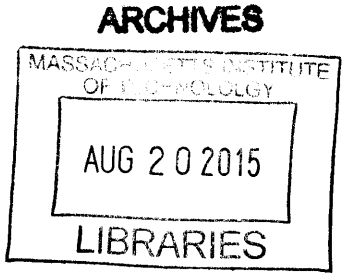
Harnessing the Power of Data Visualization to Improve Cities



by

Jennifer Jang

B.S., Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

Signature redacted

Author ....
Department of Electrical Engineering and Computer Science
February 5, 2015

Signature redacted

Certified by..... .....
Sepandar D. Kamvar
Associate Professor
Thesis Supervisor

Signature redacted

Accepted by ....
Albert Meyer
Chairman, Masters of Engineering Thesis Committee

Harnessing the Power of Data Visualization to Improve Cities

by

Jennifer Jang

Submitted to the Department of Electrical Engineering and Computer Science
on February 5, 2015, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

With the advent of the internet, we now have access to more public data than at any other point in human history. However, much of this data still exists as bits stored away somewhere on the web, both hard to find and hard to understand. For this thesis, I worked on finding ways to visualize large datasets in succinct, comprehendable ways to tell meaningful stories about the cities that we live in. I present a portfolio of six visualizations that I made as examples of how data can be represented in understandable and beautiful ways. The first three maps deal with public schools in the city. More specifically, the maps deal with exposing trends in public school systems and pinpointing different compounding factors that may influence both the average quality and variance of education in a city. The fourth map deals with tracking immigration by sea and using that information to trace the cultural lineage of a city. The last two maps deal with public transportation and connectedness in a city. These maps may shed light on where population hubs occur in a city and what aspects of a city's public transportation system may be made more efficient. I present these maps as examples of data visualizations that may be adapted to visualize other datasets or be used as inspiration to understanding data and cities on a deeper level.

Thesis Supervisor: Sepandar D. Kamvar
Title: Associate Professor

Acknowledgments

First and foremost, I would like to thank my parents, Simon Jang and Tiffany Jang, and my sister, Justine Jang, for their neverending love and support for me, especially during these last few years at MIT.

Most importantly, I would like to thank my advisor, Sep Kamvar. I would not have made it so far and learned so much without your constant support and neverending patience. A year and a half ago, when I first joined the team, I had not expected to branch out so much from my computer science background, and now I am grateful to you for opening my mind with your vast repertoire of knowledge on topics ranging from design to urban development to social justice. I would also like to thank Yonatan Cohen for inspiring me with his many great ideas and keeping me on schedule, Jia Zhang for always being available to help with anything at a moment's notice, and the rest of my colleagues in the Social Computing lab. I had a great time working with all of you and will be sure to keep in touch.

I would also like to thank my professors and mentors throughout my undergraduate and graduate career at MIT. Without them, I would not have had the solid foundation to make this journey. I would also like to thank Brandon Sim for helping me when I didn't know where to start and Morgan Lai for being an awesome, supportive roommate.

Finally, I'd like to thank Anand Oza, who has helped me immensely throughout this project, including giving his momentary support while I wrote these acknowledgements.

Contents

List of Figures

List of Tables

1

Introduction

Since the beginning of time, human beings have been singularly obsessed with recording and preserving information. Ancient societies have left behind detailed descriptions of anything from inventions to poetry to politics. Whether their motives for keeping such detailed records were to further their own legacies or for safekeeping in case of an emergency, the reality is that human societies are incredibly information-driven. Even before computers were invented, humans kept meticulous logs wherever they went, and now with the advent of the internet, that data is more publicly available than ever. We have data from thousands of sources on millions of different topics available on the web, mostly in raw, unprocessed forms. Terabytes of CSV files are tucked away behind minimal interfaces, and scans of logs and books that once required scholars to travel thousands of miles to access are now available to anyone with a click of a button. Interpreting this data and deriving meaning from it, however, is a problem that has yet to be solved by a computer and remains the limiting factor that makes most of these datasets meaningless to the average person.

The immediate goal of my thesis is to take such publicly available information represent it in a way that can be meaningful and understandable to the average person. It should be no surprise that cities are one of the biggest sources of constant data as well as important centers of life, culture, and innovation. Thus, I focus on using datasets about cities to tell stories, raise awareness, and urge action. Hopefully, any viewer will be able to draw powerful messages from the data about the bustling

cities that we live in.

In each of the next few chapters, I will talk about one of six maps I created for the You Are Here project in the Social Computing lab under Professor Sep Kamvar. Most these maps were created using HTML and CSS for the frontend and Javascript, jQuery, and the D3 visualization library for the backend. Data scraping, cleaning, and processing was mostly done using Python and several Python libraries, including the web scraping libraries BeautifulSoup, Mechanize, and Selenium. Each chapter goes through, in approximate order, motivations for each map, data collection, design concepts and implementation, challenges, and further work.

I believe that the power of data will drastically change the way we make decisions for the good. The ultimate goal of my project is to harnesh this power to optimize city planning and improve the efficiency and utility of many high-impact public initiatives, and show how data can be used to generate immeasurable social value for a city and those whose lives are a part of it.

2

Immigration by Sea

2.1 Motivation

This series of maps (figure A-1) focuses on the history of immigration by sea. Immigration lies at the heart of American culture. Historians estimate that throughout the 17th and 18th century, fewer than one million immigrants made their way to the United States. By the end of 19th century, however, almost 15% of the population in the United States was born on foreign soil. Due to technological innovations and economic difficulties overseas, travel across the Atlantic Ocean became cheaper and more enticing. During this time period, small centers of rich immigrant culture like Little Italy in New York City and Chinatown in San Francisco began to pop up all over the country.

I developed this map for the cities of New York, Baltimore, Boston, San Francisco, and New Orleans. I wanted to create an animation that showed the diversity of immigrants flowing into the different cities of the United States during the 19th and early 20th century. There are many interesting things one can learn just by looking at the immigration records of a city. In addition to learning more about our own ancestors, we can also see when and how groups of immigrants started to shape the cultural landscape of the cities they adopted. For example, the city of Baltimore is very heavily influenced by German culture, and looking at the immigration map of Baltimore, we can see that from the years 1880-1890, almost all immigrants into

Baltimore are of German origin. We can also see how historical events such as the onset of World War I heavily disrupted the movement of people around the world. For many cities, immigration sharply declined betwen 1914 and 1918 and picked right back up after the end of the war.

The immigrants from this era, many of whom have been long assimilated into mainstream American culture, continue to contribute to the melting pot of cultures that the United States is known for today.

2.2 Data

The data for these maps came from a variety of sources. My data for New York City and San Francisco came from genealogist Stephen P. Morse, whose website held databases of passenger and ship manifests for many cities. For cities that Mr. Morse did not have data for, I scraped Ancestry.com, a website dedicated to help users find out more about their family tree. Ancestry.com was a great resource for this map since it contained millions of databases detailing the emigration and immigration of people. I also used a vital records database published by the state of Massachusetts in order to create the Boston immigration map. For most of these cities, I had sparse data before the mid-19th century and after the mid-20th century when planes began to overtake ships as the main mode of intercontinental transportation. Thus, I restricted my map to the years 1850-1925, except for the case of San Francisco, where I only had data from 1893-1933.

Since my data came from different sources, there were a few small differences between datasets. The most notable difference was that the New York City and San Francisco maps featured the number of ships that entered into a port of immigration and the rest of the cities featured the number of passengers. For New York and San Francisco, each ship was estimated to carry around 100-400 passengers. Without loss of generality, I will talk about the New York City map for the rest of this chapter, but the process I took to create the other maps was more or less the same.

The data, as it occurred online and immediately after scraping, was missing in-

formation and in need of cleaning. To create this map, I only needed to scrape the port of departure and the year of travel for each ship. However, not every ship had a departure port listed; many rows only had the departure country. In addition, for the maps that dealt with ships rather than passengers, passengers on a ship may have made multiple transfers before arriving in New York City and may not have originated from the port of departure listed for that ship. There wasn't much we could do about the latter issue, but for the former, I cleaned the data by replacing all departure ports with a departure country. Although it would have been more meaningful to use the departure city or port for the map, too many entries failed to list anything more than a country, and getting rid of those entries would have skewed our data.

