Manhattan Project

Exploring Urban Data: NYC Accessibility

Matthew Dunn, Katrina Evtimova, Michael Higgins

GitHub: http://bit.ly/1OsHass, GoogleDoc: http://bit.ly/1TfMBPF

Table of Contents

Final Report

Exploring Urban Data: NYC Accessibility

Matthew Dunn, Katrina Evtimova, Michael Higgins

Abstract:

The main focus of this project was to build a transit accessibility score for each of the 2000+ Census Tracts within the five boroughs of New York City (NYC) and relate it to Census Tract data on demographics. The transit accessibility score is an aggregated measure reflecting the time needed to travel from a given tract to all other tracts. NYC bus and train data from the NYC Metropolitan Transit Authority (MTA) are used in computing the accessibility score. Our analysis shows that usually but not always people in tracts with lower mean income tend to experience longer travel times than people in tracts with higher mean income, among other results.

Introduction

Motivation

Income inequality has been increasing across the United States. The income-gap between the wealthiest and poorest residents of NYC is the largest in the nation with the top 5 percent of households earning 88 times as much as the poorest 20 percent[1]. A primary driver of income inequality in NYC and across the country is intergenerational economic mobility, that is the ability of one generation to earn more income than their parents. Further, it has been found that a primary driver of intergenerational economic mobility is access to reliable and efficient transportation. Essentially, the more time an individual spends commuting, the less likely it is they will escape poverty and earn more than their parents[2]. As a result, a better understanding of the access individuals have to transportation across NYC is an important area of study.

Problem

Currently, there lacks information on the time and cost of commuting in relation to census tract level demographic information. There are several visualization tools available to individuals trying to better understand their commute times, but there lacks a comprehensive score reflecting transit accessibility in relation to census tract level economic demographics. As a result, we are unable to examine how well NYC's transportation infrastructure serves its neediest citizens. A better understanding of how accessible the rest of the city is from any given census tract facilitates a better understand of how low income communities are served by the

transportation infrastructure of NYC and whether that infrastructure is helping or hindering intergenerational economic mobility.

Importance

As stated above, the amount of time an individual spends commuting in a single day has a direct impact on the intergenerational economic mobility of that person's family. Consequently, being able to better visualize the transportation equality across the city, we can better understand how the transit system is providing families the opportunity to escape poverty.

Big Data Infrastructure

Due to the fact that there were no bulk repositories of real time station-to-station data for the bus and train systems of NYC, we considered the stated schedules of each system suitable proxies for actual running times. Consequently, we didn't require any big data infrastructure to reduce and analyze these data sets, but significant optimization techniques were employed, i.e., vectorization, grid-indexing, etc., to be able to efficiently compute statistics for each transit system in relation to each census tract.

To evaluate whether residents of underserved census tracts could utilize private transit options, i.e., taxis, ubers, etc., we attempted to calculate the average cost, duration, and distance between census tracts for taxi data, and for this process we utilized Amazon Web Services (AWS) Elastic Mapreduce (EMR) framework to geocode each trip to a census tract.

Experimental Techniques and Methods

Methodology

Phase 1

During the initial phase of our project we focused our efforts on data collection, wrangling, and preparation, in addition to building a visualization framework. From the outset we had intended to merge all modes of transportation (bike,train,bus, taxi) and compute the optimal route (and travel time) between each pair of locations. But, during the process of mapping and reducing bike and taxi data we came across a number of obstacles which ultimately lead us to restrict our computation to the optimal routes of bus and train.

The initial problem we confronted when analyzing the CitiBike data was that in a large percentage of cases there were an inadequate number of trips to reliably measure the average time between each pair of stations, see Figure 2. In hindsight, it is clear that in order to accurately calculate the trip durations we would need to bring in directional street data and compute the estimated bike time based upon average biking speeds.

Figure 2

In addition to believing averages computed between CitiBike stations may be inaccurate, we concluded that it was unreasonable for us to approximate the travel time between census tracts using CitiBike data because most commuters don't leverage this form of transportation on a daily basis and we believed the commuters that use CitiBike for transportation do not use it in a intermodal manner. Consequently, we choose to exclude CitiBike data because we believe it currently isn't a mass transit option. Similar to the Citibike data, we decided the taxi data would not be useful in understanding the transportation access across census tracts as few people in low-income communities would utilize taxis or ubers for their daily commuting. Thus, we focused our efforts on understanding transportation access via bus and train data.

Phase 2

With the second phase of our project we shifted our focus onto bus, train, and census data. We had already prototyped our visualization framework, so now all of our efforts were on collecting, wrangling, and building credible estimates of transportation access using train and bus data. As stated previously, due to the fact that there was no bulk real-time data sets available for the bus and train systems we built our travel time estimate using schedule data which we consider suitable proxies for actual running times.

