of danger based on the analysis results of collision data. Then, this map can be posted on websites, apps, and GPS. As a result, when people are driving in NYC, their smartphones or GPS are able to inform them whether the area where they are is dangerous or not.

2. Pre-Processing: downloading and cleaning data

The New York Collision data is available on the website at cityofnewyork.us. However, as mentioned before, the data gets updated every week. In order to obtain the new data automatically, a Python script was created to download the data. Python is a programming language which has a simple syntax, a large API library and powerful script ability. By setting a task in the Windows Task Scheduler, we are able to make use of the Python script for downloading the data, at 8:00 a.m. on every Tuesday, automatically. When the data is available on a local computer, it needs a data cleaning process. “Data cleaning” means removing some useless data or incomplete data. For example, the makers of vehicles can be removed from the dataset because they don’t actually contribute to the cause of collision. Also, some records that have important data missing, such as longitude or latitude which determines where collisions happened, can be removed as well. A Python program is generated to clean up the data. The Appendix contains the source code.

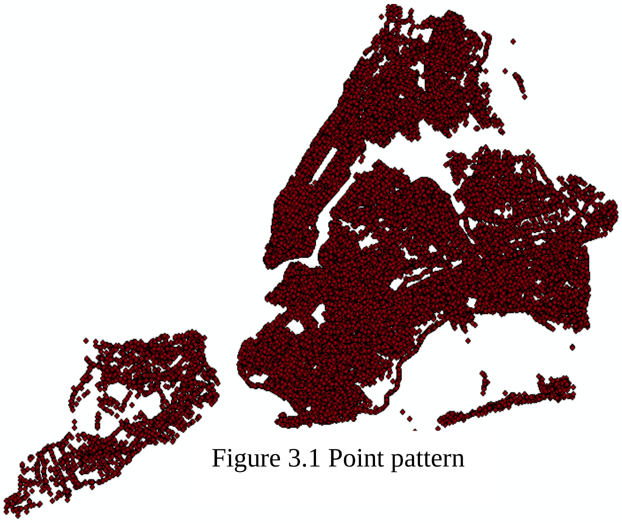

Figure 3.1 Point pattern

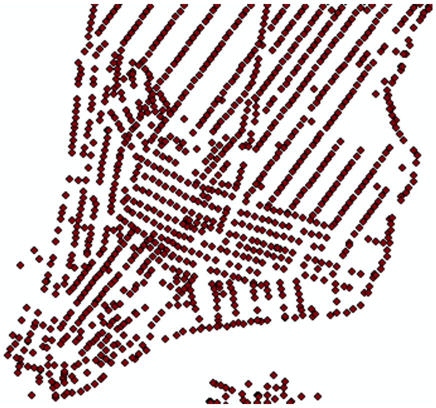

Figure 3.2 Zoom in Figure 3.1

3. Data Processing stage

When data cleaning is completed, ArcGIS is used to analyze the data; this is the most important stage. There is a tool named “make XY event layer,” which imports and transforms each collision incident’s longitude and latitude into X and Y coordinates of a map, and then displays collisions by points that are shown in Figure 3.1 and Figure 3.2. After importing collision data, each collision incident is displayed as a point. ArcGIS saves them as a new map layer; yet the new layer cannot be analyzed since it lacks OID (Object ID) for each collision. Thus, the “save to layer file” tool is employed to add OID and convert the new layer to a layer file (.lyr), which is a format of geographic data. Now, the new created layer file of collision data can be processed by ArcGIS.

ArcGIS has almost 100 tools to perform data analysis. In this project, “point density” is chosen because all collisions are displayed as points and density is the most appropriate property to show the quantities of collisions in a specific area. The “point density” tool calculates the density of point features around each cell. Conceptually, a neighborhood is defined around each raster cell center, and the number of points that fall within the neighborhood is totaled and divided by the area of the neighborhood. For example, let’s treat one classroom on the 9th floor of the Fiterman Hall as a cell and one student as a point; we can get the point density of different areas on the 9th floor through calculating the number of students divided by the number of classrooms. If there are 30 students having class in the F904, the point density of F904 is $\frac{30(students)}{1(classroom)}$; if there are 21 students in the study room, the point density of the study room is $\frac{21\ (students)}{3\ (classroom)}$ (the size of the study room is 3 times that of a classroom); and if there are 4 students in the bathroom, the point density is $\frac{4(students)}{1/2(classroom)}$.