To clean the ports of departure data, I started by writing a Python script that helped me query Wikipedia for any unknown ports that were not the name of an official country. There were over 4,000 distinct port names despite there being only around 200 official countries in the world. Many ports were famous international cities such as Lisbon or Venice. For these countries, I was able to scrape Wikipedia for their corresponding countries. For ports that did not have a Wikipedia page or were spelled wrong, I had to manually query Google for more information. To speed up the process, I used Python's approximate matching algorithm to check if a port has a similar spelling to or contains the name of a port that has already been labeled. This helped cut down on extreme cases such as the more than 200 different variants of the name "Buenos Aires" that appeared in the ship logs In the end, the number of ports that needed to be hand labeled was a little bit over 600.

For the databases from Ancestry.com that contained information about passengers rather than ships, the data was easier to process. Ancestry.com had a built-in algorithm to determine if the origin listed for a passenger fit your search criteria. For example, if you searched for all passengers from Germany in the year 1850, Ancestry.com would also return passengers that had origins listed as Berlin, Hamburg, or Bavaria. Thus, instead of scraping all results from Ancestry.com and then separating them by origin, I queried for passenger lists for every year and every country and just counted the number of entries. This method required me to make about 200

different queries per year even though most countries never sent any passengers to the United States in all 76 years (1850 to 1925 inclusive). Thus, to speed up the process, I searched for the number of passengers per country for the time period if 1850-1925. If no passengers were returned for this period, I took that country off of the search list. This proved to be much faster than scraping the entire database since there were upwards of hundreds of pages and millions of entries per year. Coupled with the fact that I didn't need to process origin ports after scraping, this method of searching by country was a great improvement.

2.3 Design and Implementation

2.3.1 Concept

I wanted to create a map that gave off a sense of movement to represent the immigrants who left their homes in order to settle in a foreign country. I also wanted to convey how diverse these immigrants were and how ships were arriving from almost every corner of the planet. To do this, I created an animated map of the world using Javascript and a D3 library called Datamaps. For every country that has sent at least one ship to the port of New York City in a given year, the map draws an arc between the midpoint of that country and New York City. The thickness of the arc and the shade of the origin country are drawn proportionally to the number of immigrants who have made that trip during that year. The map runs through the years 1850-1925 and displays the total number of immigrants per country at the end of the animation.

2.3.2 Implementation

The backbone of the animation is a Javascript timeout function that steps through the years, one year per second, from 1850 to 1925. Every year, all countries are redrawn using data from the new year. However, a few problems arose from the simple implementation.

At first, we used the attribute of opacity to substitute for shade. Since the map was overlayed on a white background, a color with a lower opacity would appear to be lighter shade than that same color at a higher opacity. This method was very simple to use since in this case, opacity would just be set to the ratio of that country's current level of immigration to the highest level of immigration in all 76 years. However, using this simple method did not give the range of colors that I wanted. Thus, I wrote a Color class that would mix two hex colors. For example, in Figure X, a value of 0 would give a color at the leftmost spectrum, and a value of 1 would give a color at the rightmost spectrum. This way, colors appeared vibrant at either end of the spectrum, instead of fading to white at lower values and appearing almost black at higher values.

Second, the changes in colors between the years seemed abrupt and unnatural. Immigration is a smooth and continuous process, so I wanted to simulate the continuity. I accomplished this by splitting each time step of 1 year into 10 separate intervals. During each interval, I slowly extrapolated the per-year immigration rate and colored the country accordingly. For example, if my data says that France has x ships arriving in New York City in 1850 and y ships arriving in 1851, I would color France for $\frac{y-x}{10}$ more ships at every time step until I reached 1851, when I am colored for exactly y ships. This made the map appear more seamless and natural, as immigration over time should.

Finally, I plotted a line graph of the total number of arriving ships year to year as well as with number of ships per year of any selected country. I also plotted a static bar graph of the total number of arriving ships per country, sorted in ascending order. One of the problems with plotting every single country in the bar graph is that most countries do not send ships to New York City and a few others have sent just a handful for the entire time period of the animation. Bars for these less active countries did not even have hoverable height when compared to the most active countries. Thus, for the bar graph, I only showed countries that have 1% or more of the number of ships from the country with the highest immigration. In the case of New York City, this country would be the United Kingdom.

In order to help viewers understand that the line graph is a graph that moves with the animation but the bar graph is static, I implemented a moving slider line that moves along the x-axis of the line graph with the animation.

2.4 Challenges and Future Work

This was a challenging map for me since it was one of the very first that I made. Most of my time was spent scraping and cleaning imperfect data. In addition to the aforementioned problem of having inconsistently labeled origin ports, I also ran into the issue of inconsistent passenger reporting. For example, even though each entry was supposed to be for one passenger, some would list entire families or groups. I saw many entries that were labeled as "3 servants" or "4 children", so merely counting the number of entries was not entirely accurate. However, due to the way I scraped my data, I was not able to parse and split up these combined entries. Since these entries were only a small fraction of the total database, I ignored them, but further work could be done to fully parse the immigration logs and count the total number of passengers more accurately.

I made this map for five cities, but the map could be adapted for even more. Ancestry.com currently has data, albeit much sparser data, for cities like Honolulu and Philadelphia, and other geneological archives might as well. Immigration logs exist for almost every city that faces the water, but for some of these cities, logs may not exist in a digitized form. Ancestry.com has scanned hundreds of thousands of these handwritten logs, but many logs have not been transcribed and/or are illegible. In addition to the technical difficulties of cleaning transcribed data, if we wanted to create expand this map to include even more cities, researchers would also have to take the time to transcribe the data that is available.

Finally, for the maps that only record ship movement, we lose information about where the passengers are actually coming from. For example, only 2 ships have ever sailed from Russia to New York City, yet there is a sizable presence of Russian immigrants in New York. This is because most Russian immigrants travel to Europe

first before making the final leg of their journey to the United States. However, my map wrongly implies that very few Russians immigrate to New York City. Further work could be done in actually tracing the lineage of individual passengers and figuring out where exactly they come from.

3

Schools and Poverty

3.1 Motivation

New York City is home to one of the largest student populations in the world, with a total of 1.1 million students attending over 1,700 schools. These schools are separated into 32 school districts, colloquially known as "zones." In theory, students in New York City can attend any other school in the city regardless of the zone that they live in. Famous public schools such as Stuyvesant High School attract many top students from all five boroughs. However, in reality, most students attend nearby high schools in the zone that they live in. Even amongst neighboring districts, huge differences abound. For example, Bedford-Stuyvesant of Brooklyn has a student body where 89% of its students are on a free or reduced lunch plan. Staten Island, on the other hand, has only half of that. It is perhaps of little surprise, then, that the graduation rate of high school students in Bedford-Stuy is only 53% versus Staten Island's 78%. This is just one of many examples where large inequalities exist, even between districts within the same city.

This map (figure A-3) aims to visualize educational inequalities between school districts in New York City. To me, one of the most shocking things about this data was the incredibly high percentage of public school students who are on a free or reduced lunch plan. A student qualifies for free or reduced lunch if their family income is 180% of the poverty line. Perhaps less shockingly, differences in average levels of

economic inequality correlate highly with almost all academic performance and school quality indicators. In order to show this, I chose metrics such as teacher experience, graduation rate, and average SAT score, and plotted the correlations between each of these metrics with the overall level of free or reduced lunch students in the district. Since we wanted to use metrics intrinsic to the schools, we used the percentage of students that received free or reduced lunch as a substitute for the average rate of poverty in that region. For the rest of this chapter, the terms "poverty rate" and "free or reduced lunch rate" will be interchangeable.

From this map, the viewer can clearly see that both the quality of a student's education and the level of a student's performance are inversely correlated with whether or not he or she is living in a region with high poverty rates. Interestingly, average costs per student in a district correlate positively with poverty levels. This may seem surprising at first because that implies cost per student is anticorrelated with metrics like average teacher salaries and student performance. Clearly, even though the city is spending far more per student in poorer districts, the mere act of spending more money alone is not increasing the students' graduation rate, attendance rate, or standardized test scores.