Once we had collected the bus and train system schedule data we had to extract the start and stop times from the schedules. We fully detail this process later in the paper Results and Discussion section. After overcoming the challenge of extracting the necessary data from the train and bus schedules and mapping all the bus and train stations, we had to address the issue that many stations are within walking distance of one another. Consequently, we had to merge stations within walking distance together due to the reality that commuters transfer from one mode to another when stations are in close proximity.

As a result, we opted to consider bus and train stations in close proximity to each other stations along the same route. Because of this reality and that there were no data sets available on the travel time between stations in close proximity to each other, we created a simple one-pass heuristic to reduce the number of stops. The algorithm works as follows:

> For every stop round the lat/long to 3 significant digits; this is roughly equivalent to the grid index. When we perform a search for neighbors to a given station it will only be within this rough lat/long group. The algorithm requires a cluster radius r. Start at the beginning of the dataset, looping through each point and assigning itself and all if its neighbors (within radius) that have not been assigned yet to a cluster. These new clusters will be referred to as reducedId's for the remainder of the paper. This reduced our number of stations from 14,201 to 5,462. We computed the center of each cluster by taking the average lat/lon for all stations in the cluster.

We then replaced all of our MTA station Id's with the reducedIds and the result can be seen in Figure 3.

Figure 3

With this much smaller dataset of stations we used an algorithm from the NetworkX library called "all_pairs_shortest_path_length" to compute the time between every pair of reducedId's using the optimal route. We also picked several starting locations and computed the optimal path in terms of time from these locations to every other location using "single_source_dijkstra_path" algorithm that was apart of the NetworkX library. As the name suggests, given a weighted graph and an origin node, the Dijkstra algorithm determines the shortest path between the origin node and all other nodes. We used the

“single_source_dijkstra_path” algorithm to visualize the transit time between the reduced number of stations as shown in Figure 4 and the “all_pairs_shortest_path_length” to compute transit accessibility scores between the reducedIds.

When we attempted to visualize all of the possible paths between reducedIds we found that excluding the merged paths actually improved the legibility of the map. This was do to the fact that if a bus takes a meandering path between two stations it leaves the impression that all points along that meandering path are accessible. By pretending that all paths between stations were straight we mitigated this effect, reduced the complexity or our code and the loading time of data onto the d3.js visualization web page.

Figure 4 - Note from the close up it is apparent that although multiple routes to every location is possible, only the one that is optimal in terms of travel time is displayed.

Phase 3

The third and final stage of our project focused on developing an access score with the prepared data we worked on previously. Specifically, with the duration calculated between the reduced number of stations, we set about geocoding the reduced stations to census tracts so we could analyze an accessibility score in relation to census tract statistics. In defining our accessibility score we thought the most natural measure would be the average velocity. We believed this because average time is a poor metric as it will be skewed to low values when the tracts are dense.

Once we had geocoded the reduced stations to the census tract level we were able to calculate accessibility scores at census tract level and analyze these scores in relation to income data. We detail all analyses in the Results and Discussion section. One outcome of our analyses was a desire to analyze the cost of alternative transportation by census tract, so very late in the project we returned to the mapreduce scripts we’d drafted earlier in the project for taxi

data and began the process of geocoding taxi trips to census tracts. The trials and tribulations we encountered when attempting to deploy python scripts to AWS EMR are fully detailed in the Results and Discussion section, but a brief overview follows. Initially, our mapper implementation for the taxi data was suboptimal. It iterated through all the tracts for each of the trips which resulted in a lot of unnecessary checks. Inspired by the lecture and lab on spatio-temporal data, we ended up building a more efficient mapper using the Python modules shapely and geoindex, but we found the process of bootstrapping AWS clusters, poorly documented and challenging to debug.

Tools

In addition to the standard scientific computing libraries, i.e., Numpy, Pandas, Scipy, etc., we utilized some specialized libraries to accomplish project specific tasks. Most notably to efficiently geocode all of the train stations, bus stations, and taxi pickup and drop-off coordinates to census tracts we utilized a library called shapely which facilitates the analysis of geometric polygons. To geocode points efficiently we utilized the geoindex library which enabled us to minimize the number of census tract associated polygons we need to evaluate in order to geocode lat and lon points to a census tract. To compute optimal routes we used the NetworkX library. For the geocoding of taxi pickup and dropoff points, we attempted to utilize AWS EMR.

Experimental Setup

To facilitate our analysis throughout the project, we focused significant effort early in the project on building a visualization framework using d3.js that allowed us to visualize and interpret our data at each step in the process. Further, once we computed transportation access scores between each census tract, i.e., duration, distance, and velocity, our visualization framework and reproducible scripts allowed us to quickly evaluate these metrics in relation to census tract statistics quickly. To this end, early on in the project to facilitate working across multiple data sources we setup a data dictionary (Figure 5) to ensure we generated standardize formats from the raw input data.

