Mapping Accident Prone Regions

Lauren Hearn

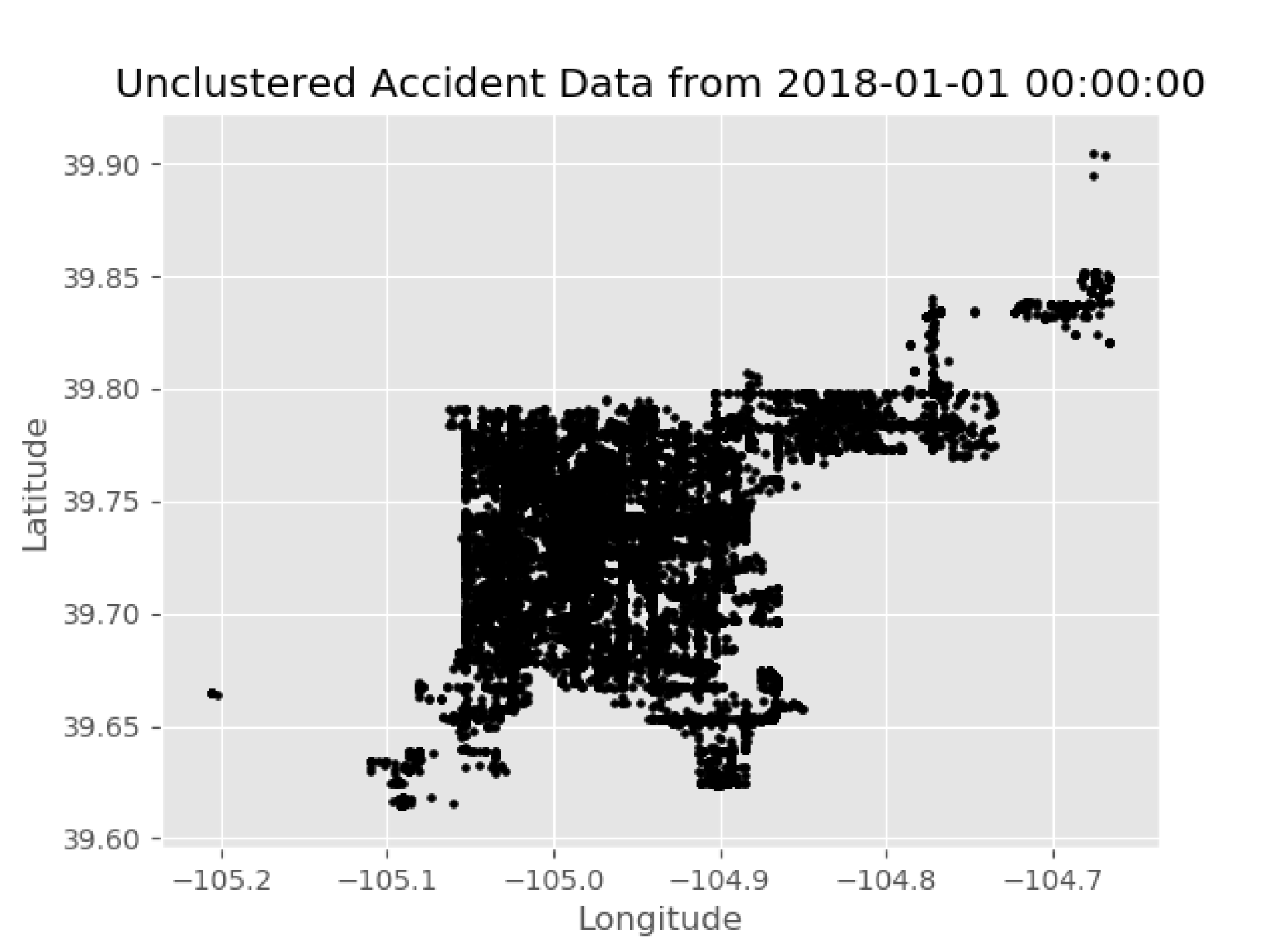
Department of Mathematical & Statistical Sciences

