Visualizing Bike Share data on the San Francisco Bay Area

Information Visualization, 27/11/2023
DETI - UA

Diogo Gomes, 93025, MEF
Martinho Tavares, 98262, MEI

Mini-Project Presentation
https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share

The dataset

San Francisco Bay Area contains several bike stations throughout various cities.

These bike stations contain rental bikes that are shared between people.

The dataset contains:

- Information regarding the station-to-station trips performed over time
  - Duration
  - Bike used
  - Which customer type performed it
- The number of bikes available in each station over time
- Station information (city, coordinates, maximum bike capacity)
- Weather information, which is not used

The users

This application is intended for those who manage San Francisco Bay Community Bikes (administrative bodies).

With it, we hope to arm the user with a tool that will help them know where there is room for improvement, and why.

The user is expected to effortlessly know the main traffic trends, have a clear and broad notion of supply and demand, and know which bikes were used the most.

Lastly, they should be able to understand the differences in bike usage between the two client types: customers and subscribers.

Proposed visualizations

Map

- Where the user can explore the cities/stations and select them, and to also have a better understanding of the bike traffic

Bar chart

- Where it is displayed the traffic between cities/stations
- Also used to display information about the type of user

Line chart

- Where the availability of bikes in the city/station docks can be tracked

Histogram

- Where the amount of time each bike was used can be tracked

Data preprocessing

The dataset is very big, with one particular file occupying close to 2GB.

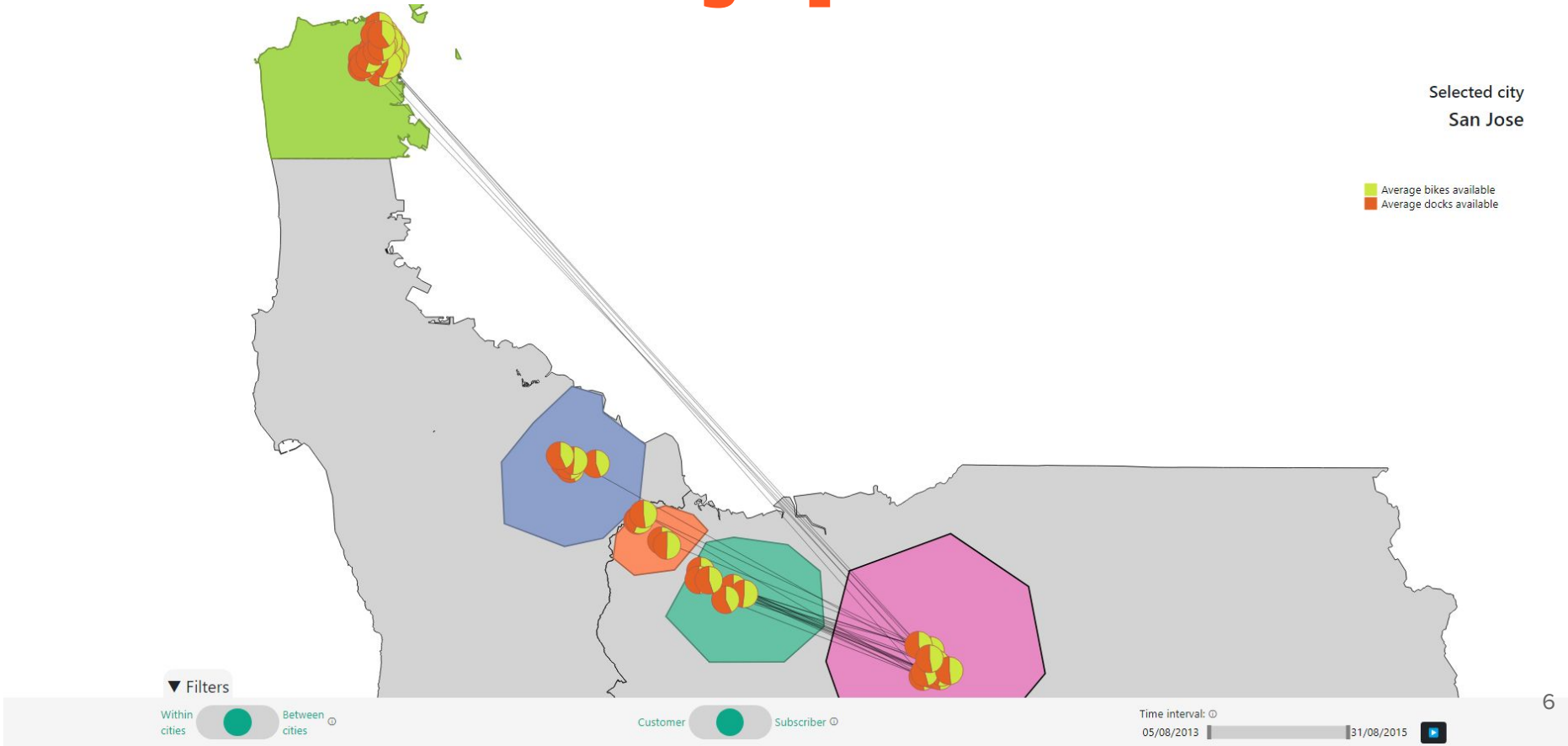
We had to greatly simplify the data in order for it to be tractable in a web application on the client-side.