The poorest areas of New York City lie in northern Brooklyn, The Bronx, and upper Manhattan. Not surprisingly, these areas are also the lowest-performing districts in the city. This map aims to pinpoint the most underperforming schools in New York City and their areas of need, whether those needs are for more funds, better teachers, higher teacher retention, or more student support. Hopefully, this map will also convey that there are deeply ingrained variables at play here, and that just blindly throwing money at the problem will not make it go away.

3.2 Data

All of the data I used is available for free online. Most of it came from the New York State and New York City Education Department website. For each of the 32 school districts, the NYSED site compiles yearly information on school demographics and

enrollment. In Table 3.1, I list the categories of data that were included as well as the ones I ended up using.

Category	Metric	Included
Demographics	Gender	No
	Race	No
	Limited English proficiency students	No
	Students with disabilities	No
Economic Indicators	Free/reduced lunch	Yes
	On government assistance	No
Teachers	Certified	Yes
	Less than 3 years experience	Yes
	Turnover Rate	Yes
Costs	Cost per student	Yes
Students	Attendance rate	Yes
	Suspension rate	No
	Graduation rate	Yes
	Post-graduation plans	No
	New York state assessments	Yes
	SAT scores	Yes

Table 3.1: The categories of data available on the New York State Education Department website.

In addition, the school accountability report published by the New York City Department of Education details the total cost spent on education in each district as well as the number of students enrolled. Dividing the former by the latter gives a good rough estimate of the cost per student in that district.

Two of the metrics, average teacher salary and SAT score, were not published on the state- and city-wide databases. In order to get average teacher salaries, I crawled the privately-owned site `http://seethroughny.net/`, which gave median teacher salaries as well as salaries at the 5th, 25th, 75th, and 95th percentiles. For SAT scores, I took SAT scores from a database published by NYC Open Socrata, a website that publishes free data about all different aspects of living in NYC. Since this database listed averages by schools, I had to calculate the district value by averaging SAT scores over all schools weighted by the number of attendees at each of those schools.

At first glance, racial demographics seemed to be a good piece of data to include

in the map. The racial makeup of a district is highly correlated with the quality and performance of the schools in that district. Areas with a higher percentage of minority (Black and Latino) students have less qualified teachers and fewer resources as well as higher dropout rates and lower test scores. Not surprisingly, districts with large minority communities are also correlated with higher poverty rates. Due to the sensitive nature of comparing race with academic performance, I decided not to include racial demographics in my map. I did not want it to detract from the broader message of poverty in school.

3.3 Design and Implementation

I wanted a visualization that would allow the user to quickly understand the data and feel compelled to engage with the map, since most of what the map offers requires user interaction. I created a table that allowed the user to choose from the group of ten metrics and colored the map based on whatever metric was chosen. In addition to coloring the map, I also drew a scatter plot with the chosen metric on the x axis and the poverty rate on the y axis. Finally, I drew a best fit line for the scatterplot.

In order to set the color of a district based on the selected metrics, I first set all districts to be the same color and then changed each district's opacity. These district polygons exist on top of a white background, so changing the opacity has the same effect as changing the shade of a color. I calculated opacities by the following formula:

$$\text{opacity} = \frac{\text{current value}}{\text{max value}}, \tag{3.1}$$

where max value is the maximum value for that metric over all 32 districts. Districts with higher values for a metric will be darker, and districts with lower values will be lighter.

The best fit line is drawn using the least squares method. In order to get the linear best fit line with equation

$$f(a,b) = a + bx, \tag{3.2}$$

we minimize R^2, the sum of the squares of the vertical deviations from the mean:

$$R^2(a,b) = \sum_{i=1}^{n} [y_i - (a + bx_i)]^2 \tag{3.3}$$

$$\frac{\delta R^2}{\delta a} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)] = 0 \tag{3.4}$$

$$\frac{\delta R^2}{\delta b} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)] x_i = 0. \tag{3.5}$$

Solving these equations will give us variables a and b which define the best fit line for the graph.

At first, I wanted the user to be able to compare and contrast any two metrics. I designed an interface that featured two identical maps of New York City and a table that showed all of the selectable metrics. The viewer could select any two metrics to fill the maps and a scatterplot would appear that related the two chosen metrics. However, it made little sense to be able to correlate any two metrics, and this extra functionality detracted from the message of the map. Several pairs of metrics were either very obviously correlated, such as NYS scores and SAT scores, or misleading, such as test scores and teacher salaries. I decided not to confuse the user by adding unneeded functionalities, so I fixed the y axis to poverty rate instead.

I used geographic information system (GIS) boundaries from NYC Open Socrata and used Javascript and D3 to render the polygons. I also adapted a scatter point graph D3 library to render the graph.

3.4 Challenges and Further Work

The biggest challenge to creating this map was figuring out what exactly I wanted to convey. Consequently, it was hard to decide what design direction to take with the map before I knew what to do with the data. I didn't know what part of the sheer volume of data published by the department of education was interesting, what part I should visualize, and what part I should leave out, if any. However, actually analyzing the data revealed some very interesting patterns that I thought were thought-provoking, surprising, and eye-catching. Those patterns helped me select the right data to show and also laid the groundwork for designing the final map.

So far, I have only produced the Schools and Poverty map for New York City. This type of map would be difficult to produce for cities like Los Angeles, San Francisco, and Boston, since all of these cities have one unified school system and are not segregated into zones the way New York City is. In addition, most of the data I collected for this map was published directly by the New York City Department of Education. The same data may not always exist for other cities. Similar data may exist, however, so a related but slightly different map might be feasible. Lastly, New York City is unique in the sheer number of schools it has, which is no doubt one reason why zoning was necessary.

Certain cities like Portland and Austin do have districtization, but the populations in these citis are small enough that each district might only have one public high school, so a map that shows district averages would basically be comparing individual schools and would not be as meaningful. Further work could involve adapting this map to different cities. For example, if we wanted to create a map for Los Angeles, the second largest city in the United States, we could adapt the idea of districts to represent individual high schools instead, and create our own district borders by drawing a Voronoi diagram with the schools as points.

Both New York State and New York City have far more data than for just the current year. In fact, this same exact data exists all the way back to the 1990s. Future work could involve making an animated map showing changes in poverty

levels, graduation rates, and other metrics throughout the years. One challenge of this is that many of the old data exist in PDFs which may be hard to scrape.

Wanting our students to succeed is an often-talked about issue that remains one of the most pressing issues today, but how to help them do so is rarely agreed upon. This map serves as a reminder that inequalities in the classroom cannot be fully addressed without addressing the engrained problem of poverty and its related evils.

4

High School Dropouts

4.1 Introduction

While making the Schools and Poverty map, one statistic stuck out to me: high school dropout rates were astronomically high in nearly every school district in the city. While New York City boasts several of the nation's top high schools, it is also home to some of the country's lowest-performing ones. In fact, some districts have upwards of a quarter of their students dropping out every year. Even for the district with the lowest dropout rate, Northwest Brooklyn, over 8% of students do not complete high school. Comparatively, the national dropout rate, defined as the percentage of 16 to 24 year olds who are not enrolled in high school and have not earned a high school degree, has hovered around 7% since 2012 and has dropped from 16% in 1990.

Students who drop out of school are not destined to live a life of poverty or crime. However, statistically, the future for dropouts is bleak. Students who drop out of high school earn less than two-thirds the salary of a typical graduate over their lifetime. They can also expect to earn a million dollars less than someone with a Bachelor's degree. Dropouts make up more than half of the unemployment rate and are 3.5 times more likely to be incarcerated. Dropouts also take a heavy toll on society in the form of lost wages, welfare payments, criminal justice, and medical care. If we can raise graduation rates throughout the country, we start a positive cycle by saving

$1.8 billion every year on lost wages alone and using that money to further improve our schools instead.

It is clear that the dropout issue is multifaceted and complicated, but one thing is certain. The areas with the highest dropout rates are also likely to be the poorest areas in the city, and underperforming schools are highly correlated with poverty. Students who live around or below the poverty line are more likely to attend "dropout factories," or high schools that graduate less than 60% of its students and accounts for more than 50% of all dropouts in the country. In recent years, efforts have been made to increase graduation rates in New York City public high schools. Graduation rates are now at an all-time high, but there are still many tens of thousands of students every year being left behind.