University of Colorado Denver
College of Liberal Arts and Sciences

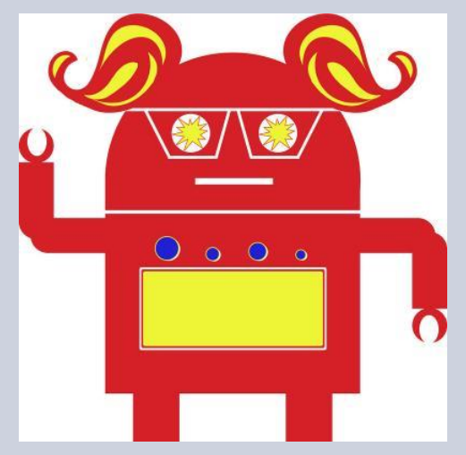
Objectives

Clustering methods are an important tool when analyzing traffic accidents, as these methods are able to identify groups of road users and segments which could be suitable targets for countermeasures. Ideally, cluster analysis is a statistical technique that can be used to group items together on the basis of similarities or dissimilarities.[1]

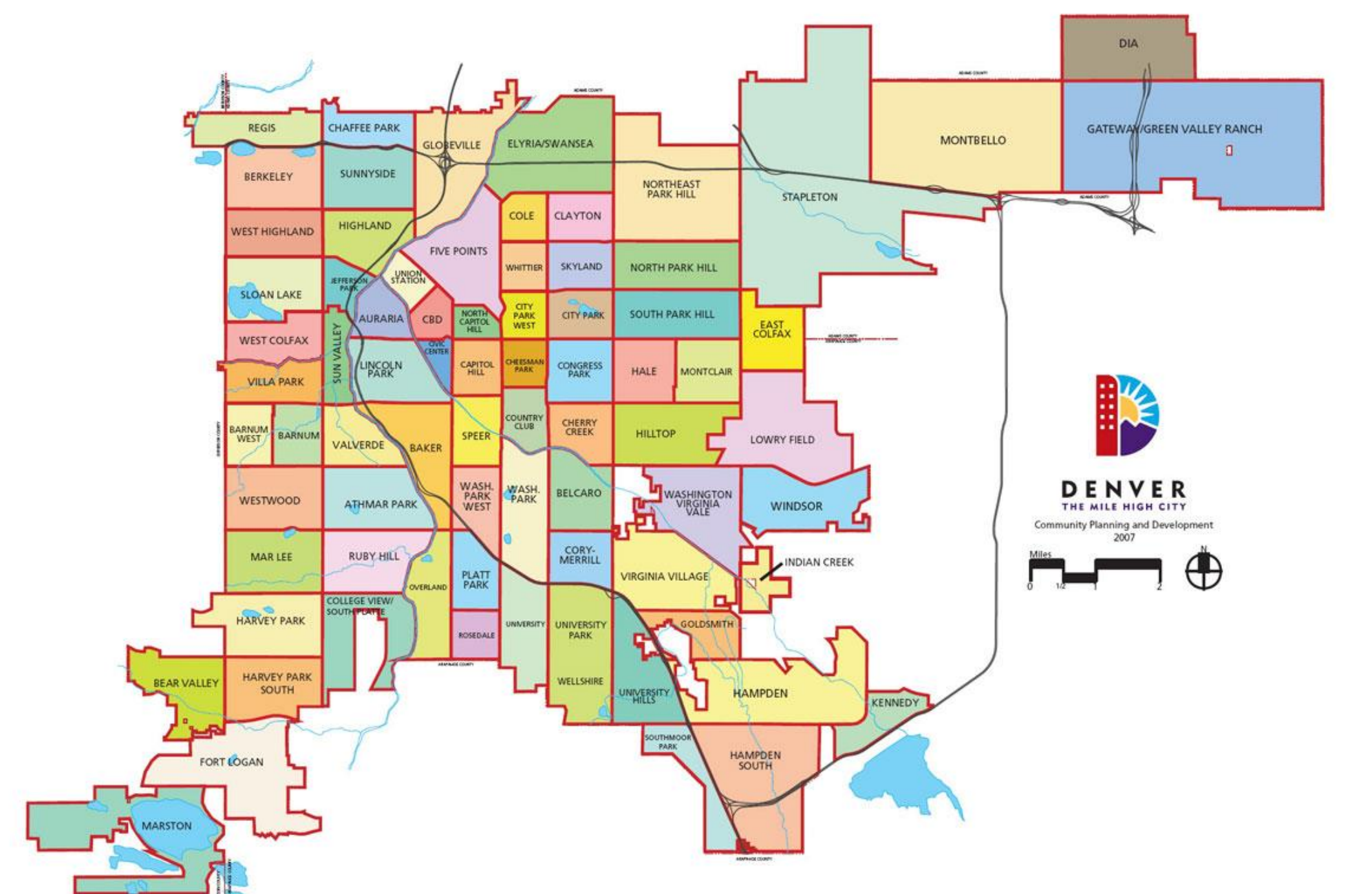
The goal of this project is to better understand what areas are especially prone to accidents, and why. Analysing and comparing this data along with other factors, including one-way roadways, traffic signals, and high-use thoroughfares should supply a pathway for city administrators to better plan updates to infrastructure. In addition, this research could assist the police force in location-based scheduling.



