CityView: Urban Data Visualization

Major Qualifying Project

Written By:
Carlos Barcelos
Justin Myerson

Advised By:
Yanhua Li, Ph. D
Xiangnan Kong, Ph. D

A Major Qualifying Project
Worcester Polytechnic Institute

Submitted to the Faculty of the Worcester Polytechnic Institute
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in Computer Science.

23 March 2018
Abstract

The goal of this project is to create a website that would give users the power to analyze datasets of varying spatial-temporal categorization while providing them with real-world scenarios in which this system could be used for urban development. We accomplished this goal through data visualizations, data tools, and data stories which are all displayed to the user in a clear and informative manner. We believe that CityView is a customizable system which can be successfully expanded upon and we suggest possible directions in which to make that expansion.
# Contents

Abstract ........................................................................................................................................... 1

Table of Figures .............................................................................................................................. 3

1 Introduction .................................................................................................................................. 4

2 Background ................................................................................................................................... 5

   2.1 Urban data ............................................................................................................................... 5

   2.2 Prior Work .............................................................................................................................. 5
       2.2.1 The SURV System ........................................................................................................... 5

   2.3 Existing Systems .................................................................................................................... 7
       2.3.1 UN-Habitat Urban Data .................................................................................................... 7

   2.4 Technologies ........................................................................................................................ 10
       2.4.1 Mapbox ........................................................................................................................... 10
       2.4.2 Leaflet ............................................................................................................................ 10
       2.4.3 CARTO .......................................................................................................................... 10

3 Methodology .............................................................................................................................. 11

   3.1 Goals ..................................................................................................................................... 11

   3.2 Design .................................................................................................................................... 11

   3.3 Implementation ...................................................................................................................... 12
       3.3.1 Utilizing Other Resources ............................................................................................. 12

   3.4 Analysis with Data Stories ................................................................................................. 17
       3.4.1 Static Data ...................................................................................................................... 17
       3.4.2 Time Series Data ............................................................................................................ 20
       3.4.3 Dynamic Data ............................................................................................................... 22

4 Results ......................................................................................................................................... 25

   4.1 Usage ................................................................................................................................... 25

   4.2 Results .................................................................................................................................. 25

5 Conclusion ................................................................................................................................... 26

   5.1 Future Work ........................................................................................................................... 26

References ....................................................................................................................................... 29
Table of Figures

FIGURE 1 UN-HABITAT URBAN DATA SELECTION AND VISUALIZATION INTERFACE .................................................................7
FIGURE 2 THE WATER PLANNING VISUALIZATION BY THE URBAN VISUALIZATION LAB ..........................................................8
FIGURE 3 THE TRANSIT QUALITY & EQUITY VISUALIZATION BY THE VUD LAB ........................................................................9
FIGURE 4 GITHUB’S ISSUE TRACKING SYSTEM WITH AN EXAMPLE TICKET .............................................................................11
FIGURE 5 THE CITYVIEW SYSTEMS PROJECT FILE ORGANIZATION ......................................................................................12
FIGURE 6 MAPBOX IMPLEMENTATION OF STATIC DATA ........................................................................................................13
FIGURE 7 MAPBOX IMPLEMENTATION OF TIME SERIES DATA ..................................................................................................13
FIGURE 8 LAYER SELECTION FEATURE FOR THE TIME SERIES DATASET .................................................................................14
FIGURE 9 MARKERCLUSTER IMPLEMENTATION OF TIME SERIES DATA AT A FAR ZOOM LEVEL ....................................................15
FIGURE 10 MARKERCLUSTER IMPLEMENTATION OF TIME SERIES DATA AT A MEDIUM ZOOM LEVEL ....................................15
FIGURE 11 MARKERCLUSTER IMPLEMENTATION OF TIME SERIES DATA AT CLOSEST ZOOM LEVEL ........................................15
FIGURE 12 VISUALIZATION BUTTON ALLOWING USER TO SWITCH BETWEEN IMPLEMENTATIONS ..............................................16
FIGURE 13 CARTO DATA OPTIONS INTERFACE .....................................................................................................................16
FIGURE 14 FINAL CARTO DATA VISUALIZATION .....................................................................................................................16
FIGURE 15 A SMALL SAMPLE OF BOSTON'S STREETLIGHT DISTRIBUTION ..............................................................................17
FIGURE 16 THE ROUND HILL/EDGE HILL STREET NEIGHBORHOOD MAY BE POPULATED BUT IT IS ALSO POORLY LIT .............18
FIGURE 17 USING CONTOUR LINES HELPS TO AVOID PLACING BIKE STATIONS ON HILLS THAT ARE TOO STEEP ..................18
FIGURE 18 USING ALL AVAILABLE DATA IN CONJUNCTION CREATES A CLEARER PICTURE OF THE DESIRED SOLUTION ......19
FIGURE 19 ALL DAYS OF THE WEEK SHOWN IN ONE VIEW .......................................................................................................20
FIGURE 20 ONE DAY OF THE WEEK (TUESDAY) SHOWN TO VISUALIZE COVERAGE ON A GIVEN DAY ....................................20
FIGURE 21 VISUALIZING THE REDOUBLING OF RESOURCES ON WEDNESDAY AND FRIDAY COVERAGE ..........................21
FIGURE 22 THE USER CAN VIEW THE TRADITIONAL PICKUP/DROPOFF MAP OR VISUALIZE PICKUP AND DROPOFF DATA SEPARATELY WHICH CAN HELP IN DISCOVERING A CLEARER PICTURE OF THE DATA .................................................................22
FIGURE 23 PICKUP DATA POINTS ARE SHOWN IN RED ................................................................................................................23
FIGURE 24 DROPOFF DATA POINTS ARE SHOWN IN BLUE ..........................................................................................................23
FIGURE 25 ISOLATING A CERTAIN TIME PERIOD AND COMPARING THE PICKUP/DROPOFF FREQUENCY OF THREE LOCATIONS ....................................................................................................................................24
1 Introduction

