Steve Jarvis, Cristina Estupiñán
CS591 Data Mechanics
Spring 2016

Optimizing Green Line Stops

Introduction

The Green Line seems to stop too frequently. Presuming the best location for a stop is where one doesn't already exist and people would most use one, we think analyzing the popularity of stops against the distance to alternatives can help decide which existing stops are most valuable and which can be omitted. To make such a determination, we use a variant of the *k*-means clustering algorithm that takes into consideration the value of the current stops to the commuting community.

Methods

Our goal is to determine the optimal locations for Green Line stops, and to calculate the location of these optimal points we use to the *k*-means clustering algorithm. This algorithm divides the dataset into *k* clusters that optimize some measurement. In our case, the measurement is the aggregate distance between existing stops and the *k* means, and the algorithm's goal is to minimize this net distance[1]. Once the clustering algorithm is finished running, we have a set of optimal stops for each branch.

We also create a score called *people seconds* to gauge the utility of each current stop. This is a measure that uses both the popularity of the stop and the walking time to the nearest alternative stop, with the goal of approximating the time cost to the collective commuting community by that particular stop. It does this by considering the time saved for all the people using that stop compared to the cost to the people that are already on the T for needing to make it:

$$people\ seconds\ =\ (P\ -p) * s0\ -\ p * s1$$

Where *P* represents the popularity (boarding count) of all stops, *p* the popularity of the stop in consideration, *s0* the estimated time it takes to make the stop, and *s1* the time it takes to walk to the next alternative stop.

The score ultimately provides a weight, or importance, for each stop. A low score is good; this means that the stop is valued and saves the greatest amount of time for the collective commuting group. This score will let us perform a weighted *k*-means to find the optimal *k* stops, and favor those stops we consider highly valued. We use the *people seconds* measurement to give a weight to each stop and influence the position of each mean.

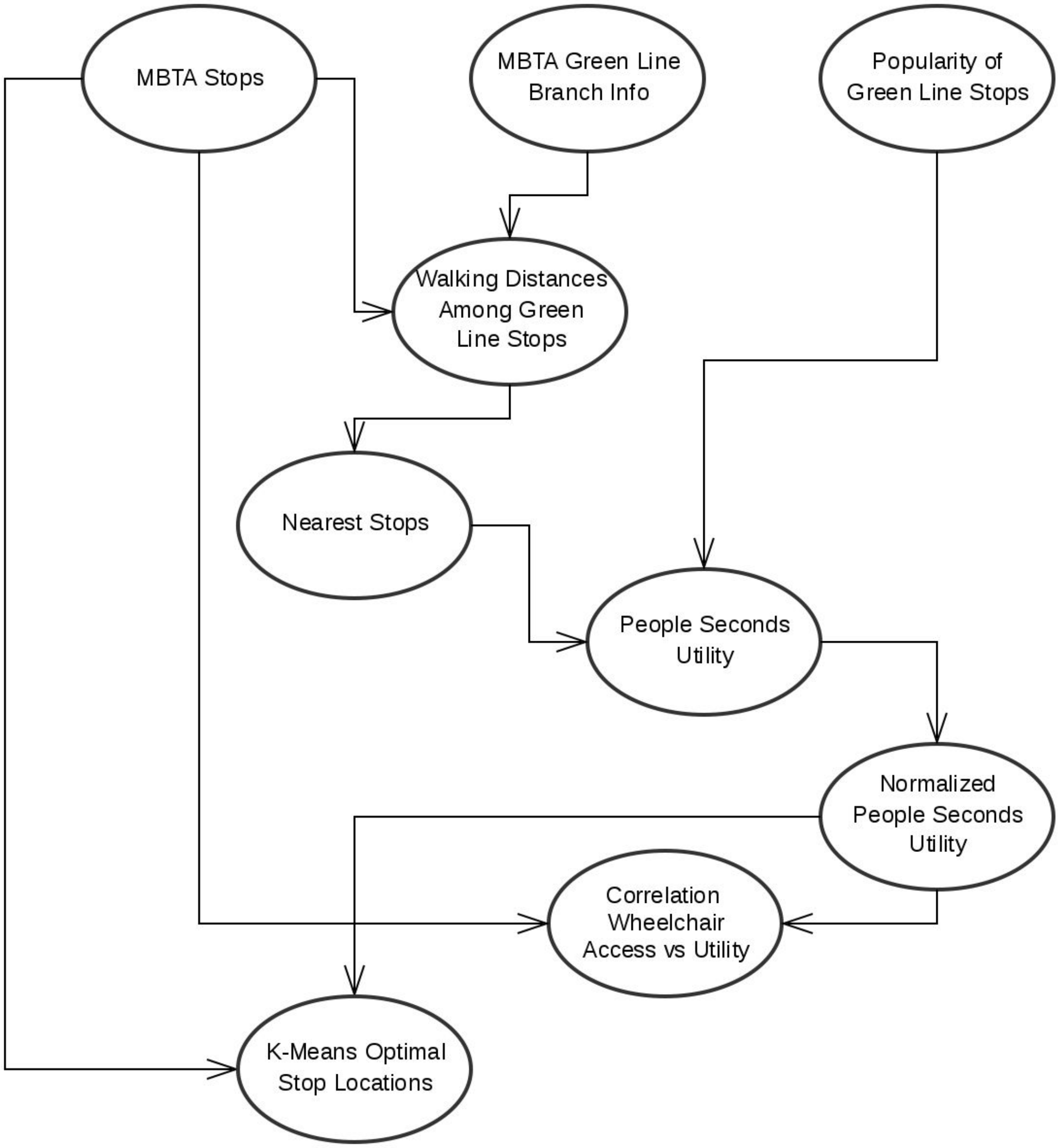
[1] While *k*-means does minimize the aggregate distance from all existing points to *k* optimal means, the inclusion of a weight on the averages in this case does mean the final optimal stops do not directly minimize distances.

