Citi Bike NYC:

Analysis of Bike Movements from Station to Station

Jon Leavitt

General Assembly

Data Science 9

Spring-Summer 2014

**Citi Bike Background**

Citi Bike operates 330 stations with docks for several thousand bikes in lower Manhattan and Brooklyn. Annual subscribers and customers purchasing daily or weekly passes can take bikes on unlimited short trips from station to station. Commuters in particular use Citi Bike to get to a train station, or to get back. The average distance is just over 1 mile, and the average duration is about 15 minutes.

**Citi Bike’s Inventory Problem**

During the warm months, Citi Bike customers and subscribers take tens of thousands of rides per day. From August 2013 (just 3 month after beginning operations) through October 2013, bikers took over 1 million rides per month. Since riders often want to take bikes to and from the same stations at the same times as other riders, there are often no bikes available at desirable stations, and no docks available at desirable destinations. Since there are no restrictions on which stations riders use, the only limiting factors are bike and dock availability. This can be frustrating for the people that use Citi Bike.

Citi Bike gets no money from New York City. They are dependent on riders paying to participate. There have been fewer of the very profitable short-term customers than Citi Bike projected, but there have been more annual subscribers providing revenue than expected. Citi Bike is motivated to have bikes and docks available because they need annual subscribers to resubscribe, and they want word-of-mouth to attract new subscribers as well. Recently, Citi Bike has stated that they are ‘experimenting with using bike trailers for “rebalancing” – moving bikes from stations with no empty slots to docks in other areas.’ This could be a crucial step in addressing riders’ complaints and increasing customer satisfaction.

**Citi Bike Data**

Citi Bike has made data available (at <http://www.citibikenyc.com/system-data>) that covers all rides taken by subscribers and customers at all 330 stations from July 2013 through February 2014. The data includes the exact time and beginning and end stations of all trips, as well as a few other variables.

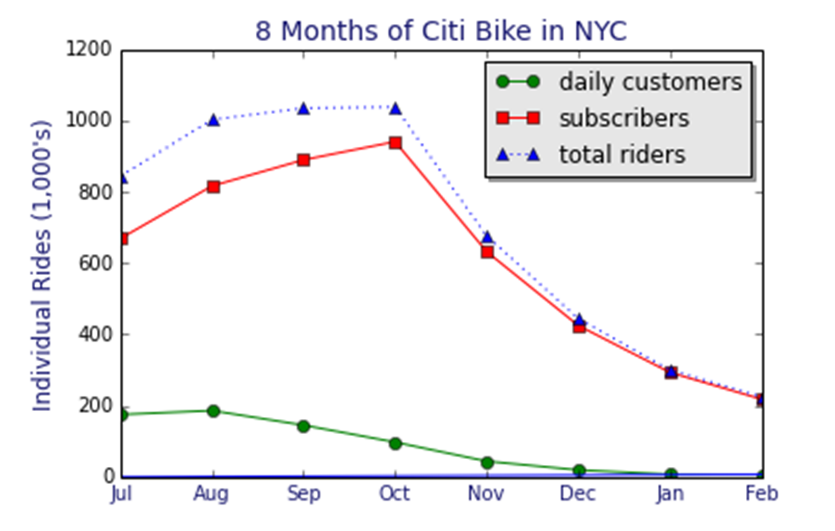
**Goals**

The main goal of this project is to use the Citi Bike data covering several million trips over 8 months and use it to describe and visualize the movements of bicycles from station to station in a meaningful and intuitive manner. Additional purposes are to illustrate and differentiate between bike movements at specific stations, and to create functionality that can generate insight to aid Citi Bike in rebalancing stations.

**Initial Analyses**

Initial visualization explored trip durations, which were almost universally quite brief (although short-term customers are much less adept than annual subscribers at returning bicycles within the allotted time frame), and also trip distances, which are generally quite short. (Trip distances were generated by geographic distance between start and end stations, so in many cases they underestimate the actual distances biked.) Differences in bike use in terms of numbers, duration and distance were determined and visualized by gender and by customers vs. subscribers. These visualizations can be produced or recreated if desired.

Below is a summary of rides taken per month by customers and subscribers for July through February. The percentage of riders who were short-term customers by month coincidentally almost perfectly matches the area under the green curve below, which reflects how Citi Bike underestimated the potential impact a harsh winter would have on short-term rentals.



**Mapping Bike Imbalances**

To map bike imbalances, it was necessary to sum across rides by station. Summary data frames including numbers of departures and arrivals by station were generated from the data for each month. The large amounts of data tended to crash the iPython Notebook, so the summary data was written to csv files so it could be retrieved without loading the massive main data files. The summary data files were then combined into a master file with data spanning eight months.