In today’s industries, collecting data is becoming increasingly important and more widely utilized. Both the number of devices which can collect data and the types of data that are being collected are expanding rapidly. This effect is magnified in dense, urban environments. This leads to more data being created daily than existed ever before. While it may appear that more is better in terms of data aggregation, this is not necessarily true. In order to make sense of this data and give it practical meaning the data must be analyzed systematically and interpreted.¹

It is important that there are systems that can help decision makers visualize the data that they are gathering. It is no surprise that big urban data is not useful on its own but must be used with “thoughtful care”.² There are existing systems that visualize data, but few that are readily available to help the user understand the visualization that they are being presented. For the data visualization to make an impact in affecting change, the decision maker must understand the implications of the visualization. CityView aims to deliver a clear visualization with detailed information that are presented with thoughtful care.

CityView is a web-based resource designed to analyze large sets of urban data. CityView has a clear and informative interface that can be used by an individual doing research or an organization trying to make a decision in an urban environment. The tools provided offer a framework for future experimentation. The core website offers unique visualizations and clear maps in an easy to use and accessible package. We designed CityView to utilize pre-existing systems - Mapbox, Leaflet, and CARTO - and leverage their APIs to make a cleaner and more informative experience.
2 Background

2.1 Urban data

In today’s increasingly high-tech world, large amounts of data can be found for most topics of interest. This data is collected by companies, individuals, and, in this case, even cities. The city of Boston is just one of many major cities that has a website dedicated entirely to interesting datasets that can be used by anyone from home enthusiasts to professionals. The Boston city website has categories of datasets ranging from financial, to geospatial, to city services. While the datasets are readily available, the visualization and interpretation of this data is not. In fact, the Boston website has a section reserved for third-party projects creating interesting and useful visualizations since the city itself cannot feasibly do so with the amount of data collected.

The urban data that cities collect can be categorized into three combinations of spatial-temporal data. These pairs are generally defined by a geographical location and a datetime. More specifically, \textit{spatial-temporal} can be broken down into its two parts. Spatial refers to the location of the data point, generally defined by an ordered pair: (latitude, longitude). Temporal refers to the time at which the data point was collected, often in the form: \textit{yyyy-mm-ddTh:mm:ss}. For the purposes of this project, we will consider the following combinations of spatial-temporal data:

- Spatially-Static and Temporally-Static Data will be referred to as Static Data
- Spatially-Static and Temporally-Dynamic Data will be referred to as Time Series Data
- and Spatially-Dynamic and Temporally-Dynamic Data will be referred to as Dynamic Data