We created:

- `metadata.json`: miscellaneous data required by all visualizations (stations)
- `status_small.csv`: the original `status` was in seconds, we aggregated to days
- `trip_small.csv:` trip data was aggregated by trips, and some attributes were removed
- `bike_usage.csv`: extracted only bike duration from trip data

Additionally, for the geographical visualization, for all cities other than San Francisco we drew the borders by hand due to lack of such data.

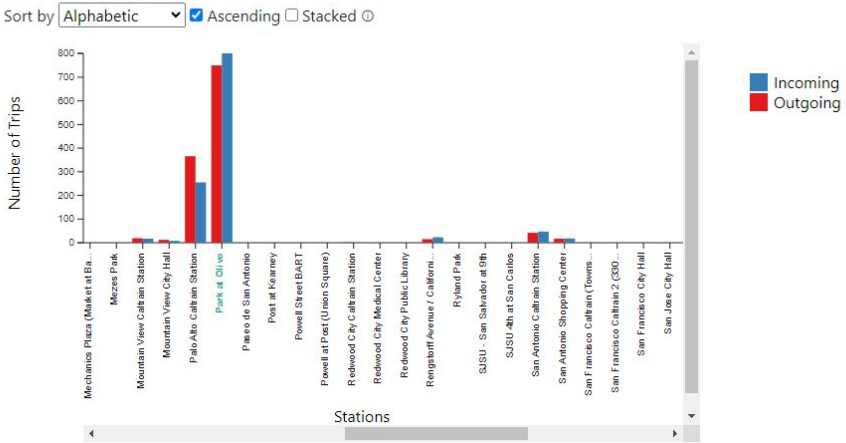
Visualization 1 - Geographic view