Once the optimal locations are calculated, we are able to consider *k* to be *x* fewer than the number of existing stops to determine which *x* stops are best to remove from existing routes: the *k* stops nearest the cluster means (and only one stop per mean) would be the ones to persist.

As a metric to help establish our utility measurement as a desirable and positive one, we also calculate the correlation of the *people seconds* score against the availability of handicap access at the stops, to see whether our value system tends to favor stops with suitable access.

To do all this, we need to know the walking distances to nearest T-stop alternatives, as well as the popularity and the approximate time it takes to make each stop. The MBTA publishes data on the boarding counts at each stop and the GPS locations, as well as branch association, of all stations. We assume deboarding counts are proportional to the boarding at each location.

Data Sets and Algorithms Involved


MBTA Stops
MBTA Green Line Branch Info
Popularity of Green Line Stops
Walking Distances Among Green Line Stops
Nearest Stops
People Seconds Utility
Normalized People Seconds Utility
Correlation Wheelchair Access vs Utility
K-Means Optimal Stop Locations

The above image depicts an overview of the sequence of transformations we applied to the data to get to our results.

1. GPS Location of T Stops

This dataset uses information provided by the MBTA[2], which gives the exact GPS locations of each T-stop on the Green Line. This is used to help us create the derived datasets. The dataset is built from a comma-separated value file published by the MBTA[3].

2. Branch Associations of T Stops

To determine alternative stops, we consider the branch of the Green Line as well as physical proximity (only those stops servicing the same branch are considered as alternatives). This dataset includes the stop ID, stop name, branch, next inbound stop, and next outbound stop for each branch. We use this to determine how each branch is ordered, and again to create the derived datasets. This information was manually assembled based on stop IDs used by the MBTA[4] and a map of T lines[5].

The map used to handcraft this data set exists as a provenance entity, with a source of where we found it online, but any map of the MBTA stops likely provides the same information.

3. Popularity of Each Green Line Stop

This dataset provides to average boarding population per day at each stop along the green line. We will assume that the number of boardings is proportional to the number of passengers that disembark. This information was provided by the MBTA, using the most recent boarding counts (2013)[6].

The dataset is handcrafted based on a PDF published by the MBTA and the stop IDs included in the GPS dataset[7].

4. Walking Distances to Other Stops on a Branch (Derived)

This derived dataset uses Google's API to get the walking distances to the next nearest stop on the corresponding branch. It considers the walking distances from the passenger's current T-stop position to all other T-stop positions within that branch.