2.2 Prior Work

2.2.1 The SURV System

The precursor to the CityView system was the SURV system developed by a group of REU students and WPI personal. The SURV system was a basic urban data visualization system comprised mostly of HTML

\footnote{Datasets with the combination of spatially-dynamic and temporally-static are impossible to collect, as this type of data implies instantaneous change, which is not feasible for physical data sets.}
and JavaScript. The SURV site has four main features: Data Animation, 3D Visualization, 3D Visual Interaction, and Data Upload Example. Many of the features and technologies found in SURV were a great inspiration when creating CityView.

However, not all the features functioned correctly and were often difficult to decipher how they worked. This influenced our decision in creating a system which was fully functional, favoring the omission of features from the final release unless they were fully complete, and would be easily digestible by a future developer. We wanted to improve what the SURV team had already created while expanding on the initial work presented.

**Data Animation**

The first part of this system is the data animation feature. This animation was developed with the help of CARTO to show a time-sensitive heatmap of a single data set. By using CARTO, spatial-temporal data can be read in as a GeoJSON file and displayed dynamically on a map. The data, taken from taxi rides in Shenzhen, China, contains a timestamp, longitude and latitude information, instantaneous velocity, and other small details. For the purpose of the Data Animation demonstration, only the location and time information were used in order to create a heatmap of the taxis over time. CARTO’s API allows the animation to extend for the entire dataset or be selected for a particular block of time. The API also includes other interesting analysis tools which can be used to further explore this Data Animation.

**3D Visualization**

The second part of this system is a 3D visualization page that utilizes MapBox to show a static heatmap of a single data set. From a CSV of longitude and latitude data, the dataset is converted to a GeoJSON file. The data is then clustered and passed through an algorithm to produce a heatmap of location data. Different than the previous data set, this spatial data provides information about vehicle density within the streets of the city.

**3D Visual Interaction**

The third part of this system is a 3D visualization page that allows the user to change the time frame associated with the data, as well as the day of the week. At the time of writing, this part of the system does not work. When loading the webpage, the GeoJSON file created is too large for the browser to handle and continuously freezes before crashing due to running out of memory.
When working, this demonstration would act much like the 3D Visualization page with the addition of a time scale and the inclusion of the “speed” property. This demonstration acts much like the CARTO visualization with only the resources of MapBox behind it. In order to utilize this feature, it is necessary either to decrease the computational cost of loading such a massive dataset or segment the data set such that only part of it is active at any given point in time.

Data Upload Example

Finally, the system includes a page that allows the user to upload their own set of data and have the site present a heatmap of their data. This tool is limited to rendering the heatmap of one dataset at a time but allows for greater control by the user by allowing them to upload any spatial dataset. This feature was not fully completed when the final project was made public.

2.3 Existing Systems

There are many existing systems whose goal is to visualize urban data. Those systems all have unique goals and visualization styles and tools. Below are a few systems worth mentioning:

2.3.1 UN-Habitat Urban Data

UN-Habitat created a dynamic data visualization system. The interface, seen in Figure 1, allows users to select data from certain years, region, and indicators which create unique visualizations that can be helpful in making sense of urban data. Those indicators range from population to transportation giving the user a great level of control over the results they see. All the data used for the visualization can also be downloaded for individual use. The most lacking area of this system is that it does not provide clear use cases for the visualizations that it is able to produce.

![Figure 1 UN-Habitat Urban Data selection and visualization interface.](image-url)
2.3.2 Visualizing Cities

Visualizing Cities is not a visualizing platform itself but an aggregation of existing urban visualization projects. The platform is made to “bring together urban visualization projects from around the globe” and does so using a variety of visualization mediums: animations, infographics, animations, physical installations, etc. The system itself does not have an interactive visualization interface and the associated projects are largely hidden in academic publications. It is a good inspiration for style but not replicable in any way. During our initial research, this system influenced our decision in making our data and visualizations as accessible as possible.

