Many-to-One Commute Time Visualization for Public Transit Cities

Lamya Alowain* Devin Tark†

University of Illinois at Urbana-Champaign

Abstract

This paper presents a tool that produces a map visualization reflecting commute times to a given destination point using an isopleth map. Understanding the overall public transit commute times is challenging as it varies across areas according to different destinations and is not a linear function of straight line distance due to fixed and sometimes sparse transit routes. We show how we generate the commute times data and how this data can be reflected on a map to allow users to visualize the commute time distribution within a bounded area. We also discuss impact of different colormaps and underlying map types to the interpretation and understanding of the visualization.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Isopleth maps; Human-centered computing—Visualization—Visualizing commute times

1 Introduction

Developing fair, sustainable, and efficient transportation systems is registering high interest from multiple areas of studies. Transit times represent the time distance between two locations within an area according to the mode of transport. This transit time information reveals vital signs useful for users or even researchers in urban planning and development. To easily infer the most valuable conclusions of that information and their relationships across a large scale, proper visualization of this data must be designed. Public transportation commute times are a particular challenge in terms of determining and visualizing their relations to different locations across a wide area. This is because public transit commute times are not linear between two different locations, and transit routes are sparse and mostly fixed according to the city transportation system.

This paper illustrates a tool that visualizes public transit commute times across a large area as an isopleth map exhibiting what areas are near or far to a particular destination regarding public transit commute times. The paper explains the design of the isopleth map and elaborates the choices for visualizing public transit commute times. The scope of this work is focused on New York City public transportation data.

2 Motivation

One of the major considerations people make when searching for a home or an apartment is the commute time to their work. In some areas, where personal vehicles are the main mode of transport, one can reasonably approximate commute time by looking at a map, mentally drawing a circle around their commute destination, and sizing it appropriately based on known motor vehicle travel times, and restricting their search to that area (after filtering for other factors). In big cities with extensive public transport systems where driving is less common, this mental math can be more complicated. Due to the shape of existing transit architecture (i.e. fixed subway lines, bus routes, etc.), one's commute time follows a different function, one in which two areas might have different commute times despite being the same straight line distance from the destination, or in which even a geographically closer area may have a longer commute time than a farther one, due to the availability of a direct transit line.

One of the authors of this report was recently searching for an apartment in New York City, and found having to do a new Google Directions query for every new apartment looked-at repetitive and cumbersome, due to this non-linear function described. The motivation for this project was to simplify this by allowing an apartment hunter to visualize a continuous isopleth map of New York City according to the public transit commute time to his or her workplace. This can help narrow down neighborhoods in which to search for an apartment.

Transit based isopleth maps like this one can also be used from a broader public policy standpoint to draw attention to "transit deserts" or communities within cities that are under-served by public transit.

3 Tool Components

3.1 Constructing the Grid

Constructing the grid was helpfully made simpler by New York City's making available a shapefile of its boundaries. This shapefile was loaded as a GeoDataFrame, a data structure provided by the geopandas python library. We can plot this GeoDataframe as shown in Figure 2.

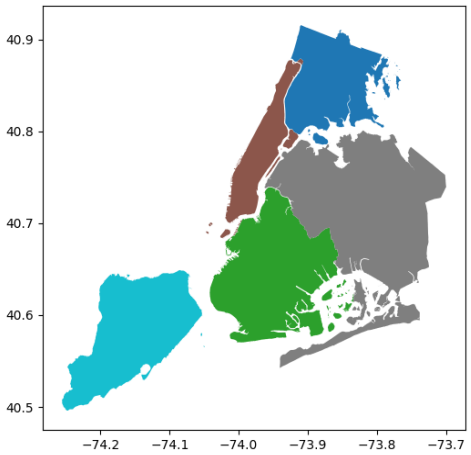

Figure 1: New York City shapefile plotted by borough.