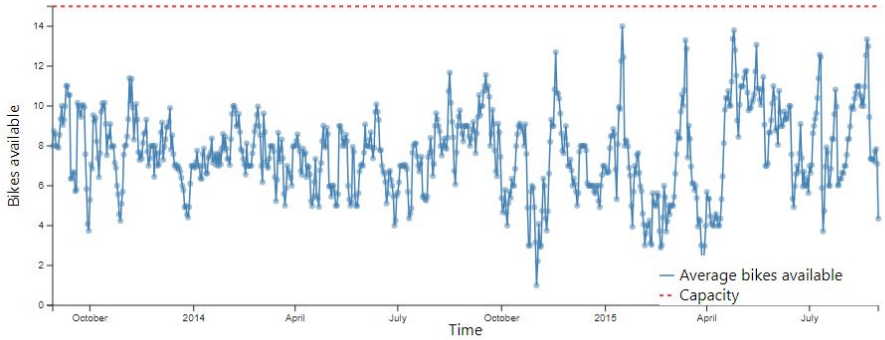
Visualization 2 - city/station details

Details of Park at Olive

Bike traffic from/to other stations



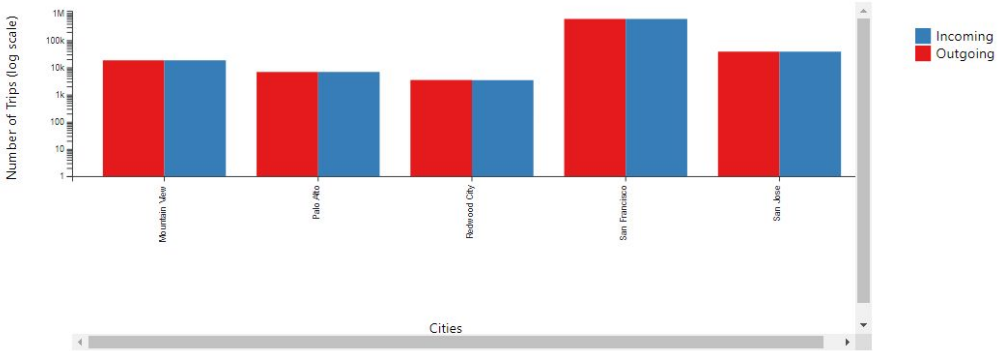
Bike availability



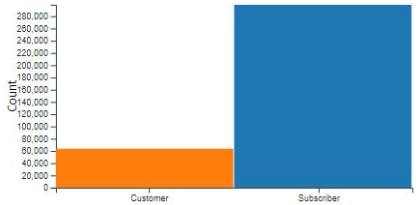
Visualization 3 - global details

Global bike traffic for each City

Sort by Alphabetic Ascending Stacked

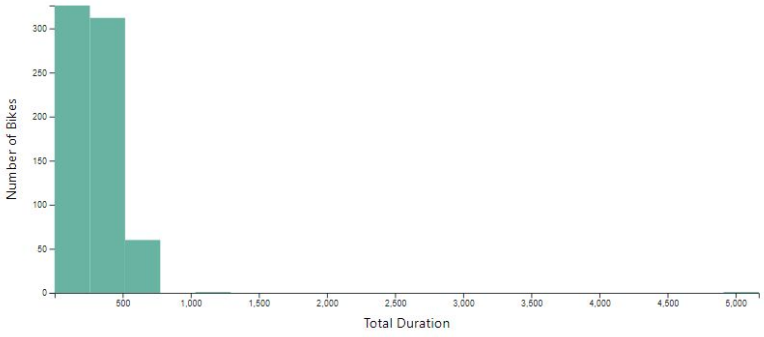


Total trips made by each client type



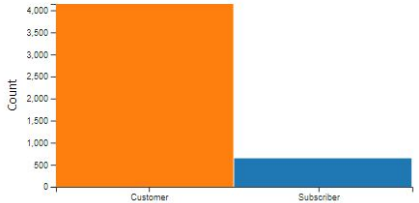
Bike usage

Number of bars to display: 20



Bike ID	Total duration (hours)
...	...