The inspiration of this map (figure A-5) came from a series of ads by the website BoostUp.org, which provides dropout statistics by state and demographic and urges the viewer to make a difference. One of their ads found in Cambridge stated: "7,000 high students drop out every school day. Their empty desks form a line four miles long." I was inspired by the powerful way this ad humanizes high school dropouts and gives off the message that students are not statistics, but are human beings with bright futures.

4.2 Data

The data I used came from a database published by the New York State Education Department. NYSED publishes educational accountability data every year, including the percentage of students who have dropped out of school as well as the total number of dropouts in that district. To calculate dropout rates, NYSED follows a cohort of students from freshman to senior year. My data was collected from the 2012 cohort, so the dropout rates on my map refer to the percentage of students scheduled to graduate in 2012 who have dropped out of school. Dropout rates also approximate the percentage of high school students from all four years who have dropped out that year, if we assume that dropout rates are roughly staying constant.

4.3 Design and Implementation

4.3.1 Concept

I wanted to create a map that showed dropout rates in the city by school zone, but I wanted to show each dropout as something more than just a statistic. To do this, I decided to make an animated map that spans an average 180-day, 43-week school year. My goal was to show each dropout as a circle on a map on the day that he or she drops out of school. Over the year, dropouts will accumulate on the map and cover the entire city. Even though I wanted to place a circle on the map to symbolize a dropout, I did not have any actual data about where these students lived, nor did I have any idea when students usually dropped out of school through the school year. Because of my limited data, I made the obviously oversimplified assumptions that students lived uniformly at random in a district and dropped out of school uniformly at random as well. Since my goal was not to document actual dropouts through one specific school year, I felt that these assumptions were fine to make as long as they were clearly stated on the map.

In addition to the map, I also wanted to show the sheer number of dropouts per year. Since the circles on the map frequently overlap each other, it was hard to see exactly how many circles have been rendered. Thus, I created a bucket that caught the falling circles and accumulated them as the animation ran. As the circles accumulated, I compared the total number of circles to various metrics, such as the number of United States military casualties since 2001 and the entire population of the capital of Vermont, in order to shed some perspective on the number of students that our education system is leaving behind every year.

4.3.2 Implementation

To create the animated map, I needed to find a way to randomly generate a point on a polygon. The most accurate way is to do this is to triangulate the polygon, but this was too computationally expensive to do online and too memory-extensive to do

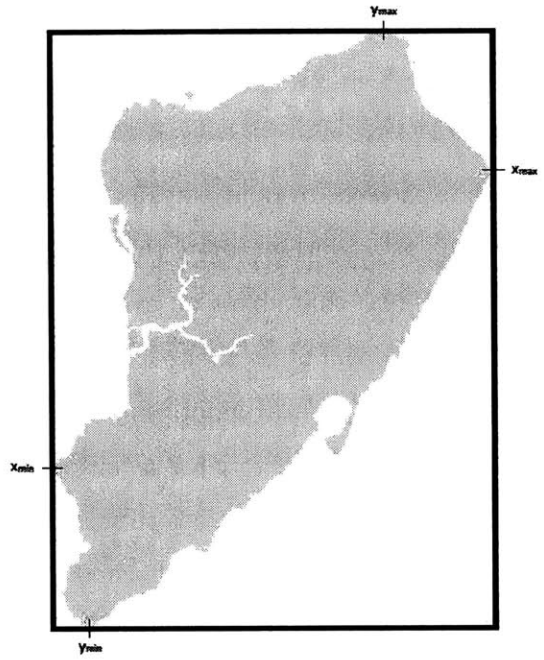


Figure 4-1: A bounding box example.
The bottom left point (x_{min}, y_{min}) and the top right point (x_{max}, y_{max}) fully characterize the bounding box of this polygon.

offline. Thus, I tried to generate points using bounding boxes.

First, I calculated the bounding boxes of every polygon by finding the minimum and maximum longitude and latitude points. This way, as in the example in 4-1, the points (x_{min}, y_{min}) and (x_{max}, y_{max}) would make up the bottom left and the upper right points of the bounding box, fully describing the bounding box that encapsulates the polygon.

Afterwards, I generate a random point in the polygon by simply querying for a random number between x_{min} and x_{max} for the point's x coordinate and a random number between y_{min} and y_{max} for the point's y coordinate. For the last step, I needed to test whether or not the point was in the polygon. If not, I would discard the point and generate a new point until I generated one that was admissible. The pseudo-code below explains how I determined whether a point was inside a polygon. The inputs to the algorithm are `x` and `y`, the x and y coordinates of a point, and `polygon`, an array of points that determine the bounds of a contiguous polygon. There were a

few districts that were actually composed of multiple disjoint polygons, so for those cases, I looped through each polygon and returned *true* if the point were in any of the polygons.

Algorithm 1 Point in Polygon

```
1:  procedure PointInPolygon(x, y, polygon)
2:      n ← length of polygon
3:      inside ← false
4:      p1 ← first point of polygon
5:
6:      for i in {0, 1, ...n} do :
7:          p2 ← ith point of polygon
8:
9:          if y > min(p1[y], p2[y]) then:
10:             if y ≤ max(p1[y], p2[y]) then:
11:                 if x ≤ max(p1[x], p2[x]) then:
12:                     if p1[y] ≠ p2[y] then:
13:                         xints ← (y − p1[y]) * (p2[x] − p1[x])/(p2[y] − p1[y]) + p1[x]
14:                     if p1[x] = p2[x] or x ≤ xints then:
15:                         inside ← ¬inside
16:     return inside
```

Using this method, I can now draw as many circles as I needed to within a polygon. To determine the number of circles I need, I divided the total number of dropouts for a district by the number of days in the school year (180). If that number is not an integer, I add the circles probabilistically. For example, if 2.6 students drop out per year, I would add two circles to the map with probability 1 and one circle with probability 0.6. This way, there are 2.6 circles drawn every day in expectation.

District	Compactness
District 18: East New York	0.220
District 27: Southwest Queens and the Rockaways	0.233
District 2: Upper East Side	0.240
District 28: Jamaica	0.302
District 23: Bedford-Stuy	0.616
District 16: Central Brooklyn	0.633
District 9: Central Bronx	0.665

Table 4.1: Compactness of selected districts

One caveat to using this method instead of the triangulation method is that there

is always the possibility of generating too many inadmissible points. To see how likely that was, I calculated the compactness of each polygon, defined as the ratio of the area of the polygon to the area of its bounding box. I listed the compactness of a few of the most compact and least compact districts in Table 4.1.

Even for the least compact district, District 18, its compactness greater than $\frac{1}{5}$. This means that on expectation, fewer than 5 points need to be generated for every point that is admissible. The most compact district, District 9, has a compactness of 0.665, or close to $\frac{2}{3}$, so in expectation only $\frac{3}{2}$ need to be generated per point drawn in District 9. In expectation, we had to draw

$$\sum_{\text{all districts}} \text{compactness}^{-1} * \text{total dropouts} = 29,061 \text{ total points} \tag{4.1}$$

for a total of $11,731$ points. This gives us an efficienty of 2.47 points per dropout.

These compactness values were calculated using QGIS, a geographic information system (GIS) application that provides viewing, editing, and analysis capabilities on GIS objects. To get each value, I calculated the areas of each polygon and bounding box and divided the areas of the polygons into that of their corresponding bounding boxes.

For the second part of the map, I made a copy of the circles that I rendered on the map and drew them inside a rectangle with height h and width w. Since the indexing of a Javascript SVG started in the upper left hand corner, the y value of a point was equal to the height h of the rectangle minus the distance the circle was away from the top of the rectangle. The column number c was initiated at 0, incremented after every circle drawn, and reset at the beginning of a new row. The row numer d was incremented after every $\lfloor w/(2 * r) \rfloor$ number of circles. The radius r was a hardcoded value representing the radius of each circle. Each circle's final placement was calculated as follows:

$$(x, y) = (2rc + r, h - 2rd) \tag{4.2}$$

In order to make the animation look nicer, I wanted to create an animation where the circles would rain down from the top of the bucket before settling into their final positions. To do this, I generated a random starting coordinate at the top of the bucket and used SVG's `transition()` function to create the movement of the circles. The random starting coordinates were generated by picking a random x axis across the width of the bucket and a random y coordinate from a 20-pixel range at the top.