Next, we identified the bounding box longitude and latitude coordinates for the area we wanted to build the map. This included the complete boroughs of Manhattan, Bronx, Brooklyn, and Queens. We then created a mesh grid of every longitude and latitude in a 140x140 grid. This created spacing of about .0025 degrees in both

*e-mail: alowain2@illinois.edu
†e-mail: devinvt2@illinois.edu

directions, or about 2 square city blocks. Then, we discarded all points that we're not on land or within a borough limit defined by the shape file. Next, we constructed polygons for every grid cell of coordinates. Because New York City spans 3 islands, this grid included some water ways, which will be taken care of in the filtering step. For every polygon, we checked that the centroid of the polygon was within the borough boundaries defined by our GeoDataFrame (i.e., on land), and if not, it was discarded. We write out this geometry along with each polygon's centroid coordinates to a GeoJSON file, which will provide compatibility with the Leaflet.js library we use for mapping on the front end, along with many other mapping technologies.

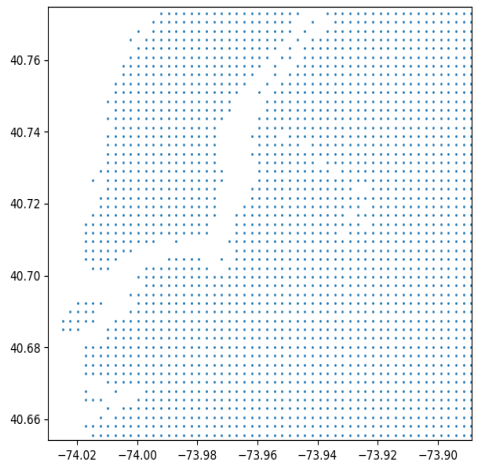

Figure 2: Filtered Grid Created for Polygon Production. (Williamsburg / Lower Manhattan)

3.2 Third-Party API

We had two options for calculating commute times. One was to consume a set of files in a standardized format called General Transit Feed Specification, or to call a directions API, like Google's, to get the commute time for every centroid on our polygon grid to a destination point. While using the API likely takes much longer than an optimized GTFS approach, we decided it was better to abstract away the route planning part of this project and focus on the visualization. This is an area for future improvement. We used the HERE public transit routing API ("/transit.router.hereapi.com/v8/routes") to get a public transit route for each of our grid cells. The API can calculate mixed-mode transport for anywhere in the world. Being that there are over 10,000 grid cells, the entire process takes about 35 minutes and this is likely bound to the http requests.

3.3 Calculating & Coloring Commute Times

This API returns the walking route, subway or bus line and station, as well as the elapsed time for each leg of the trip. Upon receipt of the response from the API, we add up the elapsed time of each leg, which includes a mix of walking and public transit, and return the commute time in seconds.

Algorithm 1: Determining a Polygon's Color

```
Data: Commute Time in Seconds
Result: Color Value in Hex from Colormap
colormap[240] = Named Color Map;
if time = -1 then
    return #000000;
else if time > 7200 then
    return colormap[239];
else
    return colormap[commuteTime / 30];
end
```

As shown in Algorithm 1, we segment our colormap into 240 discrete buckets. This is the max allowed by our color mapping library that provides these to us. If our API caller returns a -1 as commute time, this indicates an error in the HTTP request and we simply color the polygon black. These seem to happen only a handful of times in our tests, mostly if the polygon was on a waterway or highway. If commute time is greater than 2 hours, or 7200 seconds, we color the polygon the lightest color. We decided on this cut off for two reasons. First, there aren't many routes in New York City that take more than two hours. And secondly, being that most commutes are about 10 - 60 minutes, this would convey enough information for the average user. If the commute is between 0 and 7200 seconds, we do an integer divide on the time and color it according to the index in the color map array.

That being said, if we *do* know that a particular commute will be reachable from almost anywhere within a shorter amount of time, by performing some analysis first on the entire set of commute times, we can improve our visualization even more. We will discuss in our results section the improvements made by scaling down our buckets even smaller, if we can assume good coverage below say, 5400 seconds, instead of 7200. This can be achieved from either finding the max commute time, or perhaps cutting off after some standard deviation.