[2] The stop data was contained in a zip file, originally retrieved from here: http://www.mbta.com/rider_tools/developers/default.asp?id=21895

[3] Dataset is hosted at http://cs-people.bu.edu/sajarvis/datamech/mbta_gtfs/stops.txt.

[4] The same stop IDs are used throughout, based on those published in dataset 1.

[5] The handcrafted branch association data is hosted at http://cs-people.bu.edu/sajarvis/datamech/green_line_branch_info.json.

[6] The MBTA publishes this information in "Ridership and Service Statistics Fourteenth Edition 2014". The version we used can be found at http://www.mbta.com/uploadedfiles/documents/2014%20BLUEBOOK%2014th%20Edition.pdf.

[7] Dataset is hosted at http://cs-people.bu.edu/sajarvis/datamech/green_line_boarding.json

The provenance data does not list each URL queried, since they change for every combination of source and destination and are dependent on the combined data being used. We instead insert placeholders for the coordinates, e.g. "<source_lat>" for the source latitude.

The current implementation takes a long time to generate the dataset because Google's API throttles the number of requests we can make on the free tier. We are limited to roughly one request per second, which amounts to around 45 minutes running time. If the project instead hosts our own routing service, the script could be trivially modified and complete in only a few seconds.

5. Time to Nearest Neighbor Stop on Same Branch (Derived)

This derived dataset holds information regarding the nearest neighbor stop on the same branch for each existing stop, including the distance and time it takes to walk there. If the two stops are very close together, that indicates the current stop is less important and has a greater potential for removal.

6. Utility Measurement Based on Passenger Saved Time (Derived)

In this derived dataset, we create the *people seconds* metric to gauge the utility of each stop. As described above, this is a score that uses both popularity and walking time to the next nearest stop, and provides a score for each stop.

7. Normalized Utility Measurement (Derived)

As originally calculated, the *people second* measure is not directly suitable as a weight to *k*-means. We transform the score to make the more favorable scores larger, positive numbers, and scale all ratings to range from 1 to 1000. This dataset is a measure directly usable as a weight on averages by the *k*-means optimization.

8. Weighted *k*-Means Optimization for Each Branch of Green Line (Derived)

This derived dataset stores the coordinates of the optimal stops, as computed by *k*-means, for varying values of *k* on each branch of the Green Line. It holds the *k* optimal stops for each *k* from 1 to the current number of existing stops on each branch.

9. Correlation and P-Value of Utility Ratings Versus Handicap Access of Stops (Derived)

In this derived dataset, we store the correlation coefficient and p-value of the correlation between our utility score (*people seconds*) and the availability of wheelchair access at each stop[8,9,10].

[8] The wheelchair accessibility of each stop is found in a zip file, originally retrieved from here: http://www.mbta.com/rider_tools/developers/default.asp?id=21895

[9] Google Develops provides the codes for wheelchair boarding at https://developers.google.com/transit/gtfs/reference#agencytxt

Results

We were able to obtain the measurements we initially wanted to find. In the end, we have the optimal stop locations for any desired number of stops (fewer than the number of existing stops), and with a preference towards the locations the commuting community currently finds most useful.

In 2014, the MBTA proposed removing four stops from the B line: BU West, St Paul St., Pleasant St., and Babcock St., replacing them with one new stop just west of BU West and a second between Babcock St. and Pleasant St. Our methods agreed; if we were to use it to improve the Green Line B branch by just one stop, it would remove the Pleasant St. and Babcock St. stops, replacing them with a new stop in between.