2.3.3 University Based Laboratories

Individual universities have proprietary data visualization laboratories. While the visualization systems used in these labs are not publicly available it is important to mention them for completeness and inspiration. The Urban Data Visualization Lab out of the University of Illinois has completed many projects after collecting local urban data. The projects vary in visualization style and functionality; take the Water Planning project, seen in Figure 2, for example which includes an intricate map of the area and waterways. This project provides inspiration on how to visualize waterways in a region based on location, geography type, or subject area.

![Figure 2 The Water Planning visualization by the Urban Visualization Lab](image)
The VUD Lab out of UC-Berkeley is an internal organization which helps students learn and create visualizations to clarify urban data. While their technology exists within the university, their project portfolio is made public and provides inspiration for work in urban data visualization. An example of their work is the Transit Quality & Equity visualization seen in Figure 3. The visualization overlays transit data and income levels to find the equity impacts of transit service.

While these projects may be proprietary to their respective universities and not replicable, the results are still important. They can be used as a starting point for new work and offer inspiration for the applications of urban data that may not have otherwise been obvious.
2.4 Technologies

In order to build the system, we made use of several resources in addition to basic programming. The main part of the website is comprised entirely of HTML, CSS, and JavaScript. To build the visualizations, we used several other APIs from open-source JavaScript libraries and plugins.

2.4.1 Mapbox

Mapbox is a location data platform available for use with several programming languages, including JavaScript. Mapbox works by plotting longitude and latitude points onto a geocoded map layer to design and create maps. As discussed in 3 Methodology, we used Mapbox for the first version of our visualizations as it was easy to implement and had many of the features we wanted.

2.4.2 Leaflet

Leaflet is a JavaScript library for making interactive maps. It can interface with Mapbox for the base map layer but is also usable with numerous other map creation services such as OpenStreetMap. In addition to the standard library, Leaflet offers a growing list of plugins to add enhancements to map interactions, most notably Markercluster, which we used to power our spatially static data visualizations.

2.4.3 CARTO

Carto is a cloud computing platform that provides web mapping tools that can be displayed in a web browser. CARTO is great for displaying trajectory data (data that is displayed in motion over time), whereas Mapbox and Leaflet are best for creating interactive visualizations of stationary data. For this reason, we used CARTO for the dynamic dataset in our system.
3 Methodology

3.1 Goals

The goal of this project was to create a website that would analyze and correlate several sets, of varying spatial-temporal categorization, while providing a real-world scenario in which this system could be used for urban development. To accomplish this goal, we developed CityView with the following requirements:

- The web interface must have features which allow the user to interact with the data animations to provide better them with data analysis.
- The website should provide depth and clarity surrounding the data types, featured datasets, and use cases for each data visualization.
- The website must be able to handle reasonably large data sets to provide robust visualizations which accurately represent dense urban environments.
- The final project must allow for future experimentation. This means code that is well organized, well documented, and easily readable by future experimenters.

3.2 Design

The design of CityView revolves around the different types of spatial-temporal datasets we utilized for the system. Each type (Static, Time Series, and Dynamic) has its own section of the system with visualizations specifically tailored to that type of data. To build the system itself, we used GitHub Pages hosting to keep a live version of the site running at all times. This also allowed for continuous integration through versioning control on the repository. Furthermore, we made use of GitHub’s issue tracking feature to create one-week “sprints” between weekly meetings, similar in style to what one might see at a software company. The intent was to have as much of a full-fledged development cycle as possible for this project to ensure both requirement completion and minimization of system-breaking bugs - nothing was pushed to master and all pull requests required an outside review.

Figure 4 GitHub’s issue tracking system with an example ticket
The website and its features are made up of a combination of HTML, CSS, and JavaScript in different areas. The basic site pages consist of primarily HTML and CSS with most of the framework coming from Bootstrap\textsuperscript{xiv}. Bootstrap was a critical framework allowing us to quickly create navigation bars, dropdown lists, and other page elements which allowed us to focus more on creating the data visualizations and features. The visualizations themselves are done with JavaScript, and make use of several third-party APIs: Mapbox, Leaflet, Canvas, Markercluster and CARTO.