3.4 Visualization

The purpose of the tool is to illustrate the overall public transit commute times from all points within an area to a particular destination within the same area. With decent visualization, we aim to reflect how those commute times distribute around the entire area. Given that the data indicates location-related information, we use an isopleth map to engage the users in visualizing the distribution of commute times. The map will ultimately show the user what sub-areas are close in time to the destination when using public transportation. We approach with isopleth map to show the pattern of intensities revealing the commute times. As illustrated earlier, the points represented by the grid are fixed locations which may have different values according to a destination point. Each point on the grid will have a value ranging from a minimum value of 0 seconds to a maximum of 7200 seconds (i.e. 2 hours) of commute time.

The choice of colormap affects the perception of the data. Since each point represents a value of commute time scaling from a minimum time to a maximum time, we choose a perceptually uniform sequential colormap. Sequential colormaps display color brightness transitions incrementally to indicate ordering information [3]. They can convey commute time information and assist users in recognizing the differences in intensities at different points. Perceptual uniformity of the colormap allows us to guarantee that the short spans between different time intervals of commute times will give similar appearing colors while larger time difference will result in different appearing colors, uniformly.

We consider choosing an appealing colormap that does not leave

a misleading impression on the end-user. For this reason, we avoid colormaps that are common with other Geo-related areas symbolizing other intuitions such as temperature. There is no prior standard colormap used in the area of our problem, and we tested the visulaization results with viridis and magma colormaps as they can both show the varying densities while leaving a pleasant impression on the user. One important factor we consider is the fact that the resulting colors should be slightly transparent to reveal the underlying map graphics and texts indicating the location information (e.g. streets, subways, buildings, and names of neighborhoods). This allows the user to explore the underlying features of the areas as they zoom in and navigate through the resulting isopleth map. For this reason, we compare the results of viridis and magma colormaps in terms of sustaining the color meaning with lower opacity. Viridis is a perceptually uniform purple-green-yellow colormap while magma is a black-purple-pink colormap. Both colormaps are perceptually uniform sequential colormaps, and the colors with which those colormaps are designed make them friendly to colorblindness. Liu et al. show that the viridis colormap has higher accuracy in terms of user effect using studies of response times (RT) [1]. However, there is no study evaluating how these colormaps function with lower opacity, which we found negatively affects the perception of these colormaps. In the following section we demonstrate how the results differ with our application.

4 Results

We visualize the public transit commute times in New York City for the same destination point we chose in central Manhattan using viridis and magma colormaps as described in section 3.4. Fig. 3 shows the results with viridis colormap and 0.7 opacity of colors while Fig. 4 shows the results with magma with the same opacity of 0.7. The black points on both results are due to data errors as explained in section 3.2

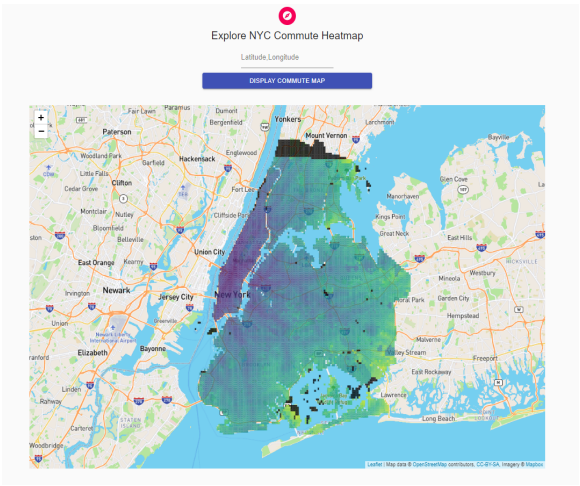

Figure 3: New York City public transit commute times isopleth map using viridis colormap

Both results show inference to densities in terms of areas close in commute times to Manhattan's destination point. In viridis results, the closer areas are more purple while the further are green. Magma gives darker purple for closer areas and yellowish shades on the areas that are far. From these results, we can see that there are scattered darker areas with higher densities, and sometimes they form a connected spread revealing part of the underlying public transportation infrastructure. The opacity slightly affected the clarity of the distribution with both colormaps. However, the concept in which darker is perceived as the closer is not affected.