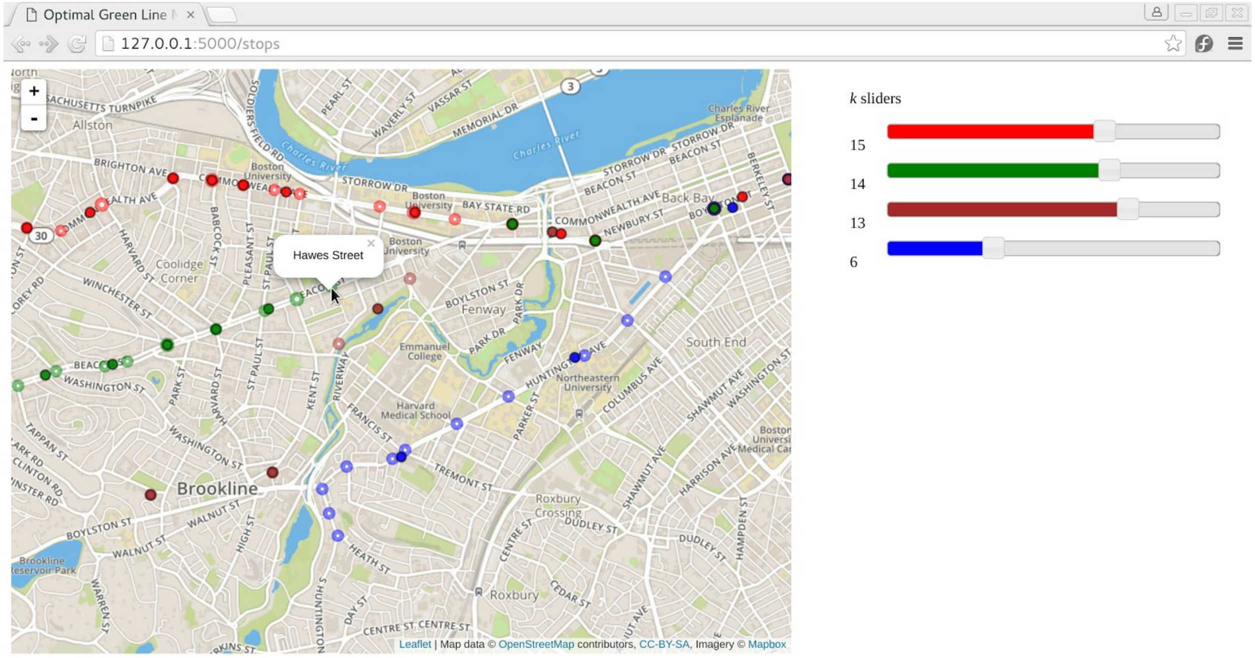
We also found a notable relationship between our *people seconds* score and existing handicap access. We calculated the correlation coefficient between the two dimensions as 0.395 and the p-value as 0.0009996. The low p-value indicates a significant positive correlation between a high utility and wheelchair accessibility.

Visualizations

The visualizations are viewable in a web browser. These visualizations need access to the database from the browser, which can't be directly obtained. So we use a small Flask application to act as a middleman to facilitate this access.

[10] This derived dataset requires the scipy for Python3.