Average length of trips made by each client type



Prototype changes and additions

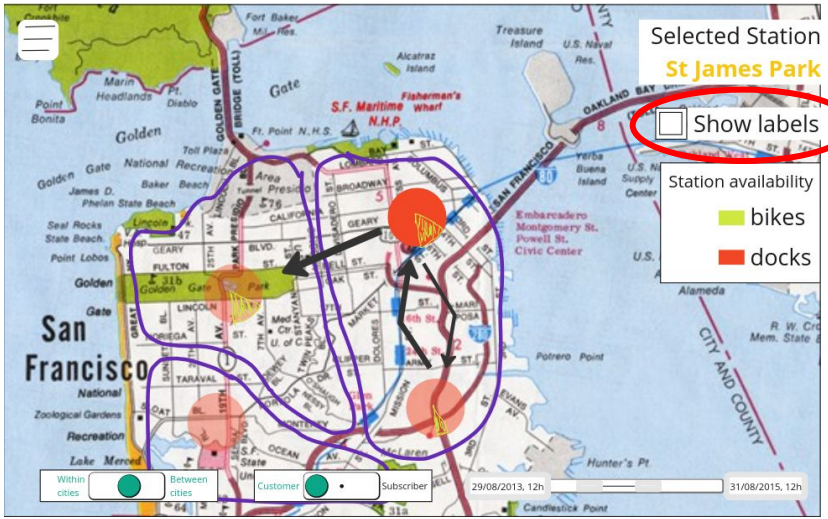
The project suffered some changes since the first prototype.

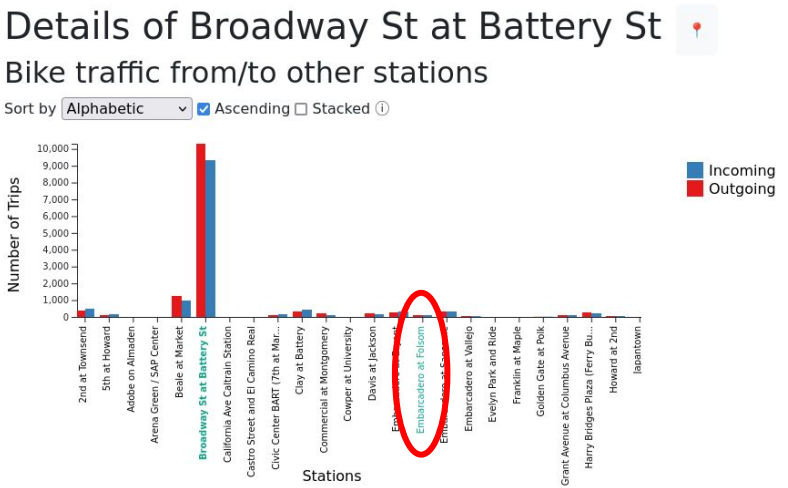
Some visualizations and interaction features were changed according to the feedback received from the first prototype and heuristic evaluations done since then.

Prototype changes and additions

“Show labels” was supposed to show all city and station labels, to make it easier to select a city/station by name.

Due to implementation issues, opted to allow for selection in the X axis of the trips bar chart (second visualization), which allows alphabetical sorting.