To reach our design goal of having the final project be ready for future experimentation, we designed the website with a simple and clear architecture. The final file structure is outlined in Figure 5. The entire website directory is small and lightweight, relying on CDNs where it can to reduce the server load allowing each webpage to focus on rendering its visualization instead of loading scripts and fonts. The package structure within the directory is simple and understandable. The clear file and folder names allows easy navigation for any future parties that might expand on this project.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5}
\caption{The CityView systems project file organization}
\end{figure}

3.3 Implementation

Originally, the system was designed to operate entirely off the CARTO visualization architecture to offload most of the processing of the data. While this was an easy way to visualize our data, we quickly hit a wall in terms of features, customizations, and performance. We kept the CARTO implementation for the dynamic data as we found it to be the most effective resource for that dataset but switched to other integrations with the static and time series data.

3.3.1 Utilizing Other Resources

The datasets that are used for our visualizations were visualizations were taken from publicly available city data provided by the cities of Boston\textsuperscript{xv} and New York\textsuperscript{xvi} government websites. Our first three datasets to be used in a visualization were the Boston streetlight locations, Boston Hubway bike share bike station
locations, and geographic contours around Boston. Together, they formed the static data visualization. Our second dataset was the weekly schedule of recycling collection in the city of Boston. That dataset formed the time series visualization.

Mapbox

For both datasets, our first implementation was with Mapbox, which is a JavaScript add-on for geography-based visualizations. Each dataset is a large CSV file with information that designates which “layer” a data point will be displayed on, as well as latitude and longitude values. In Mapbox, and indeed all of the other APIs we used, a layer represents one dataset in the overall visualization. This distinction between layers and the ability to toggle them on and off is one of the critical features of this system. The result of the Mapbox implementation was a well-performing, clean visualization as seen in Figure 6 and Figure 7. But Mapbox did not provide enough features for our needs. In order to meet all of our goals, we needed to use something that had the desired features customizability.

Figure 6 Mapbox implementation of static data
Figure 7 Mapbox implementation of time series data
The next iteration of the implementation was done with Leaflet. It is the framework for Mapbox which provides a similar variety of functionality for geographic visualizations. Leaflet also allowed us to implement layer toggling within the base maps and datasets. Toggling between base maps gives the users more options in the way of visible accessibility, while toggling between datasets allows users to create custom visualizations.

Additional features included dedicated buttons for zooming, an information box popup button that gives an overview of the visualization to the user, different map background styles, and the “carousel” feature. The carousel feature cycles through all of the layers presented on the map in a sequence, giving the impression of time series data. As a result, we used this feature for the recycling collection schedule as it gave a good day-by-day representation of the data.

While Leaflet offered us many more features and customization options, the problem was the performance. Initially, we experienced significant issues with dark shadows showing up on the map and interfering with the visualization since the points were plotted as an image with a dark background. This image was taken from the Mapbox servers and it was not possible to apply the individual points served from Mapbox, it was only possible to apply the image of the dark layer of points. This was a nearly unusable visualization.

The solution to the Mapbox image layer was to use Canvas, a Leaflet feature, and D3\textsuperscript{xvii}, an additional JavaScript library. Together, they provide the most customizable and feature-rich visualization implementations. We used D3 to read in the CSV files with the pertinent data to the visualization. Then, the information from that CSV file was plotted to the map through Canvas. The problem with this solution

\footnote{We would like to acknowledge Shijian Li for help with Canvas and Markercluster solutions.}
was the performance. Because each data point was a single entity which must be dynamically rendered on the map the visualization became computationally intensive. So computationally intensive that we needed to find an alternative solution.

**Markercluster with D3**

Markercluster is a plugin for Leaflet that allows customizable aggregation of data points at different zoom levels on the map. The result is that we were able to combine the highly customizable qualities of Leaflet with a better performing display of the data. Like the Canvas solution, we used D3 to load in all the data points from a CSV file. However, with Markercluster we did not need to render them all on a map at the same time. As the user zooms in, each cluster breaks apart into smaller subsets of clusters at predefined zoom scales. As the user zooms all the way in, the clusters eventually break apart into the individual data points. As the user zooms out, the opposite happens, with clusters forming labeled with the number of data points contained in them. The noticeable performance improvement stems from the fact that at any given zoom level, the system is only displaying a small subset of the overall dataset, whereas with the other implementation, hundreds of thousands of data points are plotted right from the beginning.