Contact



RobotLauren.com
Github.com/mandlebrot
Lauren@RobotLauren.com

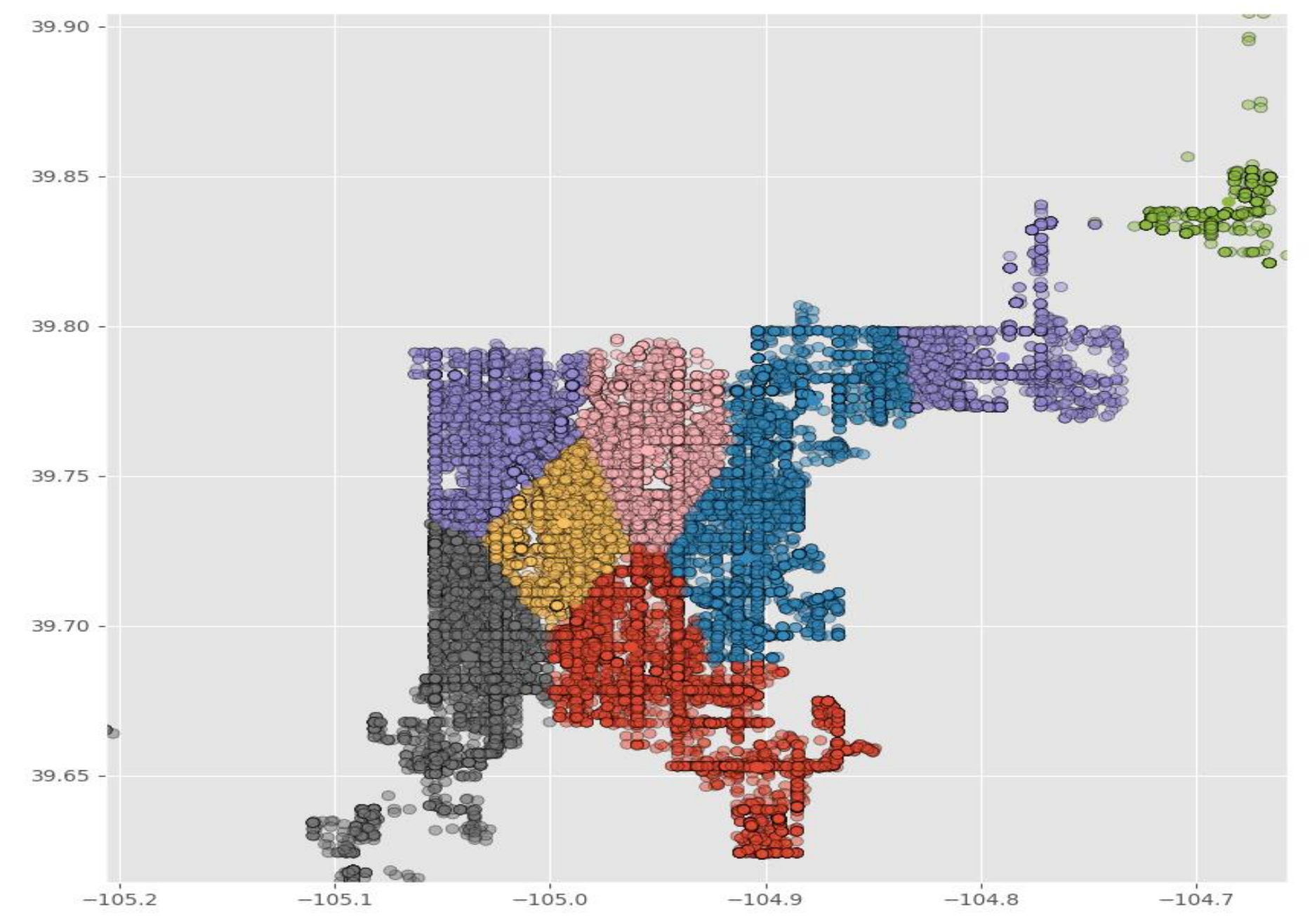


Denver Neighborhood map

Methods

This project began using a common clustering method, K-Means[2][3] (right), to explore accident data in Denver. The method splits the data into groups of equal geographic area. However, it is more helpful to modify the approach, clustering the data into groups with an equal number of data points[4]. In this way, we can look at "accident densities".

Accident data clustered with K-Means Algorithm



Tools

Using the "Traffic Accidents" dataset from the Denver Open Data Catalog and common optimization tools in Python (version 2.7), [6] a linear program was written to assist in looking for various "hot-spots" of traffic accidents. In order to accomplish this, the data was partitioned into clusters using geolocation data and cluster size bounds to compare the density of accidents per partition.
The Pyomo[8][9] optimization tools package for python was then used to code the model. Like AMPL[5], Pyomo does not come with native solvers, so the glpk[10] solver was chosen, predominately due to its being opensource as well as utilized in many online examples of Pyomo models.