4.3.3 Bottlenecks

Many problems come with such an animation-intensive map. Since almost all calculations are done while the animation is running, I didn't have as much of a problem with loading the map as I did with the rendering. However, rendering the points and placing them were slow and laggy. At first, I wondered if it was because I had to generate the points on the fly, so I thought about the pros and cons of generating all the points offline and creating a deterministic map out of one set of randomly generated points. The drawbacks of doing this were that saving the locations of tens of thousands of points would require a lot of memory, and the points would need to be regenerated offline whenever I made any changes to the dropout rates and data. Luckily, after profiling the script, I realized that the rendering was actually not the bottleneck that was slowing my code down.

After profiling, I realized that jQuery's `fade()` function was one of the biggest bottlenecks that slowed down the animation. I used jQuery's `fade()` method to fade the points on the map after I drew them. The `fade()` function took around 2 seconds during the beginning of the animation to generate an iteration of circles. As the circles accumulated the rate of each iteration sometimes took upwards of 10 seconds! After doing some research, it seemed like one of the main problems was using the jQuery selector functions. Thus, I switched to using SVG's `transition()` function once again and now the animation takes on average less than 0.1 seconds per iteration. This is faster than the speed at which I wanted to run the animation, so now the map is no longer bottlenecked by the rendering of the circles.

Hovering was also another source of inefficiency. Whenever the viewer hovered over a district, I needed the map to highlight the hovered district and show its statistics. To accomplish this, I attached mouseover handlers to both the polygon and the circles that were rendered over it. Due to the large number of circles drawn, the script had to rerun the mouseover function every time the viewer's cursor moved, even if the cursor did not leave the district. Towards the end of the animation, when almost every district was completely covered with circles, it took up to a full second to highlight the district.

To fix this, I drew two copies of every district, one on top of the other. The topmost layer covered all of the dots that were being rendered and had no color or opacity. I only attached mouseover handlers to the topmost, invisible layer, so when viewers hover over a district, the script can render the hover effect as if the SVG were just one simple layer. This cut the time it took to highlight a district down to about half a second, which is still slow but bearable.

Finally, while the animation of the map and the animation of the bucket were fast enough that their collective runtimes were within the allotted time for each iteration, there was still enough of a difference that it was clear the circles rendered first on the map. This was a little bit disconcerting to the viewer, since the circles were supposed to represent the same dropouts in both animations. To fix this small problem, I separated the two animations with enough space on the page that it would be hard for the viewer to see both animations very clearly in his or her line of sight. The elements were still close enough that this change did not detract from the map.

4.4 Challenges and Future Work

This map was only made for New York City. New York City was great to work with because of the large number of students, ample datasets related to school performance, and convenient number of districts. Other cities like Austin and Portland also have school districts and district-level performance data and may be good candidates for a similar map. However, neither city has the population density nor total number of

dropouts as New York.

While the data that was used for this map was simple and straightforward, the main challenge here was actually conveying the statistics in a relatable way. I could have easily created a map that colored each district a different shade according to the dropout rate in that district – in essence, a one-metric version of my Schools and Poverty map – however, I felt that actually taking the time to render a circle for each dropout tells the story in a more powerful and humanizing way.

In addition, since the data was so simple, I had to make many assumptions about the distribution of the data for my animation. Most notably, I made the assumption that students lived uniformly in a district and dropped out of school uniformly across the school year. In reality, neither of these assumptions is true: even within a district, there are areas of varying population density. For example, even though no one lives in Central Park, my script still accepts points generated in that area. Also, it is unlikely that students drop out of school uniformly throughout the year. Most students probably make the decision not to go back to school before and after major holidays and breaks. I had to make sure that these assumptions were easily understood to the viewer, but still create a map that conveyed the message I wanted to convey.

A possible future improvement for this map would be to collect data about the actual distributions of these variables and sample from these more accurate distributions. Population maps could be used to generate the actual locations of these dropouts, and research could be done into the most common times when students decide to leave school.

Finally, one criticism of this map is that it ultimately looks like a population map. Since I fill districts based on the number of dropouts rather than dropout rate, highly populated areas in Manhattan and Brooklyn will obviously appear darker because of their high human density. One alternative was to do a normalized version of the map where we draw a dot to represent one dropout for every 1,000 students, but this made larger districts lighter and the map became more of a map of land area, which was even less meaningful. Later, I decided that since my message was to

showcase the sheer number of dropouts in New York City, and I included a histogram of comparative rates, it didn't matter as much. A future improvement could be to find a way to represent the dropouts in a way that allows the viewer to get a sense of both the dropout rate as well as the total number of dropouts in that area.

5

Costs of Schools

5.1 Motivation

At the crack of dawn, more than 1.1 million students wake up and head to one of almost 2,000 schools in New York City. Schools are where we begin to cultivate and encourage the next generation of brilliant thinkers, and it is also once again the focus of this next map. New York City is known for its many prestigious schools, from famous public magnet schools to expensive day schools that cost more per year than the average household income in the United States. The city is also home to inner city public schools labeled only by a string of numbers, parochial schools managed and funded by a nearby church, and public special education schools that serve the city's special needs children. For this map (figure A-7), I aim to visualize the range of tuition and per-child costs throughout all New York City high schools. Even among public schools in the same region, average costs per child may vary wildly. This map will help give an understanding of where it costs the most to send a child to school and what areas have the highest demand for expensive, private schooling.

5.2 Data

Wikipedia has a list of all high schools in New York City, which was already conveniently labeled as either public or private. Unlike the previous maps, this map did

not have a centralized database that I could crawl to get the rest of the information I needed. Thus, most of the work that needed to be done for this map consisted of finding data by hand and writing Python scripts to help speed up the manual part of the process.

The first part of the data collection involved finding the longitude and latitude coordinate points of every school that I wanted to feature. I needed these points so I could mark their location accurately on the map. I used Python and BeautifulSoup to search for the address of every school on Google. For most schools, Google would return a simple, instant answer before any of their search results, so I didn't need to crawl any further webpages. If a certain Google search query did not return data in the proper form, I would have to search for that school's coordinates manually. However, the script was able to generate addresses for almost all of the schools on my list. After I got a list of schools, I used `stevemorse.org` to batch convert all addresses into coordinates. The next step was to collect costs and tuitions for each school.

The database I used while making the Schools and Poverty map that held total expenditures and total enrollment per district also contained the same information for all public schools and special education schools in the city. However, data collection was more difficult for the private and parochial schools. There were no centralized databases that listed the up-to-date tuition rates for all private schools, so the only way to find these numbers was to navigate the school's website.

In order to help with collecting tuition data, I wrote a script in Python that first searched for the school in Wikipedia. A few of the well-known schools had tuition costs listed on Wikipedia, so I did not have to search any further. If that didn't work, the script would search for the school on Google and let me manually navigate to the tuitions page to find yearly tuition costs. I did not include any book or uniform costs in the tuition, but did include other required fees such as registration and scheduling fees.

There were a few schools where I could not find any tuition data listed on the school's website. There were some schools, mostly orthodox Jewish schools, which

did not have a website at all. For some of these schools, I was able to find approximate tuition costs from third party websites, but for others, I decided not to include them in the map.

Finally, I had to label every school based on their type: public, private, parochial, or special education. The New York Department of Education website had a list of all District 75 schools, otherwise known as special education schools, so the public schools were easier to label. I had to hand-label the private schools as 'parochial' versus 'private.' Most of the time, when I searched for a school on Google, an informational bar on the right presents a summary of the school, usually a blurb taken from Wikipedia. In this summary, if the school is religiously-affiliated, the summary will usually contain words like "Jewish," "Roman Catholic," or "Christian." I sped up the labeling process by automatically labeling all of the schools with known religious keywords as parochial schools and manually checked the rest.

5.3 Design and Implementation

For this map to be successful, the viewer needed to be able to see the cost, type, and location of every school all at once. The viewer should also be able to compare and contrast schools as much as possible without needing to click on too many things. Thus, I had to find at least three properties that were orthogonal to each other and could represent all three descriptors at once. Location was the most natural: circles were used to represent schools and were placed on the map at the coordinate of the corresponding school. For cost and type, however, there were a few different possible designs.