![Figure 9 Markercluster implementation of time series data at a far zoom level](image1)

![Figure 10 Markercluster implementation of time series data at a medium zoom level](image2)

![Figure 11 Markercluster implementation of time series data at closest zoom level](image3)
In the end, we chose this implementation as our primary visualization. We also included buttons on the visualization page to allow the user to switch between each of the Mapbox and Leaflet iterations previously discussed as seen in Figure 12. We believed that this was important in preserving the history of the work done and giving users the ability to compare visualization methods.

For the dynamic data visualization, we used CARTO. This visualization used a single include line of HTML and CSS formatting. The rest of the work was accomplished through CARTO’s web interface. An important first step was to clean our data, making sure there were columns in the CSV for latitude, longitude, and date-time. There was also a column labeled either “pickup” or “dropoff” depending on which the datapoint represented. We then uploaded our dynamic data to the CARTO servers and used their tools to complete our final visualization. The result was an animated aggregation that populated the map for every hour of the date-time column. Finally, unique colors were assigned to pickups and dropoffs and the data visualization was complete. The final visualization can be seen Figure 14.
3.4 Analysis with Data Stories

The process of implementing our design decisions into CityView involved analyzing its functionality. Because it was important that this would be a useful system for individuals looking to analyze urban data we took it upon ourselves to create specific data stories for each of the Static, Time Series, and Dynamic data visualizations.

3.4.1 Static Data

Contour Lines / Streetlights / Hubway Stations

Imagine a bike share program interested in creating new locations for their existing bike share network. In this example, consider the city wide Hubway Bike Share in Boston, MA. Using just three datasets and our mapping features, we can give recommendations on where and where not to place new Hubway bike stations.

Streetlights

First consider the safety of a bike rider - new bike stations should be placed in well-lit areas. Using a data set of all street lights in Boston, we can cross-reference areas that are more or less safe for new bike rental stations.

Figure 15 A small sample of Boston’s streetlight distribution
Using this view gives a high-level view of the city. For a more useful application we could use this view to ask, “Is this area a well-lit location for a new bike station?” For example, there are not any street lights in the neighborhood of Round Hill or Edge Hill street. The bike share manager may have thought this would be a successful location due to the high population of the residential area, however our urban data visualization shows that putting a bike station in that area may lead to decreased rider safety and increased bike theft under the veil of a dark night.

Contour Lines

Next consider the users that will have to ride the bikes in their physical environment. One common problem that bike share programs have is that users will avoid areas with steep hills. By adding a contour layer, we can get an idea about the environment and make decisions based on the elevation of the land.

For example, consider adding a new bike station at New England Baptist Hospital. The bike share manager may initially think that a hospital would be a popular place for a new bike station; the hospital may promote healthy transportation and there are a lot of employees who might benefit from riding a bike to work. However, when considering the terrain, New England Baptist Hospital is on a steep hill due to the frequent and closely packed contour
lines. Using the contour data, it the bike share manager realizes that the hospital is on too steep of a hill to make it a successful bike station.

All Together

All of this data is useful individually but more powerful all together. By combining the three layers of data on one map we can show a more complete picture of the urban environment. Now the user can find the best location a new bike station, considering well-lit areas and low elevations. Furthermore, they can ensure that the locations of bike stations are neither too close together - causing redundant stops - nor too far apart - causing long uncomfortable rides - from one another.

Figure 18 Using all available data in conjunction creates a clearer picture of the desired solution
3.4.2 Time Series Data

*Recycling Data (Monday - Friday)*

Imaging a recycling company interested in understanding how their teams move throughout the city during the week. The company can use GPS locations to track each of the houses that their trucks visit during each day of the week. Using this data, they would be able to visualize the time series distribution of their resources on our website. In our example, we used existing recycling data recorded during each weekday. Five unique colors differentiate each day of the week. Within each day, each data point shows a location that was visited by the recycling company.

The recycling company can use this visualization to make some important discoveries. For example, imagine the company invests in a new truck and needs to consider which area to send that truck. They can use our system to view each day’s data one at a time to find the area in the city that has the most stops with relatively few trucks visiting those stops. By doing this sort of analysis, the recycling company can better distribute their trucks in order to maximize coverage in the city.
Furthermore, using our visualizations the recycling company can determine the best route for their trucks on any given day. If the company has a large fleet of trucks, the best option they have is to spread out their trucks throughout the city and have those trucks service large neighborhoods in one day of operation. By separating the trucks, the company can minimize the amount of traffic that they cause in the city. By servicing entire neighborhoods at a time, the company can alleviate the burden on a neighborhood by visiting the area only once.