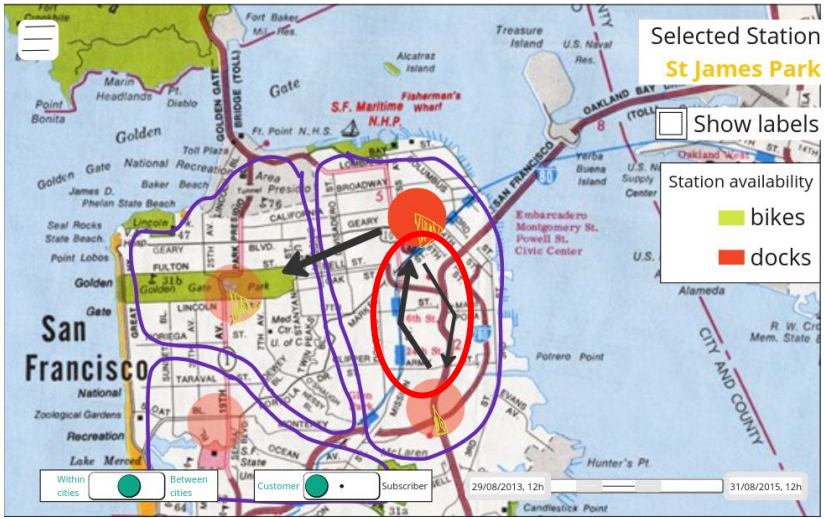


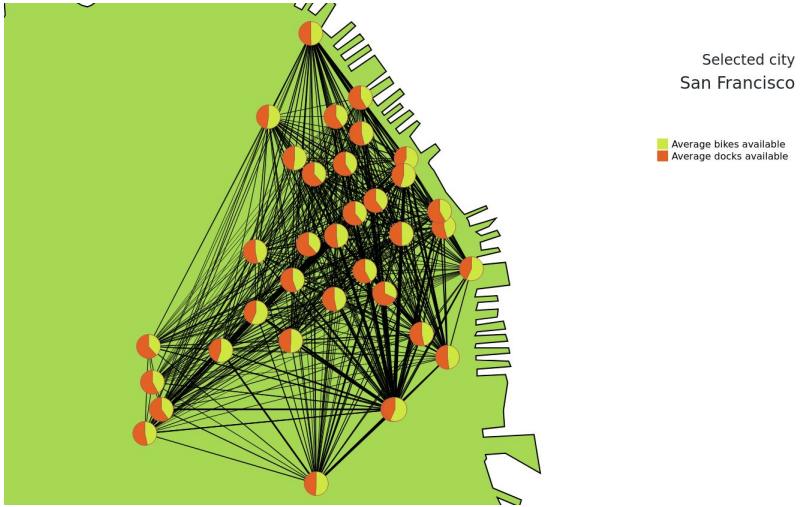


Prototype changes and additions

The trip vectors were meant to encode not only the number of trips but also the direction of those trips.

Since the map could get very crowded with vectors (especially the city of San Francisco), we decided to just encode trips in both directions, and let direction information be extracted from the trip bar charts.



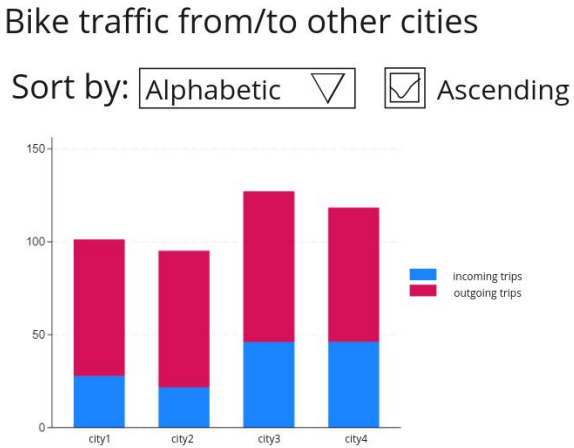


Prototype changes and additions

The trips bar chart was supposed to be stacked.

Since it becomes hard to properly compare incoming and outgoing trips, we decided to allow the option to view the same chart unstacked.

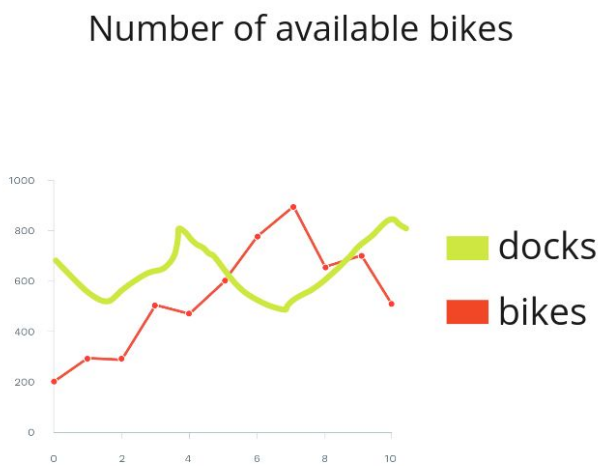
It also doesn't make sense when we are comparing cities, which requires the Y axis to be logarithmic.