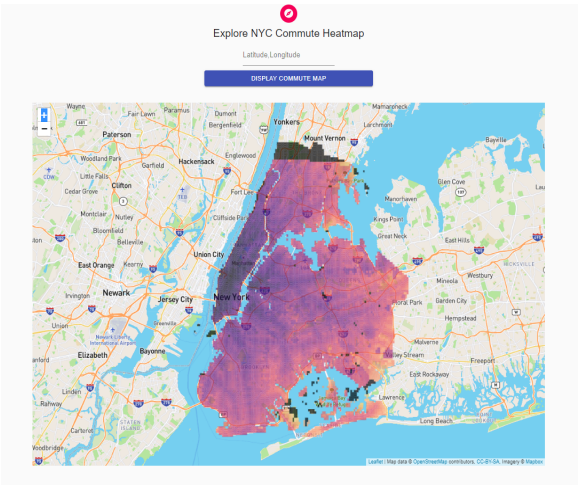

Figure 4: New York City public transit commute times isopleth map using magma colormap

It is hard to determine which colormap is more effective with the lower opacity without conducting a study on users. Finally, we conducted a test using hot colormap as seen in Fig. 5. The results show clearer distinctions with the lower opacity level.

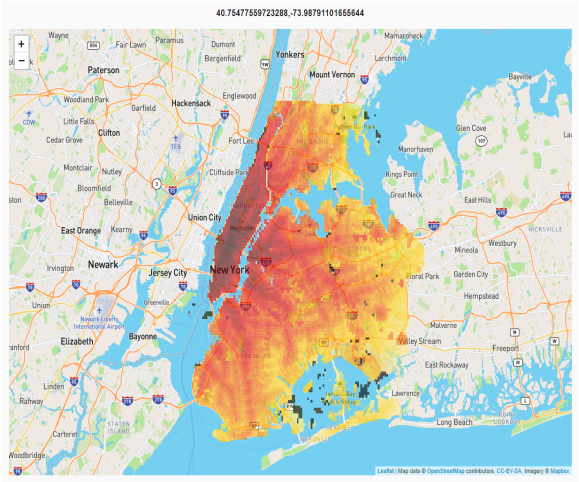

Figure 5: New York City public transit commute times isopleth map using hot colormap

Another factor that is affecting the results is the original colors of the underlying map [2]. To improve the results, we conduct a test using darker gray-scale for the underlying map and we study the effect on the results' color clarity. Removing the variation in the underlying map's color improves the visualization greatly while maintaining transparency of colors to reveal street level detail. Moreover, as mentioned in section 3.3, because we know for this particular query that most if not all commutes will be within 90 minutes, we can adjust our color mapping algorithm to scale down to 0-5400, with 240 buckets of 23 seconds each. This increases the resolution and produces much clearer maps as demonstrated by the much more distinct "transit corridors" in Figures Fig. 6, Fig. 7, Fig. 8.

From the results we find that the choice of the underlying map color plays a significant role in affecting the perception of the vi-

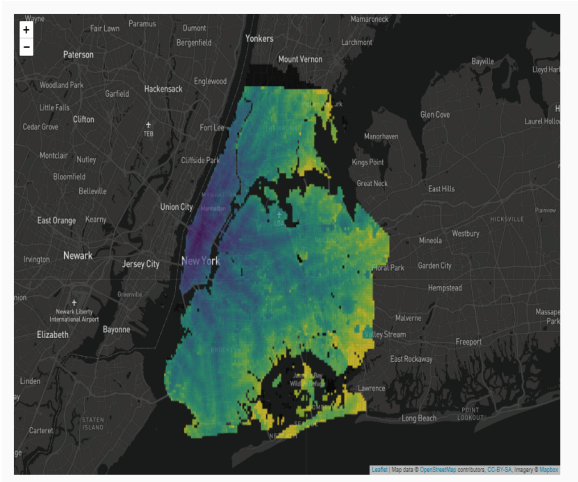

Figure 6: New York City public transit commute times isopleth map using viridis colormap, dark map, and scale of 5400 seconds