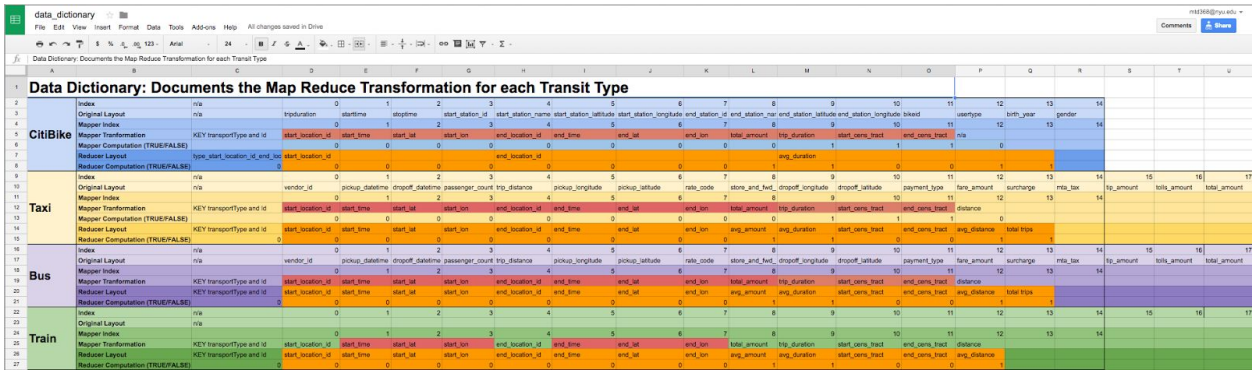


Figure 5

Data Sets

1. **MTA Bus and Train Data** - we collected public transportation data, in particular all bus and train schedules and routes available from the MTA website (http://web.mta.info/developers/).
2. **Census data** - we collected Census data at the tract level for all five boroughs of New York City. In particular, we collected statistics on mean and median income, poverty rate (fraction of the population below the poverty line), and travel time to work from the 2010-2014 American Community Survey 5-Year Estimates. We calculated the average travel time from the reported statistics on number of people within each tract with travel times falling in one of several time interval categories (an example of such a time interval category is travel time between 20 and 24 minutes). We merged together all the Census data into a single dataset at the Census tract level. (http://factfinder.census.gov/)
3. Taxi Data - Taxi data was for each day of the year and contained pickup, dropoff, duration, and fare information in our Phase 3 analysis we used a single month (~1.5GB) (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml)
4. Bike data - We collected 1.6GB of citibike data from 2015. We ultimately did not end up using it for the reasons outlined in Phase 1 (https://s3.amazonaws.com/tripdata/index.html).
5. NYC tracts shape files - downloaded from (http://www.nyc.gov/).

Cluster Configuration

We used the default AWS configuration using 3 instances (1 master and 2 core nodes) of type m3.xlarge (80GiB each). In the cluster initialization stage, we created a bootstrap action in order to install additional software (namely, Python modules that we needed for our mapper) on all nodes of the AWS cluster, which involved writing a bash script.

Results and discussion

Challenges

1. **Census Tract Geocoding** - Immediately upon endeavoring upon this project, we needed to be able to geocode bike, taxi, bus, and train data to the census tract level. We initially implemented a very inefficient algorithm to geocode lat/lon points and our geo spatio-index lecture didn't happen until much later, so we didn't realize we could change the algorithm to get significant performance improvements
2. **MTA Trip Data** - The mta bus/train schedule data came in a format that was very difficult to work with. In order to get the data in the format of startTime, stopTime, startLocation, endLocation and duration we had to split the dataframe in half, delete the first row, add an empty end row and merge it back. Each row included a "sequenceId" that told you how far along the bus route you were on. In order to separate the different bus routes we

created a dummy feature of the difference between the two sequenceId's from the split/shifted/re-merged dataframe. If this difference was not 1 then there must have been a change in routes. After using this dummy to split the data into separate trips we computed the duration of each trip. An additional issue caused by the data being in schedule format was that it was not in a typical time format. For example the time-stamp did not include this day - so we had to handle the special case of when the clock "turned over" to a new day. To address this issue we had to parse a special code in the name of the route which indicated which days of the week the route was applicable for and then track which schedule data "turned over" to a new day.