One straightforward task was to use the mapping software provided by CartoDB to map the summary of the activity over eight months at the 330 stations. Station data including name, latitude and longitude data were copied from the data files to the summary files. The goal was to create an animated torque map that would illustrate the imbalances in departures and arrivals in all the stations over 8 months. However, the torque map read through the file line by line, rather than in the desired order. As such, a new data file was created that organized the data in chronological order. Still, the torque map read unpredictable partial sets of the 330 stations at a time, creating visual effects across the 330 mapped locations that were very cool, but not really possible to comprehend. As a last-ditch effort, 8 copies of each of the 8 month’s data were made within the file to match the minimum (non-1) torque map step option on CartoDB (64 steps). The idea was that the software would divide the data into 64 equal sets of rows. If this were true, each of the 64 steps would depict complete data for one month for all 330 stations. The first 8 steps would depict July in full 8 times, followed by 8 complete Augusts, and so on. In reality, the torque map operated in unpredictable fashion, and the animated visualization continued to be cool but not comprehensible. (CartoDB announced new torque map documentation on June 22nd, so it may now be possible to trick the map into producing the desired visualization.)

CartoDB was nonetheless quite useful in being an intuitive way to produce inanimate color-coded maps of station activity. By loading data for each month separately, station imbalances could be clearly visualized. The most satisfying image was generated using a chloropleth map, which divides the data into a number of buckets (the max of 7, in this case), and maps it according to a selected color bar. The small image below depicts summary data for eight months combined. Stations are shaded such that darker red indicates stations where greater numbers of bikes were docked (stopping) rather than taken, while darker green indicates greater numbers of bikes were taken (going) rather than docked. At the extremes, a number of stations exhibited bike imbalances in one direction or the other of thousands of bicycles over the 8-month period.

****

**Depicting and Differentiating Stations**

**Determining Which Data to Summarize**

Many different approaches were taken to visualizing bike activity at different stations. Data was generally summarized per week, since patterns of bike activity tend to repeat on a weekly basis. Variations within this theme included summarizing bike use and bike imbalance by weekday versus weekend day, by day of week, or per hour. Eventually, functions were written to automate the process of generating summary data by station. Functions take a station number (or list of station numbers) and data source, and return matrices populated with summary data for movements at that station. Separate matrices are returned for departing, docking, and imbalance (or difference) between the two. One function summarizes data by hour, and another by periods of three hours. This way, it is easy to access exactly how many bikes departed from and arrived at station 519 between 3-4 pm on an average Monday, or between 3-6 pm on an average weekday, across the time period covered by the data frame.

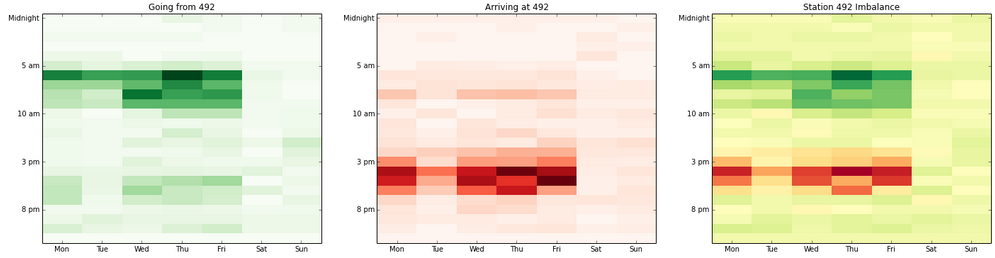
With more information from Citi Bike, line graphs could be produced to track number of bikes out of total available docks at a station over time. However, it is impossible to know how many docks and how many bicycles are not in working order. In some cases, number of docks at a station may have been altered. Even Citi Bike’s information available to riders online, listing number of docks and available bikes per station, is inaccurate. For these reasons, it is impossible to know how many docks and bicycles are actually available. Times when all bikes are away or all docks are full, which would be very desirable to know, also becomes indeterminate.

That said, most stations show surprisingly consistent usage over the 8-month period, indicating few wholesale changes.