Using our visualization tools, we can see one area in which this best practice is violated. Using the data from the Wednesday (purple) and Friday (red) datasets, we can see a situation in which the recycling company crosses the large Arborway highway unnecessarily because they fail to service the entire neighborhood on the east side of Washington Street. By simply visualizing this data we can recommend to the recycling company that they should consider servicing the Washington Street neighborhood entirely on Friday to both optimize their coverage in that area and alleviate any traffic congestion they may cause.
3.4.3 Dynamic Data

**NYC Green Taxi Cab Pickup/Dropoff**

Imagine New York Green Taxi drivers, who are interested in maximizing their profits. To do so they need to understand their customers’ habits by knowing where and when to make pickups and dropoffs. Using just one week’s worth of pickup and dropoff location data we can make a few conclusions and suggestions for these taxi drivers.

*Figure 22 The user can view the traditional Pickup/Dropoff map or visualize Pickup and Dropoff data separately which can help in discovering a clearer picture of the data*
Pickup Limitations

Pickup points are heavily concentrated closely outside of Manhattan island but cannot be found on the island itself. This is due to the policies that the Green Taxi drivers must abide by. This practice is a disservice to an individual taxi driver. It means that the taxi cabs are not being utilized in one of the most inhabited locations in the city. It is well known that many people in the heart of New York City do not own cars, and therefore would benefit from taking a taxi from their home in the city to someplace outside of the city. If the Green Taxi company were able to renegotiate their policies by allowing its drivers to make pickups in the city, they could greatly increase their efficiency and profit.

Dropoff Efficiencies

Dropoff points have a much more even distribution through the city, even finding their way onto Manhattan Island. This visualization helps us make our next recommendation to the taxi company; continuing under the restriction that Green Taxi drivers cannot make pickups on Manhattan Island, drivers should be more efficient with their time by also restricting their dropoffs to outside of Manhattan Island. Doing so would increase the number of rides each taxi can complete by a factor of two because each off-island pickup would be followed by an off-island dropoff - and so on - without any intermission without a passenger in the cab.
Restricted Time

Now imagine a driver who shares the ownership of a particular Green Taxi cab and has no control over the time during which she can drive it. Her designated hours are from 4pm - 12am. Because she is able to decide *where* to drive the cab, she is able to use our visualization to pick the busiest area to drive during her shift.

To complete this the driver would select the *Pickup/Dropoff* visualization and then identify three areas in which she may be interested in driving: for this example Brooklyn, Queens, and Bronx. Zooming into each area causes the visualization to display data points only within the bounding box of the display. With this the driver can discover that over one week there are 109, 17, and 36 pickups and dropoffs in Brooklyn, Queens, and Bronx respectively during her 4pm - 12pm shift. With this information, she can determine that her best chance at success is to spend her shift in Brooklyn.

![Figure 25 Isolating a certain time period and comparing the pickup/dropoff frequency of three locations](image)

Extensions

This example was created using only one week’s worth of data for one taxi company. Using a longer time frame for visualizing data would not only show daily/hourly patterns but could be used to show monthly/yearly patterns. Do passengers use taxis less in the summer months when the weather is nicer? Do passengers use taxis later in the summer months when it is brighter outside for longer? These questions can be answered by extending the time frame of the visualized dataset. Doing so would help the taxi company determine how many taxi drivers they need to hire during various parts of the year.

Comparing and contrasting data from two taxi companies would be useful as well. Do passengers prefer one company over another? Does each company have a portion of the city they frequent more often? Is one company better for longer trips while another is better for shorter trips? These are all questions that could be answered using our visualization. Doing so would help one company compete with another. They could make decisions based on the visualization to optimize their presence in the city.
4 Results

4.1 Usage

CityView is available for anyone to use through a web interface. The system was designed to analyze large urban data; the application of the analysis is up to the user’s discretion. We have outlined in 3.4 Analysis with Data Stories specific ways in which the system can be used, but the functionality does not end there. CityView puts interesting analysis tools in the user’s reach to power urban planning through rich data.