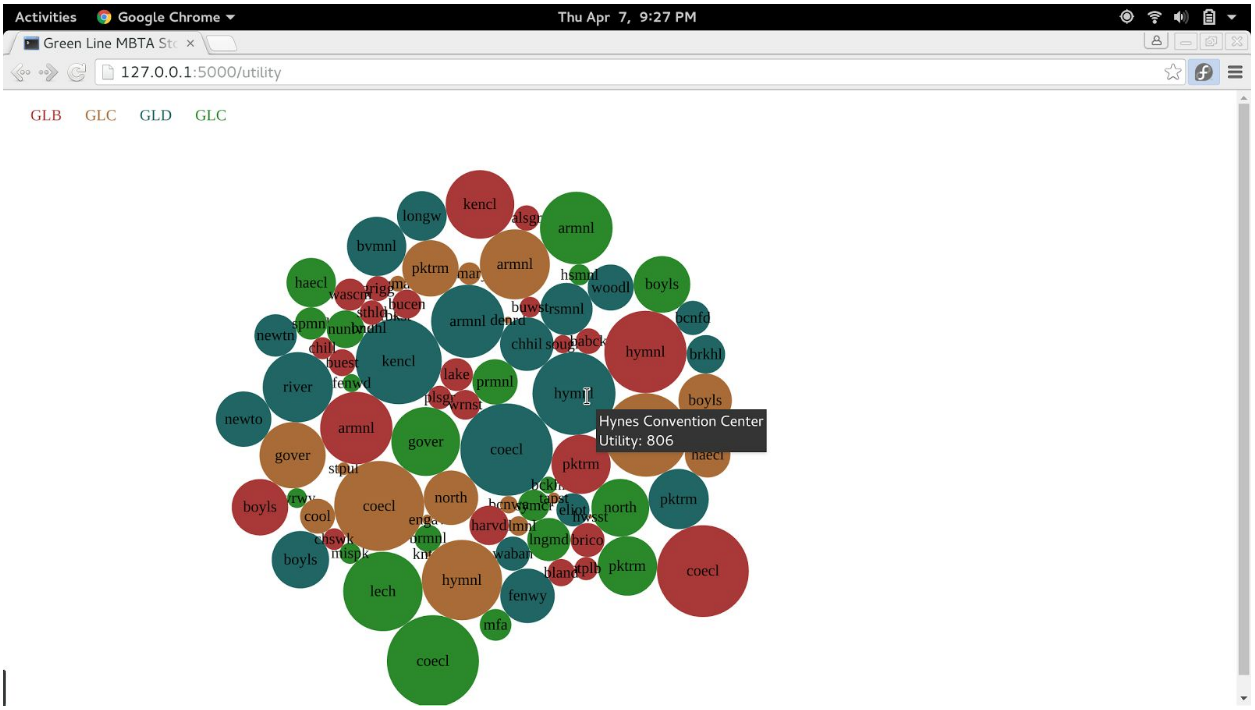
Optimal_Stops.html



This visual shows optimal stops based on different calculated *k*-means scores. Each of the original stops are represented by a transparent circle of corresponding color: GLB is red, GLC is green, GLD is brown, GLE is blue. Each optimal location will appear as a solid circle of corresponding color with a black border.

Initially, the map shows the current existing Green Line stops, where *k* = 0. By playing with the sliders and changing the value of *k*, the optimal *k* stops for that branch will change. In this visual, we have *k* set to 15,14,13, and 6 optimal stops for lines GLB, GLC, GLD, and GLE.

Utility.html



Hynes Convention Center
Utility: 806

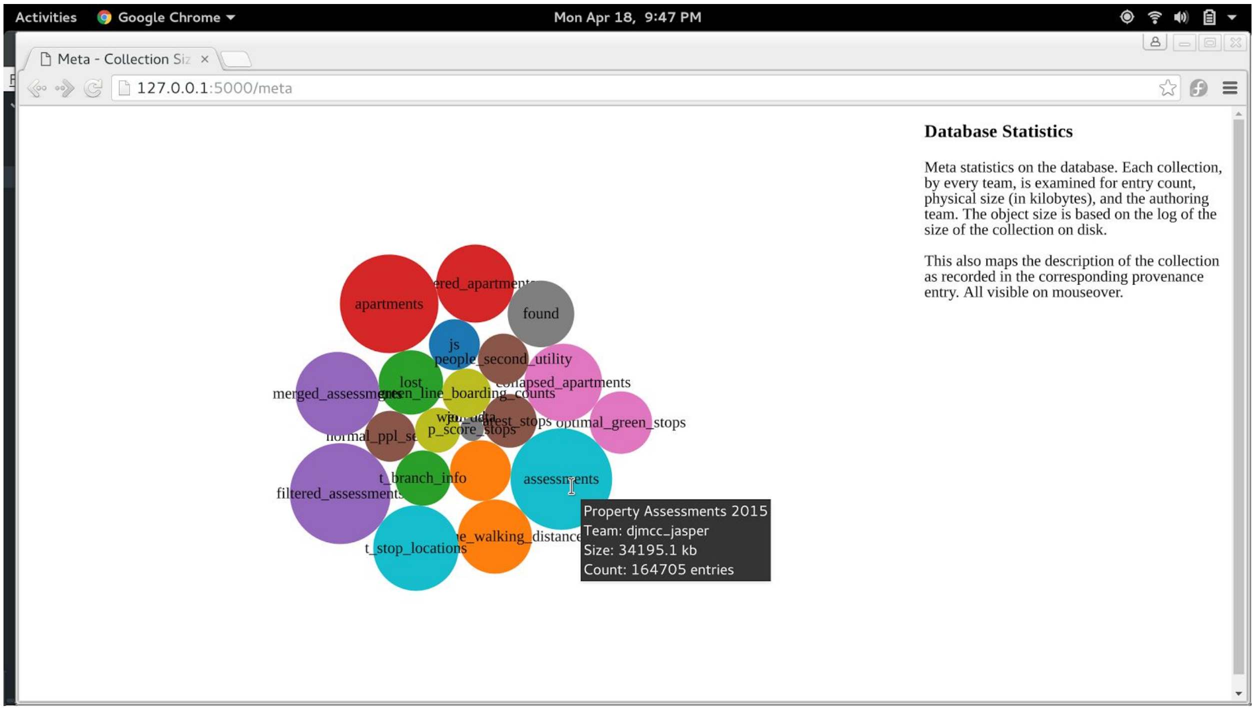
Our utility measurement is a score that uses both the popularity of each stop and the walking time to the next nearest stop. A high score is good[11]; this means that the stop is valued and saves the greatest amount of time for the collective commuting group. The measurement is scaled from 1 to 1000.