3. **MapReduce** - We faced several issues when dealing with the Yellow Taxi data in the MapReduce framework. We were able to overcome the first one which involved reading the Census tract shapefile inside the map function on AWS. The next challenge we faced was installing dependencies for our Python mapper code on the AWS cluster. We spent a lot of time researching and trying out possible solutions to the problem. We were surprised to find that a simple solution to what must be a pretty common problem was not readily available. In particular, following advice we found online, we made multiple futile attempts of passing the compressed Python modules as an additional argument in the '-files' option of adding a step to the cluster. After consulting with fellow classmates about this issue, we adopted a different approach. We wrote a bash script which is bootstrapped at the initialization of a new cluster and which installs the desired modules on all of the cluster nodes. Unfortunately, it worked only partially - for all but one of the modules we needed. This prevented us from running our more efficient mapper code on AWS, but we were able to run it locally. With the cluster configuration described earlier, the estimated runtime of our more inefficient MapReduce implementation was about 20 hours which is why we decided to terminate it before completion.
4. **Taxi Data** - As mentioned above, we were able to test out our MapReduce framework for taxi data locally on a subset of the data. In analyzing our output from the map function, we found that approximately 6% of the data was corrupted in the sense that the actual pickup or drop-off locations were missing or not valid. Our solution for this issue was to discard corrupted observations.
5. **Clustering Stations** - The MTA bus and train station data needed to be cleaned a great deal before they could be used in our analysis of optimal routes. Many pairs of stations were geographically very close but had separate id's. For example, bus stops that were across the street from each other had distinct id's. Since there was no data on travel times between these stations it was essential that we grouped them in some manner. See Phase 2 for details on our implementation of the clustering of stations.
6. **D3 Map Integration** - We made several attempts to convert our map data to a topojson format and ultimately ended up staying with geojson, partly because it was relatively small in comparison to the rest of the data that we would be loading. There was a lot of trial and error with all phases of the data/map integration, but we were able to build a working prototype that allowed us to efficiently analyze the accessibility score in relation to census tract statistics.

7. **Bikes Data** - Bike data clustered around manhattan and consequently we believed they were not a viable mode of transportation for the general population. We attempted to find optimal travel paths between stations but failed do a lack of foresight into the importance of having directional street data.
8. **Census Data** - Some of the observations were missing Census tract identifiers. Luckily, this information was contained in the Geographic ID available for each observation which allowed us to fully reconstruct the missing tract identifiers. Also, approximately 1.6% of the Census observations were missing one of the attributes of interest and we had to discard it from our analysis.

Findings

Accessibility score - using public transportation data, we derived an aggregated accessibility score for each Census tract based on the relative distance and average travel time to all other tracts. We found that our accessibility score is positively correlated with mean and median income, and negatively correlated with poverty rate and average travel time from the Census data. Although correlation does not imply causality, all of these relationships make intuitive sense - we would expect tracts with higher accessibility score to attract people with higher mean/median income.

Correlation Table between Tract Accessibility Score and Census Data

	Access Score	**Mean Income**	**Median Income**	**% Poverty**	**Avg Travel Time**
Access Score	1.000000	0.155956	0.127235	-0.029180	-0.105724
Mean Income	0.155956	1.000000	0.911385	-0.593630	-0.524174
Median Income	0.127235	0.911385	1.000000	-0.724375	-0.441823
% Poverty	-0.029180	-0.593630	-0.724375	1.000000	0.137824
Avg Travel Time	-0.105724	-0.524174	-0.441823	0.137824	1.000000

We found rather surprising relationships between our accessibility score and the Census income, poverty rate, and average travel time in terms of mutual information - all scores were above above 0.89. In the case of access_score vs. avg_travel_time we got a score of .9997 in spite of these values being calculated from completely different sources (census vs. mtaData). We used Sklearn's implementation of normalized_mutual_info_score for the calculations. Upon investigating the existing literature for a possible explanation of these results, we found a source ([3]) claiming that while high correlation is an indicator of high mutual information, low correlation does not necessarily imply low mutual information. A reason for that could be the existence of non-linear relationship between the two random variables. Please find below a D3 visualization of the relationship between the different attributes of interest in Figure 6:

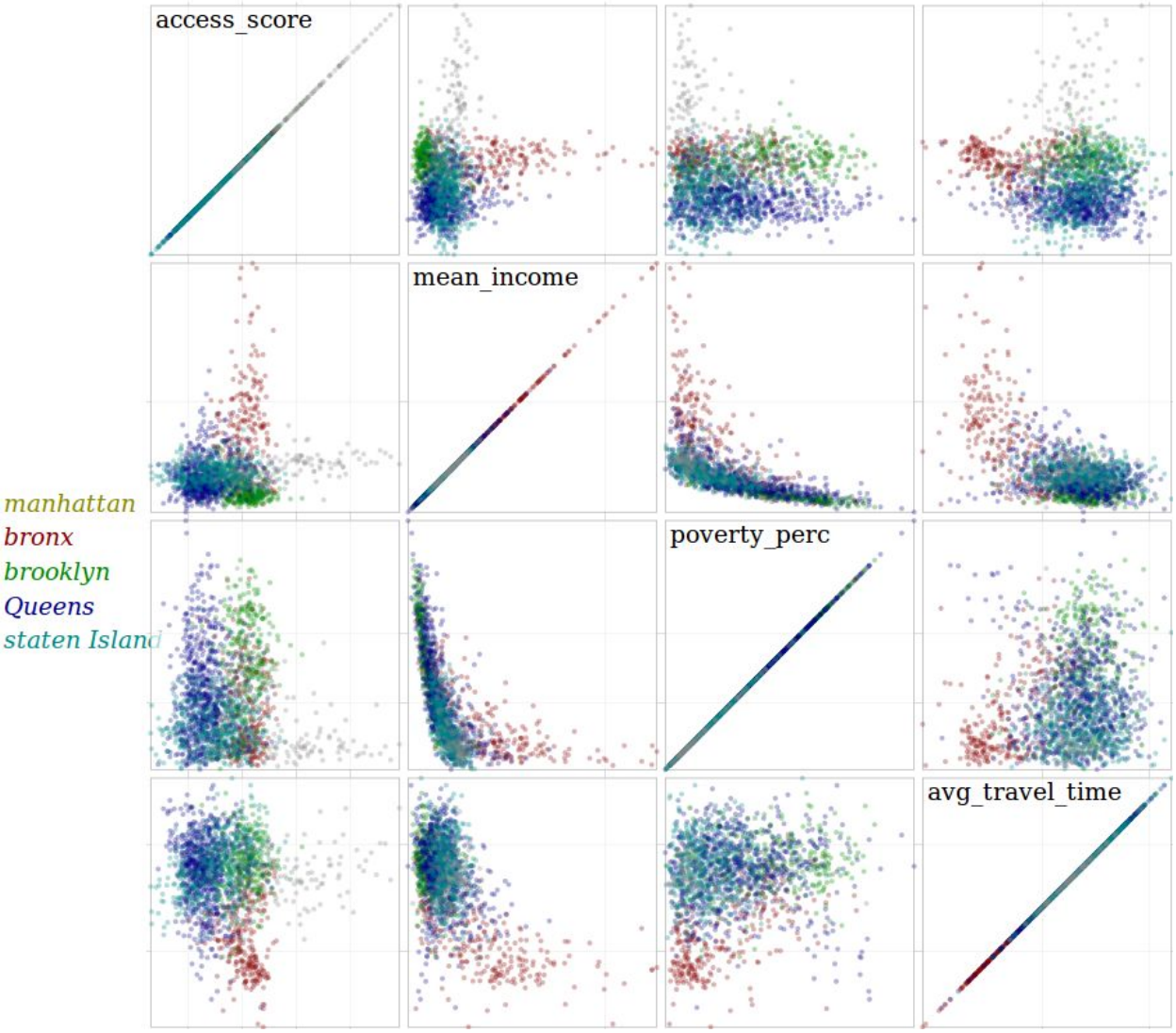
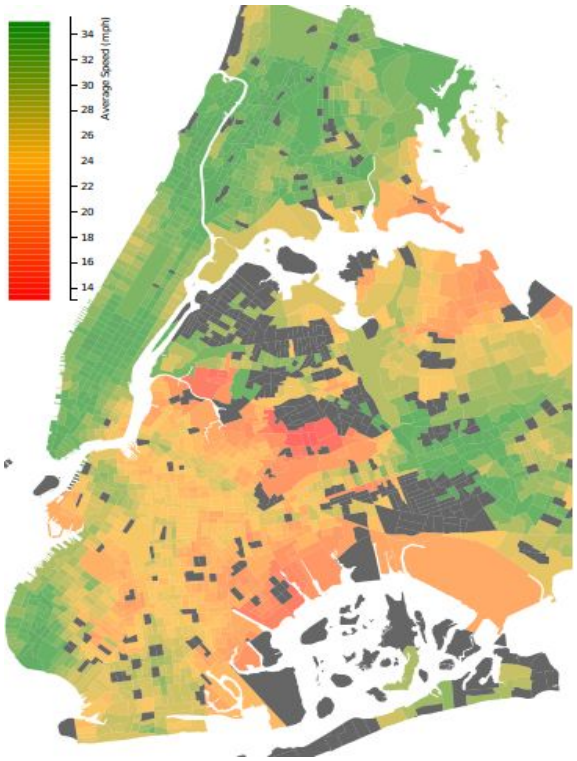

access_score
mean_income
poverty_perc
avg_travel_time
manhattan
bronx
brooklyn
Queens
staten Island

Figure 6

The following visualizations show that areas of Brooklyn and Queens have both limited accessibility and a high poverty rate. The Bronx on the other hand, has good accessibility but very high poverty rate. Perhaps this is an indicator of potential gentrification.