Furthermore, the "point density" tool accepts a parameter, called "population field," which considers a value of a point into calculation. For example, when a collision happens, some people may get injured. While the number of injured people is involved in calculation, the related point density will vary. Generally, the more people get injured, the higher the point density will be. After the calculation, the "point density" tool will label each cell by different colors according to density range. For instance, in Figure 3.3, red represents 1.3 million (this is a density, not a quantity) to 1.7 million, yellow represents 0.58 million to

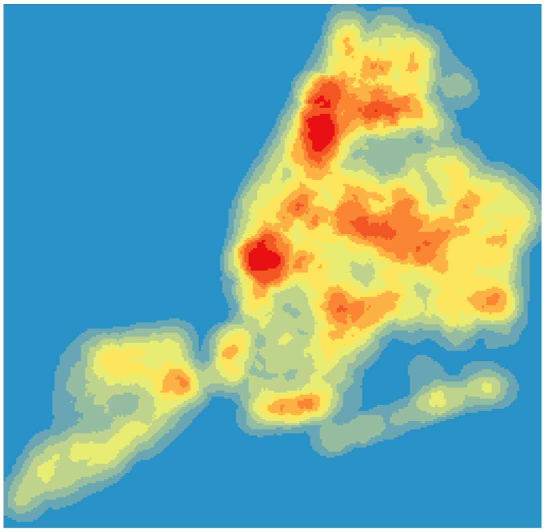

Figure 3.3 Cells are labeled by different colors

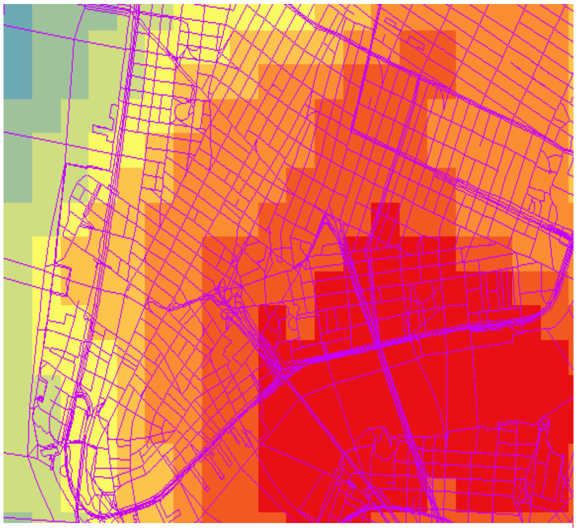

Figure 3.4 Zoom in Figure 3.3 and combined with NYC street map

0.72 million, and light blue represents 0.04 million to 0.16 million. This layer file is the final result of using the "point density" tool. The darker color means higher density, representing more collisions happened, and therefore a more dangerous area. We can combine it with the New York City standard street map in Figure 3.4 to show where the red (dangerous) areas belong to.

4. Last stage: making everything automatic

First, a function in ArcGIS called "model builder" is used to build a geoprocessing model by loading all ingredients and tools (Zandbergen, 2013). Then, the "model builder" will run this model to check whether it can be completed or not. If it fails, the "model builder" will display an error message which encourages the user to troubleshoot; if it succeeds, a layer file shown below will be generated. With several attempts, the best model is identified, which is shown as a flow chart in Figure 4.1. The flow chart demonstrates

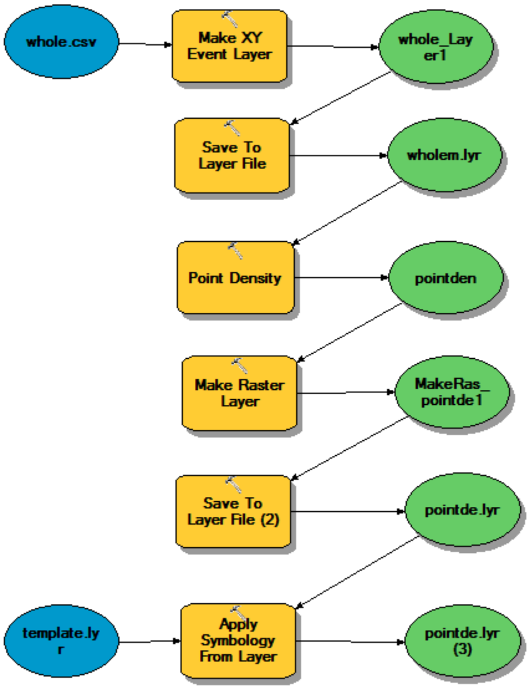

Figure 4.1 Flow chart of model

that the “save to layer” and the “point density” tools are loaded and executed by the “model builder” (the “make raster layer” and the “apply symbology from layer” tools, not mentioned here, were used in early development stages). The reason is that if the “point density” or some other tools run as a model or a Python script, the output file will be stored in memory, which means it is only temporary. When the “point density” finishes, the output file will be deleted. As a result, these two tools are used to keep all output files on the disk permanently. After the most feasible model is identified, we are able to export it to a Python script. Like other scripts, it can be executed by operation systems automatically to create a new “point density” map when updated collision data is available. (GISHelper, 2015) (ESRI, 2015).