The size of the bubble corresponds to the utility measurement; the larger the bubble, the greater the utility measurement. The bubbles are labeled with their stop ID, and colored by line: GLB is red, GLC is brown, GLD is blue, GLE is green. When the mouse is hovered over the bubbles, a tooltip will appear showing the stop name and utility measurement for that stop.

Some stops are repeated with different utility scores and for different branches. This is because each of these stops may have a different importance based on the branch.

[11] The normalized *people seconds* scores are used and after measurements are normalized, higher scores are more desirable.

Collection_Stats.html



Database Statistics

Meta statistics on the database. Each collection, by every team, is examined for entry count, physical size (in kilobytes), and the authoring team. The object size is based on the log of the size of the collection on disk.

This also maps the description of the collection as recorded in the corresponding provenance entry. All visible on mouseover.

Property Assessments 2015
Team: djmcc_jasper
Size: 34195.1 kb
Count: 164705 entries

This visual show some meta statistics on the database. Each collection, by every team, is examined for entry count, physical size (in kilobytes), and the authoring team. The bubble size is based on the logarithmic size of the collection on disk.

This also maps the description of the collection as recorded in the corresponding provenance entry. This data is visible on mouseover.

Future Work

Coordinate with pedestrian walk signs and street lights.

Currently, the pedestrian walk signs and street lights do not coordinate with the T; while the street lights are green and the pedestrian walk signs are red, the T will drive through and not wait for the pedestrian light to change, leaving the pedestrians stuck waiting for the next train. Similarly, the T gets stuck in traffic; when the street lights are green and cars are making turns, the T is stopped at a red light until the street light changes.

We would like to compare the time of the T in its current state versus the time of the T with light coordination. We hypothesize that with the T, pedestrian walk sign, and street lights coordinating, that the T will be faster and more efficient.

Find nearest existing stops for each k.

We could extend our findings to correlate each optimal stop with the nearest current stop. This could be more useful for city planning, as it would use existing stops rather than creating new stops (though the MBTA's proposal did include creating entirely new stops).

Consider stops on different branches that are close in proximity.

We would like to consider different stops within the green line that are close in proximity. For example, geographically, the Chestnut Hill Avenue (GLB), Cleveland Circle Station (GLC), Woodland Station (GLD) stops all are very close to one another. When we increase the value of *k*, we see that the optimal stops near these three stations are also very close to one another. We would like to consider the possibility of combining these three stations into one and only having one Green Line branch stop there, rather than all three. Similarly, there are a few stops towards the outskirts of GLE that are all very close to each other. When we increase the value of *k*, we see that the optimal stops are still all close to one another. We would like to see if by spreading these stops within GLE out more if it would provide an optimal and efficient trip.

All Lines

Ultimately, we would also like to extend our project to all of the T and bus lines within the MBTA.

References

1. MBTA Subway Map (http://www.mbta.com/schedules_and_maps/subway/)
2. MBTA "Ridership and Service Statistics Fourteenth Edition 2014," pg 16-20 (http://www.mbta.com/uploadedfiles/documents/2014%20BLUEBOOK%2014th%20Edition.pdf)
3. MBTA Rider Tools, *MBTA GTFS* file (http://www.mbta.com/rider_tools/developers/default.asp?id=21895)
4. Google Maps API (https://developers.google.com/maps/documentation/distance-matrix/)
5. BU Today, "T May Eliminate Two Green Line B Stops" by John O'Rourke (http://www.bu.edu/today/2014/t-may-eliminate-two-green-line-b-stops/)
6. Google Develops, General Transit Feed Specification Reference (https://developers.google.com/transit/gtfs/reference#agencytxt)