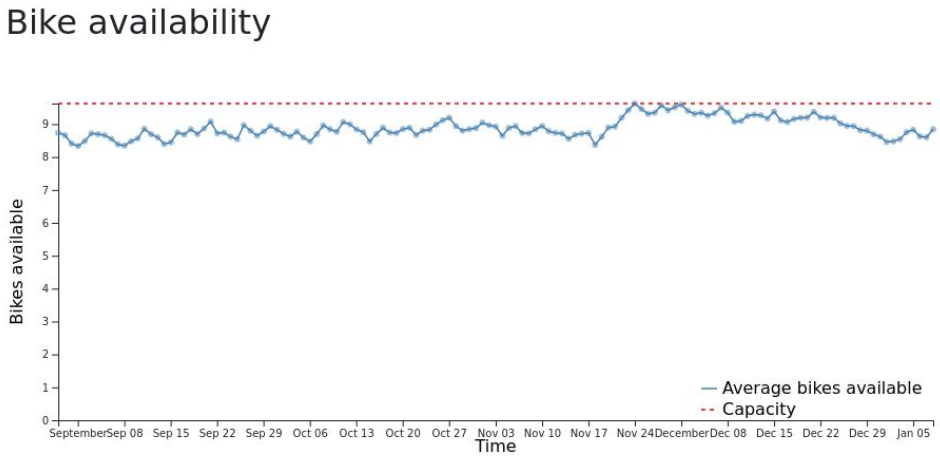


Prototype changes and additions

The line chart indicating the bike availability over time was supposed to include both the available docks and bikes.

Since each line is the inverse of the other, we decided to just plot one of them and include an horizontal line indicating the total number of docks (available and unavailable).

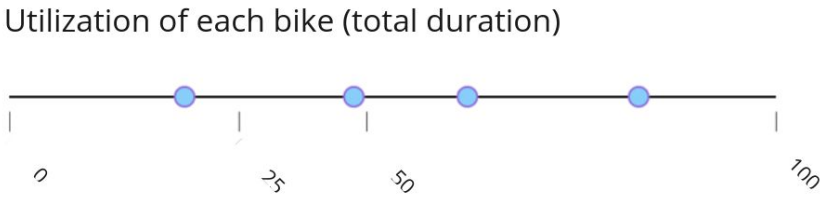


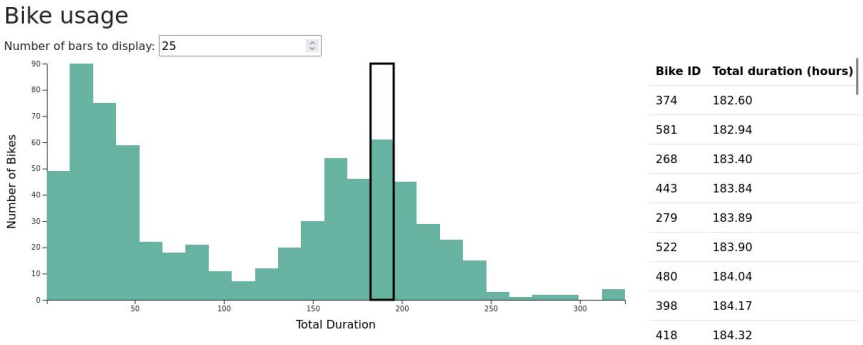


Prototype changes and additions

In order to show bike usage, we meant to use a 1D plot.

We were suggested to use an interactable histogram along with a table instead, which allows better understanding of the distribution of the data as well as knowing which bikes are present