For the first possibility, each type of school was given an identifying color: public schools were yellow, private schools blue, parochial schools pink, and special education schools green. Opacity was used to show the cost of the schools. The higher the cost per student or tuition rate, the darker the circles appeared. This design was fair in principle, but in practice, because the costs and tuitions were not uniformly distributed, the result did not look aesthetically pleasing or easy to understand. We

had a handful of very expensive schools that were five or six times more costly than the average school in the city, and these schools skewed the opacity scale so much that most normal schools appeared to be the same shade.

Since most colors look similar to white at low opacity levels, it was also hard to tell the different types of schools apart. Opacity was also an extrinsic property in this case, or a property that changes in effectiveness as the number of schools increased. In dense areas like lower Manhattan, it was hard for the viewer to tell whether area was dense, expensive, or both. When many translucent circles overlap, they can look like a few very opaque circles. Thus, I decided to stick with using color to symbolize the type of school, but searched for some other property that could represent cost. It was easy to pick four very distinct colors, so choosing color to represent the type of school made sense from a design perspective.

The second idea I had was to use a third dimension to represent cost. This idea was inspired by Jia Zhang's Noise Complaints map on the `youarehere.cc` website. To simulate a third dimension on the screen, I drew thin rectangles that gave off the illusion of depth in the map. For each school, I drew a rectangle starting at its location on the map to a height proportional to its cost. The problem with this method is that for densely populated areas like Manhattan, pricier schools quickly overshadowed the less expensive schools. I also had the same problem with scaling, since to fit the small number of expensive schools into the view, I had to scale down all the schools accordingly. Thus, I turned to the third and last idea: drawing circles with radii proportional to cost.

I had a few objections to drawing circles of varying radii. First, larger circles might cover smaller circles nearby, making them hard to see and unable to be hovered over. Second, the same problem of scaling still exists. In order to represent all schools, I have to draw a few very large circles on the map. I solved the first problem by drawing circles in descending order of cost. This way, smaller circles would always be on top of larger ones, so the smallest circles would be given priority even when there are multiple circles in that area. For the second problem, I realized that even though some circles look very big on the map, there weren't very many of them so it

did not detract from the readability or aesthetics of the map. Since I was able to find acceptable solutions to my main objections, I decided to stick with this third design.

Implementing the actual design was the easiest part of making this map. There were few technological challenges since most of the implementation involved placing circles on a map according to their coordinates and creating functionality to filter them by type and cost. The main bulk of the work lied in choosing an appropriate design and collecting the data.

5.4 Challenges and Future Work

As I mentioned above, the main challenges of this map included the data collection and data representation. Since there was no central database that stored all the tuition rates for every private school, at lot of the rates had to be manually collected. Choosing the right design properties for this map was also difficult. Extrinsic properties such as opacity, which tended to lose its meaning in denser areas, were not good candidates.

In addition to challenge in data collection and design, this map also suffered from the fact that its data is constantly and rapidly changing. Government-reported per-student costs at public high schools were generally accurate and, if needed, easily scraped again for more updated information, but tuition changes were harder to predict. For example, most of the tuition data I collected came from May 2014 when I first started this map. When I went back to many of these schools' websites recently, the listed tuition rate had already changed by as much as a few thousand dollars! I thought about whether I wanted to update the tuition data for the schools that I checked, but decided against it. If I did update a few schools, I would need to update all of them. Barring any mistakes in my scraping, the tuitions listed on the map were accurate for May of 2014, and thus should be compared with each other. However, future work on this map may include updating tuition and per-student costs for every school, as well as finding the tuitions for schools that don't list tuitions online.

This map was made solely for New York City, but can be reproduced anywhere

where there are many schools of different types that can be compared. I tried to do the same map for Cambridge, but Cambridge only has a handful of private and charter schools and only one main high school. Any city with many of students and more than a handful of high schools might be good candidates for a similar map comparing educational costs.

6

Implicit Distances

6.1 Motivation

New York City has one of the most extensive public transportation systems in the world. Millions of people who commute to work and school in the city rely on a complicated system of subways, ferries, buses, and commuter rails every day. This system allows people to live in all five boroughs as well as in nearby states Connecticut and New Jersey, and still get to work on time.

As public transportation lines change, so does the cultural identity of the surrounding neighborhoods. Communities like Williamsburg of northwestern Brooklyn have experienced skyrocketing housing prices and gradual gentrification because of their strong connection to popular commercial neighborhoods in Manhattan. Less accessible communities in Staten Island have remained more isolated from the rest of the hustle and bustle of New York City.

The inspiration for this map (figure A-9) came from my apartment-hunting experiences in New York City. Neighborhoods that were highly connected to other parts of the city were much more desirable than neighborhoods that were not. Thus, apartments that were close to convenient modes of public transportation were several thousands of dollars more expensive.

This map tries to visualize the connectivity of the neighborhoods of New York City using the metric of transit time, defined as the amount of time it would take to

get from one neighborhood to another via public transportation. For the rest of this chapter, when I refer to distances between neighborhoods, I will mean the relative time it would take to get from one neighborhood to another via public transportation unless otherwise stated. These implicit distances are more important than Euclidean distance in determining closeness of a neighborhood in the city. As anyone with a far commute will attest, travelling between two far neighborhoods that lie on the same subway line will always be easier than travelling between two close neighborhoods that don't.

6.2 Data

To make this map, I needed to query for the time it would take to travel between any two neighborhoods in New York City using just public transportation. Google Maps has a distance matrix API that lets the user make one query for the transit times between every origin in an origins matrix to every destination in a destinations matrix. However, at the time, the distance matrix API only allowed us to specify driving, walking, or bicycling as the transportation mode. Thus, I had to use the normal Google Maps API to query for transit times between the $\binom{266}{2}$ pairs of neighborhoods.

The Google Maps free API had a limit of 2,500 directions requests for every API key for 24 hours. I had over 70,000 queries that I needed to make. I emailed and asked for a trial of Google Maps API for work, which allows 100,000 direction requests per day, but was told that business accounts were expensive and I should just split up the requests over multiple days. With the help of my coworkers and friends, I was able to collect 15 API keys and was able to finish the scraping within one day.

The output of the API was a JSON file that I used Python and the JSON library to parse. I looked mainly for the "duration" value in the output, which was the time in seconds that Google Maps estimates it would take to get from the origin to the destination using a combination of subways, buses, ferries, and walking.

6.2.1 API Parameters

The parameters for the Google Maps API are listed below.

- **Origin** determines the origin of travel, and is one of two required parameters for the API. For most neighborhoods, appending "NYC" to the neighborhood name sufficed for the query. For a few neighborhoods, such as Huguenot in Staten Island, Google Maps searched for the wrong location, so I had to append the borough name as well. A few neighborhoods were so far away from any main lines of public transportation that Google Map results sometimes could not find a route. One place where this happened was Jamaica Bay in Queens. Jamaica Bay was located at the tip of the Far Rockaways, and Google Maps could not find transit routes between it and a few neighborhoods in the Bronx and Staten Island. For these edge cases, I calculated transit time by taking a minimum over all routes through an intermediate neighborhood that Jamaica Bay did have a connection with. One neighborhood, the Pelham Islands, did not have transit routes to any other neighborhood in the city at all, so I labeled it as such on the map.

- **Destination** determines destination of travel, and is the other required parameter for the API.

- **Mode** indicates the mode of transportation to use. Mode defaults to "driving" but I set it to "transit" in order to get the total transit time from the origin to the destination.

- **Departure time** specifies the desired time of departure. I made sure to query for transit times for the same time and day in every query so that the values returned would be comparable. I set the departure time to 1417798800, which corresponded to 12:00 PM on Friday, December 5, 2014. The total duration of a trip as returned by Google Maps does not actually include the time one would need to wait before starting their journey, however. For example, if I query for a route with a departure time of 12 PM, and the next bus arrives at my

doorstep at 12:10 PM, the extra 10 minutes of wait time would not be included in the total transit time. This makes sense, as most people do not leave for their journey exactly on the hour, and will adjust their schedules to fit their mode of transportation. Even though I made sure to query the database for the same departure time, I have found that transit times during normal business hours did not vary widely.

- **Transit mode** specifies the user's preference for bus, subway, train, tram, or rail. Since the fastest mode of transportation between any two neighborhoods may involve any or all of these five modes, I left this parameter blank.