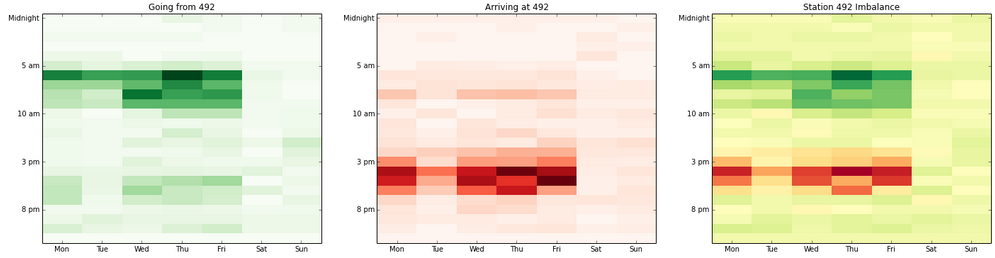
**Visualizing the Data**

Many different approaches were taken to visualizing bike activity at different stations. Data from the matrices containing average number of rides per hour/day/week were mapped in color using matplotlib. Initial matrices were difficult to interpret, as shading was fuzzy and color ranges were non-intuitive. Setting “interpolation” to “nearest” made the borders clear, uni-directional color bars made the shading easier to interpret, and adding meaningful x-ticks and y-ticks made the graphs nearly self-explanatory.

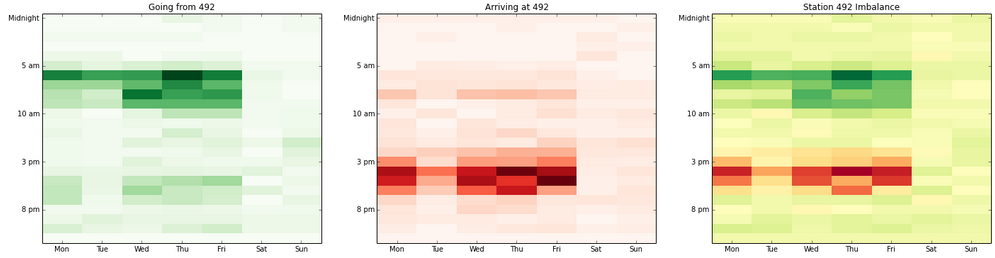
Establishing conventions for graphing station data was also very useful. Taking a bicycle is “going,” and docking a bicycle is “stopping,” and going and stopping are activities with which most people have clear color associations. Therefore, numbers of riders leaving stations were depicted in green, ranging from very light green depicting few riders to very dark green depicting many, as in this graph depicting average riders per hour in an average week at station 492:



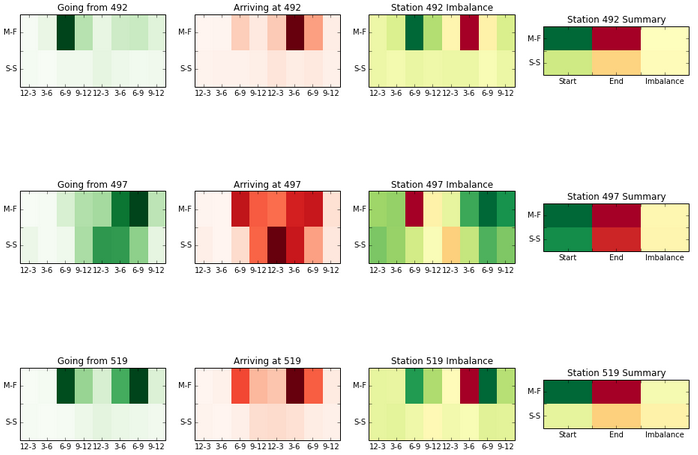
Similarly, riders docking at a station were depicted in red, ranging from light red for few dockers to dark red for many:



The imbalance at the station was then depicted using RdYlGn, which was the best available color bar for depicting sums of departures and arrivals that could be positive or negative:



While these visualizations were relatively effective in terms of visualizing movements at a particular station, they were also somewhat deceptive in the information offered. For instance, consider summary of three stations (492 at the top, 497 in the middle, 519 at the bottom) that includes three graphs depicting average activity over three-hour periods on weekdays versus weekends (going, stopping, and bike imbalance), plus a daily imbalance summary at the end:



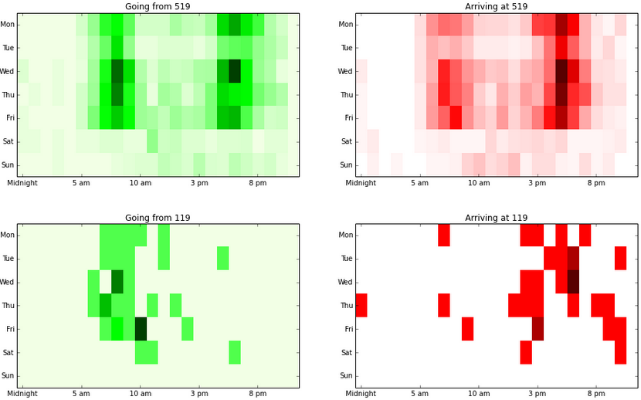
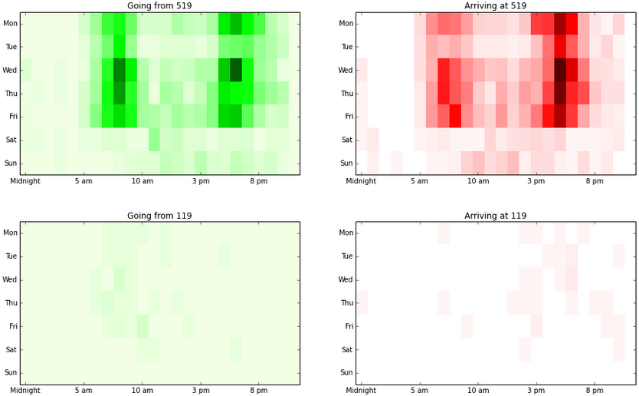
Some of the things you cannot determine from the above graph include:

* Which station has the most bicycles going or coming?
* Which station has the greatest imbalance?
* Are any station’s darkest reds greater than, less than, or equal to its darkest greens?
* Which beige-yellow imbalances reflect positive versus negative values?

The main issue is that color bars automatically scale to the data. A graph of (0, 0, 1) and a graph of (-35, -35, 80,000,000) are the exact same, because they each have the structure (min, min, max). Rescaling the matrix does not help, as the color scale will automatically adapt and remain the same. One possible solution is to pass in extreme values equivalent to the max and min of the desired represented range. However, this requires mapping extraneous data within each graph. Another solution is to program a color bar to rescale itself according to the portion of the color bar the data represent. If 0 and 100 are extreme values then (0, 0, 1) would appear as (min, min, tiny).

The secondary issue is that RdYlGn has yellow in the middle, and it is difficult to differentiate between red-yellow and green-yellow. Part of the solution to both issues is to create new color bars. Color bars were created for red and for green that ranged from white for minimum values to very dark red or green for maximum values. A third color bar was created ranging from dark red to white to dark green in order to represent negative and positive imbalances. A program was written to handle various relationships between the data and the desired extreme values by altering values or changing the color bar structure for uni-colored graphs. Another program was written to automatically update the red-green-white color bar as well; this latter program is functional but still under construction.

Here is an example of the distinction, using the very heavily trafficked station 519 and the little-used station 119. The graphs on the left depict going from and stopping at 519 on top and 119 on the bottom. 519 actually does look somewhat busier because it contains so many intermediate values rather than zeroes, perhaps making this a less-than-ideal example. Nonetheless, the equivalent graph on the right shows what these stations look like when consistent scaling is used across all graphs. It is clear from a glance that station 119 has little overall impact on the distribution of bicycles within the Citi Bike System.

**Future Directions**

There are innumerable potential future directions to go in with this project.

One crucial step will be to increase automation. This will involve determining which kinds of reports will be most useful, and also combining programs so that selecting simple options can produce distinct reports. More work specifically needs to be done in order for the bi-directional color bar to be responsive to a broader range of potential scenarios. The uni-directional color bars have greater flexibility, but could also be expanded to cover all possibilities.

Another focal area for potential automation involves the creation of a dictionary with keys referencing the bike station ID numbers. Many files created within and written from these programs have identical names, except for the ID number used to associate the type of file with the relevant station. A dictionary with station ID’s as keys could be used to programmatically store and access information about each station, processes that are currently done manually.

A more formal analysis of distinctions between stations was also an initial goal of this project. A sort of cluster analysis was done in the creation of buckets for the cartoDB chloropleth map, but it would also be worthwhile to cluster stations by multiple dimensions, such as volume of traffic, peak hours, direction of morning versus afternoon traffic, and of course location.

One specific analysis that might be worthwhile is to assess distinctions between comings and goings of short-term customers and annual subscribers. Since there is more immediate profit to be made from short-term, and fewer of them, it should be possible to aid Citi Bike by determining how to serve their needs without disrupting the system. If short-term customers tend to frequent different stations than subscribers, priority can be given to maintaining bike and dock availability as required.

In the broader picture, greater insight into the needs of Citi Bike would permit the creation of more relevant reports. Citi Bike released a great deal of data to the public, but there is also still much data within the system that is not available.