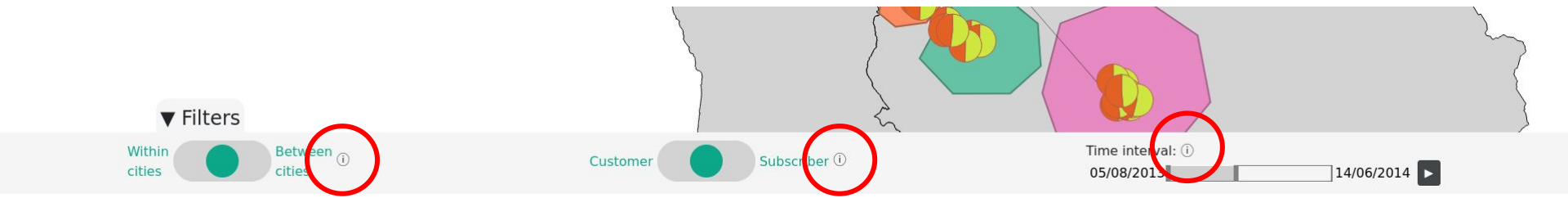




Bike ID	Total duration (hours)
374	182.60
581	182.94
268	183.40
443	183.84
279	183.89
522	183.90
480	184.04
398	184.17
418	184.32

Prototype changes and additions

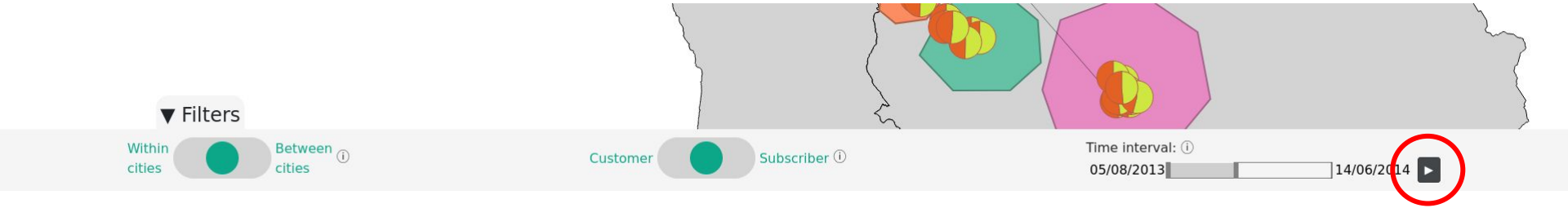
We added a tooltip to each filter to provide a brief description of what they do.



Prototype changes and additions

The time filter allows understanding how data changes over time.

However, it's slightly cumbersome to constantly change the time interval forward to understand these trends. Therefore, we added a play button that advances the filter by 1 month every 1.5 seconds

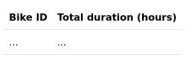


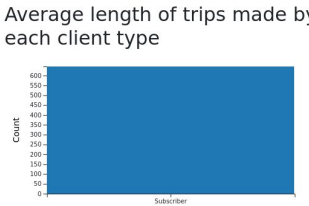
Demo

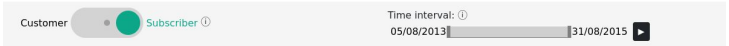
Video: https://www.youtube.com/watch?v=oT-qJGSdbEE

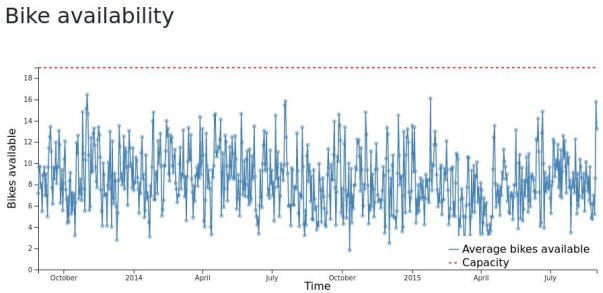
Limitations and future work

- Some filters change charts that don’t make sense to be affected by filters
  - For instance, the client type bar charts
  - This is due to the modularity and design of the code which made implementing these specific cases awkward
- The data filtering process could be optimized
  - There is too much data to handle, making the site slightly unresponsive
- The bike availability line chart could use some smoothing
  - There is too much data when we consider large time intervals, making the data hard to read









Conclusion

We successfully created a visualization application exploring most of the chosen dataset.

D3 was very helpful in efficiently incorporating personalized interaction features (such as X-axis station/city selection and the selectable bike usage histogram).

The project idea suffered changes since the prototype, most of which were integrated.

Some problems were still present, which are left for future work. Most notably, the fact that all visualizations share the same filters and the heavy processing requirements.