- Finally, **transit routing preferences** specify whether or not the traveler wants to minimize transfers or walking distances. I did not use this parameter either, since I wanted to calculate the fastest route regardless of convenience. In reality, travelers probably weigh the pros and cons of convenience and time before making a selection.

6.3 Design and Implementation

6.3.1 Concept

I tried many different methods to represent my data in a way that would allow the viewer to compare implicit distances between neighborhoods. At first, I tried using the D3 Cartogram library. Cartograms distort polygons according to their values and can be powerful in maps that portray population size. I wanted to use cartograms to represent neighborhoods because as a neighborhood shrinks with respect to an origin, it gets closer to it, and as it expands, it moves farther away. For neighborhoods that are close to all other neighborhoods, the city will appear smaller, and for neighborhoods where it is hard to get to other neighborhoods, the city will appear larger. However, cartograms expand and contract maps in an unpredictable way, and the closeness of two polygons is actually dependent on the sizes of the polygons around them. Furthermore, it was hard to develop the smallness and largeness analogy, since

a highly connected neighborhood may seem closer to the rest of the city, but a place where other neighborhoods are inaccessible may seem smaller and more isolated. I decided to approach this visualization in a different way.

After experimenting with heat maps and topological maps, I finally settled on using a radial pie chart with pieces that expanded proportionally to transit time. The pie chart was made of 266 arcs representing each of the neighborhoods. The user can select a neighborhood and the radius of each arc would grow or shrink accordingly. Shorter slivers indicated a close neighborhood, and longer slivers indicated a distant one. I was happy with this representation because it portrayed the data well on both a micro and a macro scale. On the micro scale, neighborhoods were represented by slices of the pie, and on the macro scale, the general behavior of the aggregated slices gave a good idea of how connected that neighborhood is with the rest of the city.

6.3.2 Implementation

To create the radial pie chart, I adapted an existing pie chart from a D3 library. While normal pie charts have multiple arcs that occupy a different angular sector of the pie, this adapted pie chart has arcs that have the same width but different height.

To supplement the pie chart, I also designed a linear timeline where neighborhoods were placed as points on the timeline depending on how far away they were. This graph aimed to give a physical sense of comparative distance. To create the timeline, I adapted a D3 timeline and placed neighborhoods on the timeline with a time value equal to the time it would take to reach that neighborhood.

6.4 Challenges and Future Work

So far, this map has only been made for New York City. The map was challenging to produce because of difficulties in collecting and displaying the data. However, conceptually, it is very simple and can be adapted for any city that has well-defined neighborhoods connected by a public transportation system. Sites like Zillow.com have free neighborhood boundaries data for many cities across the United States. In

addition, the Google Maps API now allows users to query for transit time using a directions matrix, which will drastically cut down on the amount of time it would take to the gather data.

It is difficult to see whether or not this map is actually helpful in differentiating connected locations versus unconnected ones. For example, when I query Google Maps for the fastest public transportation route between two points, it returns results that do not differentiate between walking and taking the subway. However, most travelers would say that a one-hour subway ride is much more preferable to a one-hour walk. In the future, it would be interesting to query Google Maps solely for the total walking time needed to travel between two neighborhoods. That data could be supplemental in determining how convenient and connected a neighborhood is.

Finally, this implicit distances map does not distinguish between populated and unpopulated neighborhoods. A neighborhood may be highly connected to many other neighborhoods, but if it is far from any centers or hubs of activity, this metric of closeness may not be as meaningful. Knowing this, I decided to weigh average transit time by neighborhood population for my sixth map, which I will describe in detail in the next chapter.

7

Connectedness

7.1 Motivation

The inspiration for this map came directly from the Implicit Distances map of the previous chapter. Unlike the previous map, however, this map will aim to visualize the average connectedness of a neighborhood with the people around it. A neighborhood may appear close to another neighborhood, but if no one lives in that neighborhood, it means very little. Of course, with the exception of Central Park, very little of New York City is uninhabited, but we can still gauge the average number of personal connections by weighing each neighborhood by its population. This map can then be used to determine the population hubs in a city, defined as the neighborhoods with the highest connectivity with other densely populated areas of culture and activity.

7.2 Data

Most of the data used in this map is the same as the transit data used in the previous map. The techniques used to scrape and clean this data were described in detail in the previous chapter. Other than that, I also needed a database of the populations of every neighborhood in New York City. The Census Bureau had data for some of the larger neighborhoods like Flushing, Jamaica, and Williamsburg. However, smaller neighborhoods like Little Italy, Soho, Tribeca, and Civic Center were grouped together

into one census block. I estimated populations for these neighborhoods simply by taking the per-neighborhood average for these groups of up to four neighborhoods. Since the neighborhoods that were grouped together tended to be right next to each other, these estimates did not detract too much from the idea of connectedness and were probably fair to make for a densely populated city like New York.

7.3 Design and Implementation

Designing and implementing this map were both straightforward. At first, I looked into using a topographical map to represent the data. Since neighborhood boundaries were continuous, it made sense to create a terrain map where high peaks represent difficult-to-get-to areas. However, I settled with a heat map of the city. While implementing the heat map, I ran into a similar problem that I had faced before: there were one or two outliers in the data, so when I normalized by the range of the data, most of the neighborhoods ended up with indistinguishable colors on the map. To combat this, I turned away from using opacity to represent darkness and implemented a color bucketing system designed to maximize the differences between regions.

7.4 Challenges and Future Work

At first glance, it seems that the average transit time of a neighborhood is highly correlated with the distance from that neighborhood to the center of the city. After all, neighborhoods that are close to the center have a shorter average Euclidean distance to all over neighborhoods in general and would predictably have shorter transit times. In reality, not every neighborhood followed this trend, and this map was helpful in pointing out various pockets of high and low connectivity in unexpected places. For example, although various neighborhoods in Staten Island are equally distant from Manhattan as the farthest neighborhoods in Queens, the neighborhoods of Staten Island are far more disconnected. Similarly, although the islands of the Far

Rockaways in Queens are relatively unconnected via public transportation, they are close enough to major hubs in the city that they are never fully isolated. In general, however, the criticism remains true. Future work could be done to normalize transit times with respect to Euclidean distance in order to find out which neighborhoods have connectivity levels that could not have been predicted by its placement on the map.

8

Conclusion

"What strange phenomena we find in a great city, all we need do is stroll about with our eyes open. Life swarms with innocent monsters."

– Charles Baudelaire

Cities are wonderful places in which we live, hope, and dream. A city is a microcosmic approximation of both the world and our minds, and we can learn as much from it as it can learn from us. That is why the intersection of big data and city planning is exciting, because there is no better way to improve the lives of billions of people who currently live in one of the world's many urban sprawls than to hear what a city has to say and then making it come true. I hope that my project has shed some light in how even the simplest of datasets can send powerful messages of action when viewed in the right way.