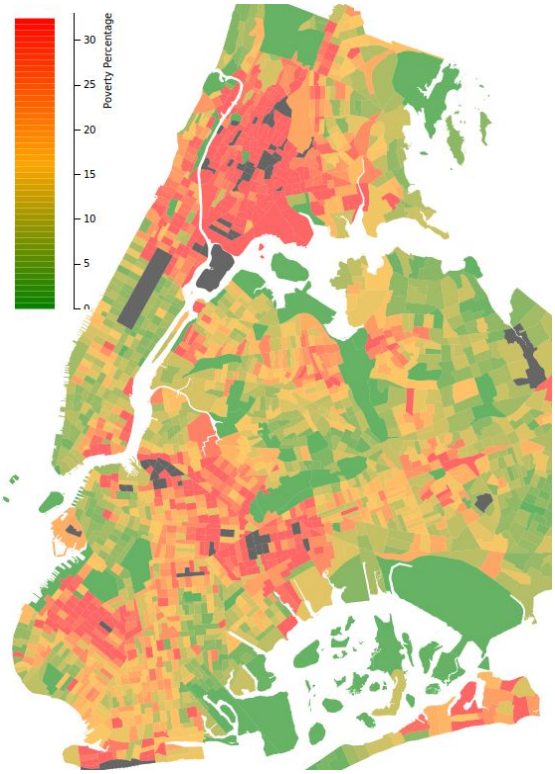


Figure 7 - Average Speed to all tracts | Figure 8 - Poverty Rate by Census Tract

Additionally we explored the data using box plots and what they reflect is the short commute times residents of Manhattan have in comparison to other boroughs (Figure 9), and the relatively short distances they have to travel to get to all other census tracts (Figure 10).

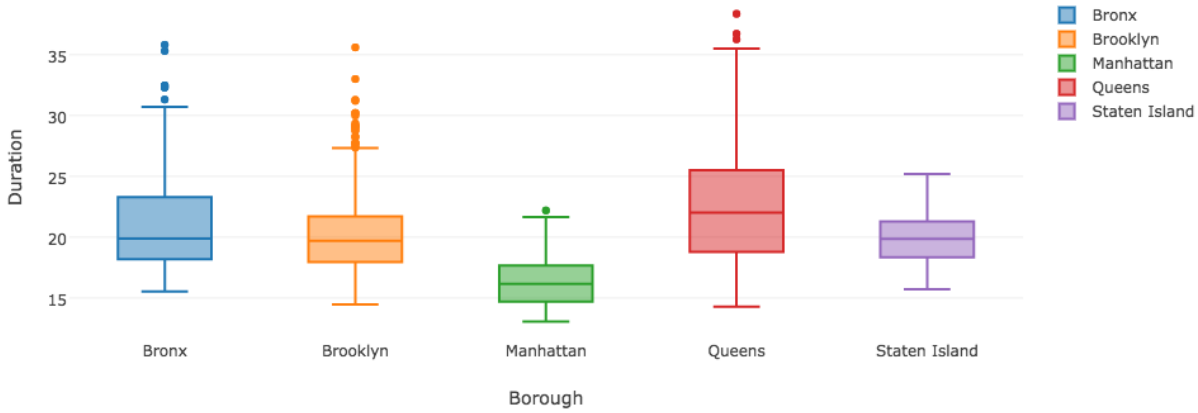


Figure 9

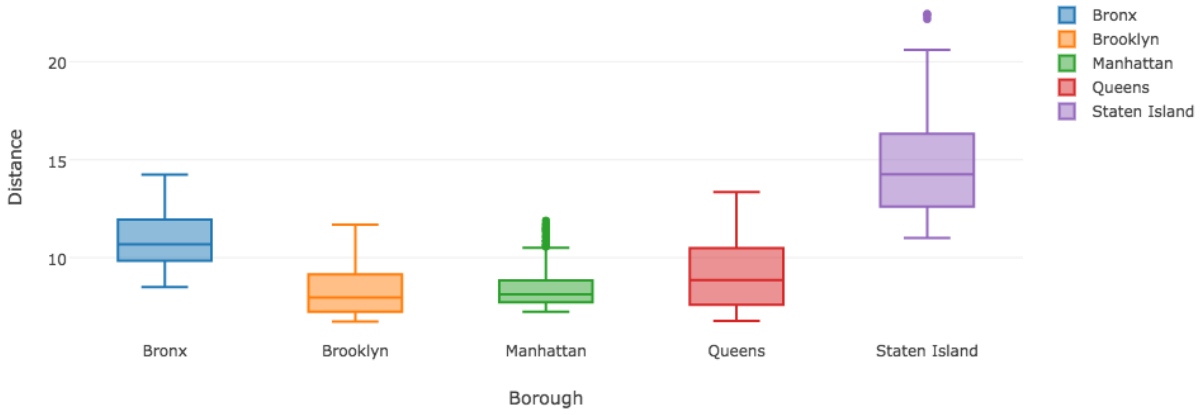


Figure 10

Individual Contributions

Michael Higgins: Worked on data visualizations in D3, merging MTA data and computing optimal routes using the NetworkX library.
Matthew Dunn: Data wrangling, mapreduce scripts, MTA data wrangling, statistical analysis, computing optimal route, geocoding scripts.
Katrina: Collecting and preprocessing Census data, MTA Train data, NYC tract shape files data; Census data manipulation and statistical analysis; assisting with the D3 visualizations, Dijkstra shortest path algorithm and distance computations; MapReduce Warrior