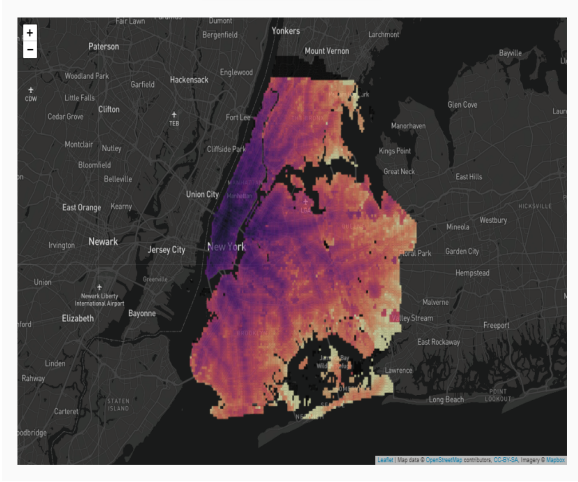

Figure 7: New York City public transit commute times isopleth map using magma colormap, dark map, and scale of 5400 seconds

sualized isopleth map. We show the different results of different colormaps and compare between them in terms of clarity and focus. We can see that hot colormap on the dark map can reveal more of the lines where there are distinctions. This can assist in previewing the overall public transportation lines across the city. On the other hand, magma shows better results for revealing the high resolution distribution of commute times, i.e., smaller changes in commute time. See how the transit corridors show more "tapering off" on magma versus hot. This better shows how commute times scale as we move away further from public transit.

5 Our code

The code for this project can be found at https://github.com/devin040/nyc_commute_map. For a brief explanation of of some of the files:

1. create_npy_preprocess.py
    - This file will create the numpy files that are created while creating the mesh/ polygon grid. If you plan to run it multiple times it saves time to store these big numpy arrays as .npy first. These take quite some time to generate
2. create_nyc_geojson.py
    - This file will use the .npy files created with the previous script to create the geojson file of polygons. This file is used by leaflet for drawing on the map.
3. construct_coordinates_no_json.py
    - This file will create and plot coordinates from the .npy files. Useful for making and viewing small adjustments to the coordinate system if needed, without creating geojson file
4. create_sample_plot_geojson.py
    - Creates a geojson file in the same way as create_nyc_geojson.py but with just a small sample (279 grid squares). For testing the API / front end.

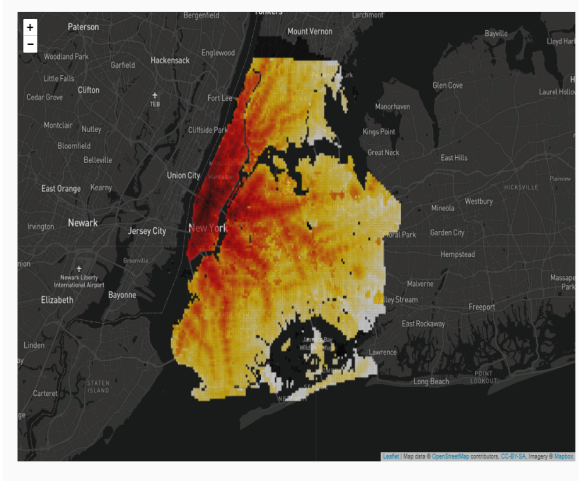

Figure 8: New York City public transit commute times isopleth map using hot colormap, dark map, and scale of 5400 seconds

References

[1] Y. Liu and J. Heer. *Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018.

[2] M. Parker, A. Silverman, A. Wang, and K. Schloss. The role of perceived opacity in interpreting colormap data visualizations, 08 2017. doi: 10.1167/17.10.1179

[3] K. M. Thyng, C. A. Greene, R. D. Hetland, H. M. Zimmerle, and S. F. DiMarco. True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography*, 29(3):9–13, 2016.