4.2 Results

The completed system contains most of the features we aimed for when setting goals for the project. In particular, a feature-rich system for visualizing urban data in order to discover patterns and important information. While we were able to handle reasonably large data sets, there is still room for improvement. This goes for features available to the user as well. While we were able to create useful features, notably the carousel feature, we believe there is still room for improvement.

Both the Static and Time Series data visualizations offer a robust toggling feature. Users can select which datasets they want to visualize to get a customizable view at their data while selecting the map type as an additional accessibility option. In our Static Data visualization this allows the user to find effective and efficient new locations for bike share hubs around the city of Boston.

The time series data visualization makes use of Leaflet’s customizability by introducing the carousel feature. This visualization can help to optimize recycling collection routes. The dataset we used shows daily trends but could easily be outfitted with hourly, monthly, or yearly datasets to discover other data trends.

Finally, we tuned the Dynamic Data visualization to use the many tools that CARTO provides. The resultant visualization can give taxi companies in New York City a foundation of knowledge about their pickup and dropoff data to negotiate better policies for both the company and the taxi drivers.

---

3 For more on future recommendations, see section 5.1 Future Work
5 Conclusion

When we started developing CityView, our goal was to create a website that would give users the power to analyze datasets of varying spatial-temporal categorization while providing them with real-world scenarios in which this system could be used for urban development. We were able to accomplish this goal through technical and practical means. Technically, we iterated CityView from a basic visualization to a more in-depth, well documented analysis tool. Practically, we provided useful details about the data used in the visualizations and included informative data stories to reveal insights into the data. The result is an interactive website which provides users with a multitude of analysis techniques.

5.1 Future Work

There are several recommendations for future work that would increase the usability and functionality of CityView. They are, in no particular order:

Add a feature allowing users to draw bounding boxes.

This feature would allow users to query specific regions and segment their datasets to look at as more specific picture. One implementation of this would allow users to limit the data points plotted on the map based on city limits or neighborhood borders. Using bounding boxes, the user would be able to make queries within the bounds such as data counts, data distribution, data label frequency, etc.

Apply labels to data.

This feature would allow users to zoom in on one particular data point and access more information about it. For example, on the street light data if a user clicks on the particular data point they could gain information about when the streetlight was installed or when it was last maintained.

Allow users to upload their own data.

This feature would allow users to upload their own data and have the option to store this information in a database. Doing so, the amount of information presented by the system would grow indefinitely. By utilizing an uploading service and a server-side database it may be possible to allow anonymous uploads shared between users.
Allow for more unique data types.

At the time of writing, CityView only utilizes point data in CSV format. Each data point is labeled according to a latitude and longitude coordinate. Leaflet, Mapbox, and Carto all benefit from being able to process more than just points such as lines and polygons. By increasing the type of data even more interesting visualizations can be applied.

In a similar vein, dedicated file format converters would be a natural extension to the CityView infrastructure. While there are other file converters out there, it would benefit the CityView user to have one built right into the website which they know will properly convert their files to the desired format. This would also be a measure in making the system future ready by handling alternative file formats such as GEOJSON.

Add live data feed compatibility.

This feature would allow users to visualize dynamically changing data sets. Cities are ever changing and so is the data surrounding them; in order to make the best decisions users need a real-time visualization of a city. By creating an interface to constantly gather and recompile new city data, users are not only able to make important conclusions based on the visualizations but are able to make those decisions in a timely manner.

Create a dedicated backend.

Creating a backend would put a majority of the computational weight on a server and relieve the client-side operations. By creating maps and data layers on a server instead of having the client-side machine iterate through data to create these layers the speed and performance of the entire system would increase significantly.

One example of a dedicated backend included storing the data information in a relational database. Storing data in a backend database allows for conveniences such as caching common map and data layers or writing algorithms in other, non-web-based programming languages. Doing so would not only increase the quality of the user’s experience but also open the door to more computationally intensive operations that could not reliably be completed client-side.
Add correlation analysis.

This feature would allow users to better understand the correlation between applied datasets through an interface to analyze empirical correlation data. By performing statistical calculations on individual data sets and groups of data sets, the user would uncover a new dimension to their data and have a deeper insight into the implications the data holds.
References