5. Summary

Through this project I learned a great deal. Firstly, I’m glad to have worked with Professor Wei, who is humorous, kind, and extremely professional in geographic information systems and databases. He led me to the path of becoming a data scientist, a hot and new-born field. Secondly, I learned how to use ArcGIS. However, what I grasped in this project is the tip of the iceberg, so I need to study more to master this powerful geographic software to accomplish more projects. From this project I also realized that I need to strengthen my skills in statistics. Most database and data-analysis tools are based on statistics. Therefore, in order to reinforce my knowledge of statistics, I have enrolled in MAT 209 for next semester. Last but not least, I chose computer science as my major because of its financial prospects and more job opportunities at first. But now, I agree with what Professor Wei says: “We ought to focus more on how technologies serve people, not money.” I hope this project will help people to drive more safely.

References

Department of City Planning. (2015, December 3). *BYTES of the BIG APPLE*. Retrieved from NYC: http://www.nyc.gov/html/dcp/html/bytes/applbyte.shtml

ESRI. (2015, December 3). |*ArcGIS Resource*. Retrieved from ArcGIS.com: http://resources.arcgis.com/en/help/

GISHelper. (2015, December 3). *GISHelper YouTube*. Retrieved from YouTube: https://www.youtube.com/user/GISHelper

Gorr, W. L., & Kurland, K. S. (2011). *GIS Tutorial 1: Basic Workbook Fourth Edition.* Esri Press.

NYC OpenData. (2015, December 3). *NYPD Motor Vehicle Collisions*. Retrieved from City Record Online: https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

Zandbergen, P. A. (2013). *Python Scripting for ArcGIS.* Esri Press.

Appendix:

1. Python code (download data):

```
 import os,urllib2,urllib

 path='D:\DownLoad'
 file_name='collision.csv'
 dest_dir=os.path.join(path,file_name)

 url='https://data.cityofnewyork.us/api/views/h9ginx95/rows
.csv?accessType=DOWNLOAD'
 def downLoadPicFromURL(dest_dir,URL):
         try:
           urllib.urlretrieve(url , dest_dir)
         except:
           print '\tError retrieving the URL:', dest_dir
 downLoadPicFromURL(dest_dir,url)
```

2. Python code (clean data):

```
 import csv
 import os

 #delete old file
 os.remove("D:\Download\collision1.csv")

 #create new file
 f=open("D:\Download\collision1.csv",'w')
 f.close()

 #open a csv file
 with open("D:\Download\collision.csv","rb") as input:
     reader= csv.reader( source )
     with open("D:\Download\collision1.csv","wb") as
output:
         writer= csv.writer( output )
         for r in reader:
             wtr.writerow( (r[0], r[1], r[3], r[4], r[5],
r[6], r[7], r[8], r[9], r[10]) )
```

3. Python code (analyze):

```
 import arcpy

 # Check licenses
 arcpy.CheckOutExtension("spatial")

 # Local variables (need to be changed according to
different computers)
 whole_csv = "D:\\gisproject\\data_whole\\whole.csv"
 template_lyr = "D:\\gisproject\\template.lyr"
 whole_Layer1 = "whole_Layer1"
 wholem_lyr = "D:\\gisproject\\wholem.lyr"
 pointden = "D:\\gisproject\\pointden"
 MakeRas_pointde1 = "MakeRas_pointde1"
 pointde_lyr = "D:\\gisproject\\pointde.lyr"

 # Process: Make XY Event Layer
 arcpy.MakeXYEventLayer_management(whole_csv, "longitude",
"latitude", whole_Layer1, "", "")

 # Process: Save To Layer File
 arcpy.SaveToLayerFile_management(whole_Layer1, wholem_lyr,
"ABSOLUTE", "CURRENT")

 # Process: Point Density
 arcpy.gp.PointDensity_sa(wholem_lyr, "NONE", pointden,
".0016519576", "Circle 1.37663133333334E-02 MAP",
"SQUARE_MAP_UNITS")

 # Process: Make Raster Layer
 arcpy.MakeRasterLayer_management(pointden,
MakeRas_pointde1, "", "-74.253530914411 40.498270085589 -
73.700125118411 40.912911443189", "")

 # Process: Save To Layer File (2)
 arcpy.SaveToLayerFile_management(MakeRas_pointde1,
pointde_lyr, "ABSOLUTE", "CURRENT")

 # Process: Apply Symbology From Layer
 arcpy.ApplySymbologyFromLayer_management(pointde_lyr,
template_lyr)
```