Conclusion

An obvious conclusion we took away from the process of handling a large scale data project is that data wrangling, data merging, and remote data processing can consume significant amounts of time. Consequently, it should be a high priority to create a road map that details a much of the final outcome as possible, as iterating through big data problems requires significant resources in terms of time and effort.

In terms of understanding urban interactions, the visualizations of the optimal travel paths gave us interesting perspectives on how one's location their commuting time and potentially their family's economic fortunes and how dramatically your optimal route changes based on a small differences where you originate. Further, there is significant uncertainty when exploring real world data, as we suspected we'd find a strong relationship between Census statistics regarding income and poverty but we found our accessibility score for a given tract was not as strong as we expected.

By far our most surprising result was finding mutual information of .9997 between our accessibility score and the Census average commute time. While having almost perfect mutual

information does not allow you to find the exact nonlinear relationship between these two figures, it's nice to know it's hanging out there somewhere in the mathematical aether.

Future Work

We would like to continue with the visualizations that we have created, bringing in street level data as well the actual MTA start/stop times. Seeing how these optimal routes change over time would be incredibly interesting. We would also like to create an alternate kind of map where the distance between geographical locations is no longer "as the crow flies" but the travel time between a given start location and every other location. D3's smooth transition functions seem to be perfect for this kind of work.

References

1. New York Times: Data Source - 2014 American Community Survey (ACS) - http://www.nytimes.com/2014/09/18/nyregion/gap-between-manhattans-rich-and-poor-is-greatest-in-us-census-finds.html?_r=0
2. Raj Chetty and Nathaniel Hendren, Harvard University - The Impacts of Neighborhoods on Intergenerational Mobility - http://www.equality-of-opportunity.org/images/nbhds_exec_summary.pdf
3. Aggarwal, Seema. Using mutual information for extracting biclusters from gene expression data, Chapter 4 - http://hdl.handle.net/10603/28365

Group Members

Katrina Evtimova <kve216@nyu.edu>
Michael Christopher Higgins <mch529@nyu.edu>
Matthew Thomas Dunn <mtd368@nyu.edu>

Project Proposal

Due April 10:

1. **Project Description:**
    a. Project Manhattan will focus its efforts on developing a model for analyzing the relationship of transportation “costs”, i.e., time, money, etc., and the demographics of the NYC boroughs.
2. **Data Sets:**
    a. 2015 Yellow and Green Taxi Data
        i. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
    b. Citi Bike Trip Histories
        i. https://www.citibikenyc.com/system-data ("Citi Bike Trip Histories" section)
    c. MTA Train
        i. http://www.mta.info/schedules
        ii. https://nycopendata.socrata.com/Transportation/Subway-Entrances/drex-xx56
    d. Census Demographic Information
        i. http://www1.nyc.gov/site/planning/data-maps/open-data/districts-download-metadata.page
        ii. https://www.census.gov/data/developers/data-sets/acs-survey-1-year-data.html
        iii. https://www.census.gov/data/developers/data-sets/Geocoding-services.html
    e. MTA Bus:
        i. http://web.mta.info/developers/developer-data-terms.html#data
        Scheduled routes for all boroughs, includes shape files

        ii. http://web.mta.info/developers/MTA-Bus-Time-historical-data.html
        Aug 1 - Oct 31, 2014 Records (needs a lot of cleaning)
3. **Milestones:**
    a. **Collect Transport Data:** All data sets listed above.
    b. **Build Trip Level Tables:**

i. Standardize the format of Yellow Taxi, Green Taxi, Citi Bike, MTA Train data so that we can easily extract:Start_Lat, Start_Long, Elapsed_Time, Trip_Cost, End_Lat, End_Long.

c. Merge with Census tract data

i. Map Trip Origin and Destination to a Census Tract

d. Build Average/Aggregate Trip Tables:

i. For given origin and destination Census tracts calculate:

1. average trip time depending on the time of day (split each day into 24 one-hour blocks)
2. Average trip price