References

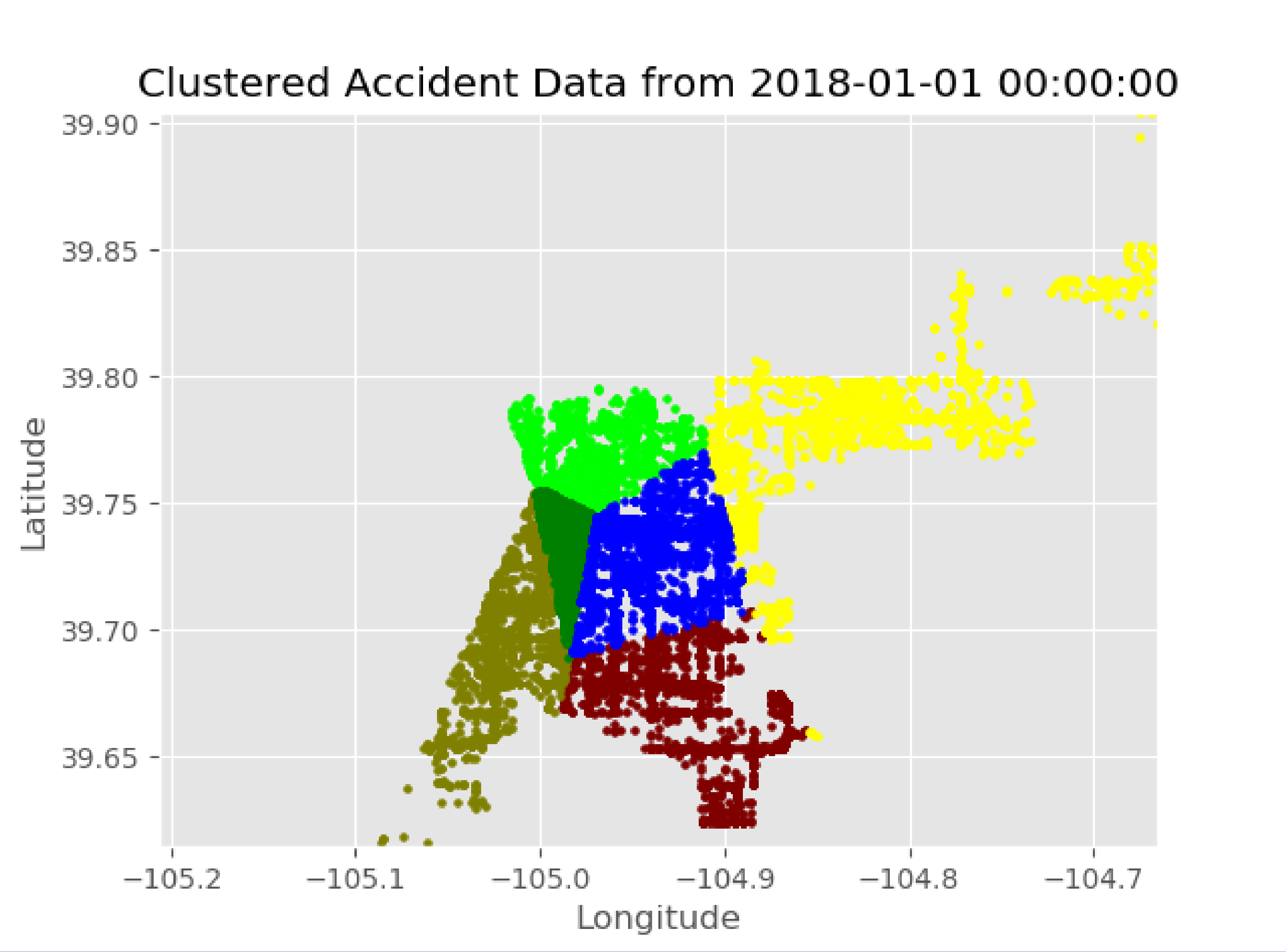
1. K.Geurts & G.Wets & T.Brijs & K.Vanhoof. (2010). Clustering and profiling traffic roads by means of accident data.
2. https://mubaris.com/posts/kmeans-clustering/
3. http://benalexkeen.com/k-means-clustering-in-python/
4. S.Borgwardt & A.Brieden & P.Gritzmann. (2017). An LP-based k-means algorithm for balancing weighted point sets. *European Journal of Operational Research*. 263-2: 349-355
5. https://ampl.com
6. https://www.python.org/
7. https://pypi.org/project/amplpy/
8. W.Hart & J.Watson & D.Woodruff. (2011). Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation 3*. 3: 219-260.
9. W.Hart & C.Laird & J.Watson & D.Woodruff & G.Hackebeil & B.Nicholson & J.Siirola. (2017). Pyomo – Optimization Modeling in Python. *Springer*.
10. https://www.gnu.org/software/glpk/

Results

Clear results were attainable by limiting data to smaller time periods.

Cursory visual analysis reveals an unsurprising trend. We see higher densities of accidents in centrally-located downtown neighborhoods, while seeing more spread-out clusters (lower accident density) as you move further from the city-center.

Future plans involve "zooming in" on data to get a clear picture of accident clustering at specific intersections.



Impact & Future Work

- Analyze and compare this data along with other factors, including one-way roadways, traffic signals, and high-use thoroughfares
- Identify suitable targets for countermeasures
- Improve traffic flow patterns
- Assist in advising updates to infrastructure
- Assist in advising better scheduling as it pertains to police presence