Appendix A

Figures

Figure A-1: Immigration by Sea, New York City

Figure A-2: Interactivity - Immigration by Sea, New York City

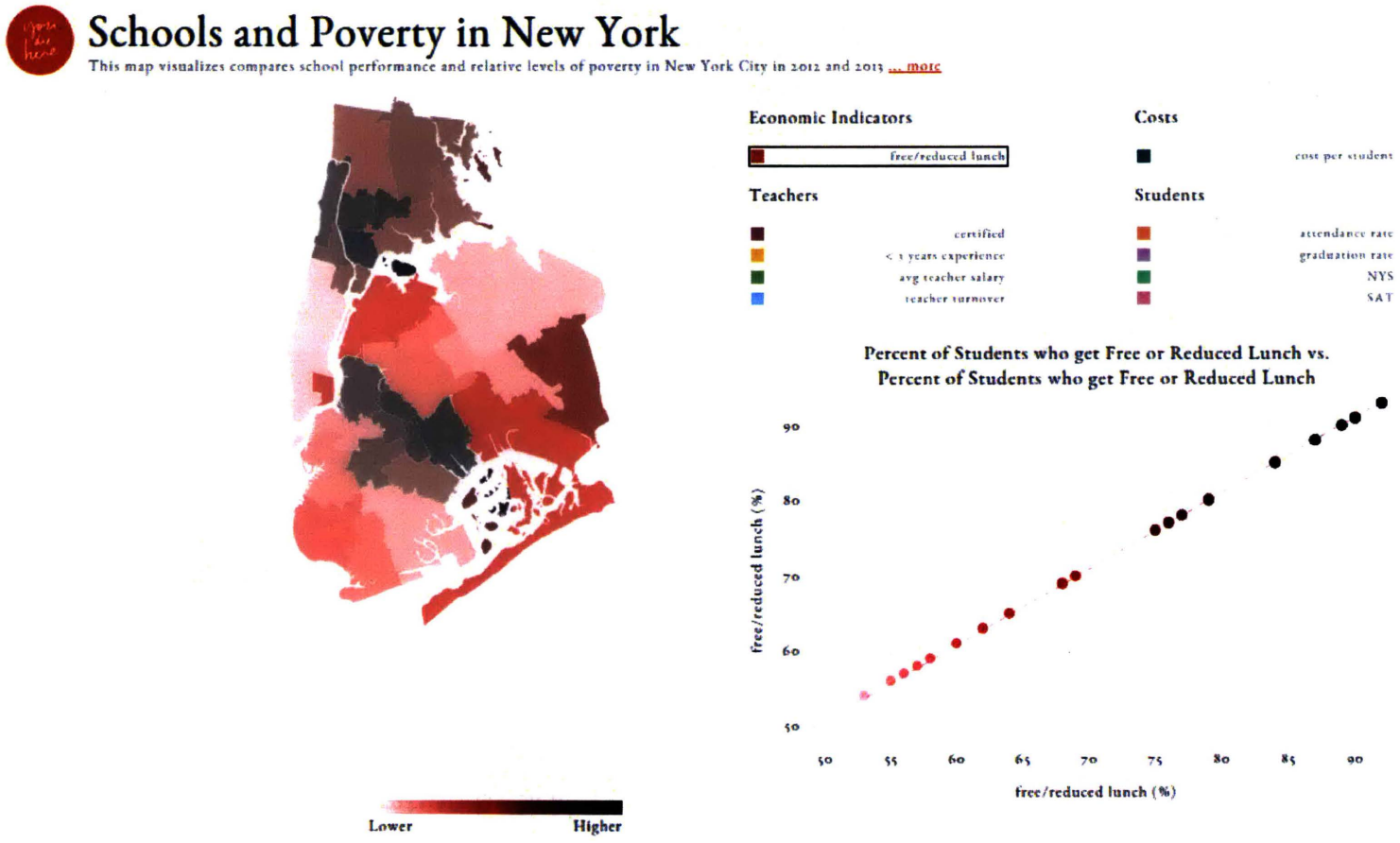


Figure A-3: Poverty and Schools, New York City

Figure A-4: Interactivity - Poverty and Schools, New York City

New York City High School Dropouts

Nearly 20,000 students drop out of New York City schools every year. This animated map runs through an average 180-day school ... more

This work is part of the You Are Here project + The Social Computing Group + MIT Media Lab + Massachusetts Institute of Technology

Figure A-5: Dropouts, New York City

Figure A-6: Interactivity - Dropouts, New York City

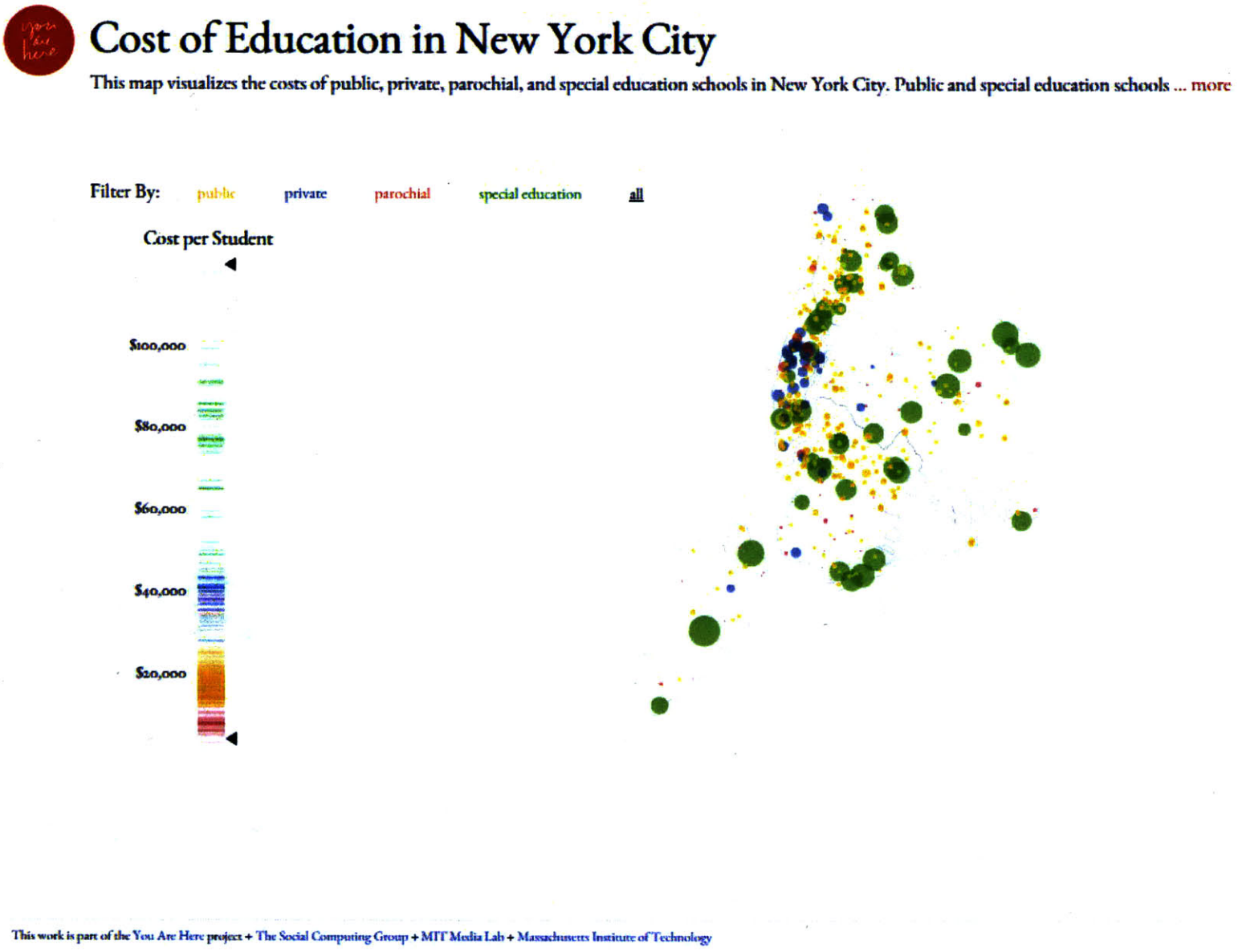


Figure A-7: Costs of Schools, New York City

Figure A-8: Interactivity - Costs of Schools, New York City

Figure A-9: Implicit Distances, New York City

Figure A-10: Interactivity - Implicit Distances, New York City

Bibliography

[1] Mike Bostock, http://bost.ocks.org/mike/. February 12, 2013.

[2] Allan L. Damon, *11 Facts About High School Dropout Rates.* https://www.dosomething.org/facts/11-facts-about-high-school-dropout-rates.

[3] *Fast Facts.* http://nces.ed.gov/fastfacts/display.asp?id=16. National Center for Education Statistics, 2015.

[4] Allan L. Damon, *A Look at the Record: The Facts Behind the Current Controversy Over Immigration.* http://americanheritage.com/immigration/articles/magazine/ah/1981/1/1981_1_50.shtml. American Heritage Publishing, 1981.

Stephen P. Morse, *One-Step Webpages by Stephen P. Morse.* http://www.stevemorse.org/.

[5] *New York City Department of Education.* http://schools.nyc.gov/default.htm. 2015.

[6] *New York State Department of Education.* http://www.nysed.gov/. 2015

[7] *NYC Open Data.* https://nycopendata.socrata.com/. 2014.

[8] *Researching Your FamilyâĂŹs History at the Massachusetts Archives.* http://www.sec.state.ma.us/arc/arcgen/genidx.htm.

[9] *Ancestry.com.* http://www.ancestry.com/. 2015.