4. Proposed timeline with weekly milestones

Date	Description of Week's Activities
Week 1 (4/10-4/16)	● Complete project proposal. ● Collect and organize data sets into csv format.
Week 2 (4/17-4/23)	● Build Trip Level Tables ● Merge with Census data ● Work on the final report
Week 3 (4/24-4/30)	● Carry out analysis and get preliminary results ● Work on the final report
Week 4 (5/1-5/7)	● Final results ● Work on the final report
Week 5 (5/8-5/14)	● Finish the final report and submit ● Prepare poster
Week 6 (5/15-5/21)	● Project presentation

Due Dates:

1. **April 10:**
   a. Submit the form with the information for your group. In the Google Doc, indicate your choice for the project, the data sets you will use, the tasks you will carry out, and a proposed timeline with weekly milestones.
2. **April 17:**
   a. Add a new section in your Google Doc entitled "Status Report" describing any issues you encountered and updates to your initial plan.
3. **May 1:**
   a. Add a new section in your Google Doc entitled "Status Report and Preliminary Results", describing your preliminary results and any updates.

4. **May 13th:**
    a. Final project report due. Add a new section in your Google Doc entitled "Project Report"
        i. You can use your Google Doc as the starting point for your report.
        ii. You should describe your experience, issues you encountered (e.g., dirty data) and how you dealt with them.
        iii. You should report on and explain the findings of your analysis -- the use of insightful visualizations is highly encouraged.
        iv. All results in your report should be reproducible -- the code/scripts you used should be made available together with instructions on how to run them to derive the results you obtained.
        v. You should describe the experimental setup (cluster configuration, number of mappers/reducers, tools you used) as well as report on the performance of your approach (e.g., report the running times of the scripts) and any optimizations you applied to speed up your code.
        vi. You should describe the individual contributions of each of the project's members.
5. **May 16th:**
    a. Project presentation: each group will present a poster with their results. The instructors will select the best 3 presentations; students in the selected groups will get extra credit: one letter upgrade (e.g., if the final grade is a B, the student will get a B+)

Research Notes!

https://www.fcc.gov/general/census-block-conversions-api
http://www1.nyc.gov/site/planning/data-maps/nyc-population/demo-tables-2010.page
https://bronx.lehman.cuny.edu/ (mta data) Username: thisisanameforsure , Pw: Manhattan1004
http://datamine.mta.info/user/reset/60586/1460223208/nF-nzC45XSG6cWQJDl1kyIcr_L9XuDl38BGzu5hwPhI (mta) username: mch529@nyu.edu, pw: Manhattan1004
APIkey: 8db2069d7152ffe85fe3d0d67297c26b
https://spatialityblog.com/2010/05/06/mta-data-in-gis-format/

Status Report #1

Relate Trips to Economic Data

One of the significant hurdles we've encountered during the past week is evaluating how to geocode the start and end latitude and longitude pickup and dropoff points. Because each census block is not perfect polygons determining which census block a ride started or ended in is not a trivial matter. As a result, we've identified a web service provided by the United States Census that takes latitude and longitude coordinates and returns a census block id along with other information if requested. We are unsure if this is a sufficient solution, as we don't think we can geocode all of the taxi trips. Possible solutions to this issue we've been considering:

1. Averaging together taxi trips based upon an appropriate radius, but we are concerned we will run into issues with group these together in a logical manner. This solution may work well as it allows the taxi trip data to indicate where drop off and pickup clustering occurs organically.
2. Adjust the above option by using a d3 clustering tool to divide up NYC based upon all of the latitude and longitude data points. We are concerned about scaling issues with this solution as it may not handle the large number of taxi trips we would load up into the system.

Bus Transport Data

An issue we confronted was how to effectively determine the pickup and dropoff locations of bus data because we only know where people enter the bus and don't know where they depart the bus, so we are considering not utilizing this data point as it seems we won't be able to accurately interpret where people enter and exit the service.

Train Station Proxy

Another issue we investigate the past week was how to address the issue of relating specific train stations to surrounding communities to provide us an appropriate estimate of their access to the train system. Currently, our thinking is to include all latitude and longitude points in a radius increasing outward from the train station location.

Data Sets

The group has reached out to several external resources for additional information about approaching the problem of mapping the different parts of NYC taking into account both time and money. We've collected the data sets we described below with some minor adjustments and additions. One such adjustment is the addition of the American Communities Survey Data which we are going to use for categorizing the income level of a specific area.

Data Preparation

We've started processing our taxi, citibike and mta train data into a single large table that will allow us to easily extract the pertinent data for our analysis. Specifically, we have been focused on getting these data sets into a standard format so we can perform statistical analysis upon them.

Status Report #2 and Preliminary Results

Progress

-Integrated census tract map with census data, made heat map according to tract/income level in D3
-Reduced number of stops for train/bus/citibike so that stops that are geographically close are considered the same station
-Overlaid route data for train/bus on map
-Found census data on commute time, joined it with other census data
-Map/Reduced trip data for bus/train/bike for one score representing ave time of travel between two stops