# Using Alteryx to Explore Data: Bixi Residential Score

## Purpose

In this assignment I attempted to create a new scoring system for areas where Bixi is available. The purpose of this score is to quantify the “residential-ness” of a neighborhood/Bixi station. The theory behind this score is the farther users travel Monday-Friday between the hours of 5AM and 9AM the more likely that station exists in a very residential neighborhood (and being primarily used as a tool to get those users to work). Since long trips indicate Bixi being a mode of transport rather than simply a tool for tourists, I figured my logic was sound. The feature in question for this residential score is to find the median distance a bixi travels from a given station between the hours of 5AM and 9AM Monday through Friday for a year’s worth of Bixi data. The higher this score – the more residential the area.

## Workflow

To begin my workflow I needed to import all monthly trip datasets into one table. I accomplished this using the Directory tool in combination with the Dynamic Input tool to merge all monthly data. Once this was accomplished I needed to import the single-file containing the station data. With everything imported I could join the data on the start\_station\_code of each ride. With this join in place I now have a table of all rides during the year joined with station information on the starting location. After a column rename I performed another join on the end\_station\_code so that for each trip during the year I had the start and end station information. After (another) column rename I was able to calculate the quantity that comprises the meat of this score: the distance, for each ride, between the start and end stations. To get this value I used the following formula to calculate the straight-line distance between the start and end stations:

Although this distance doesn’t incorporate information such as the exact route taken or the distance travelled via street – I believe it to be a reasonable representation for the distance travelled during the trip. Continuing with the workflow I removed columns that weren’t relevant for the analysis and began to filter the data. Since I am examining rides taken during morning rush hour from Monday-Friday, I needed to remove all trips occurring on Saturday and Sunday, on top of any trips that did not start between 5AM and 9AM. With this information tabulated I was able to perform a groupby function on the starting station and return the median straightline distance travelled for each station (this median obviously only including data which passed my filter). I used the median because I didn’t want outlier distance values to affect the result of the score – and the median is less likely to be affected by these cases than the average. Since this groupby removed some information about the station I wanted to keep I performed another join function on the station data by start\_name to get the longitude/latitude information back for each station. With this table I sorted each station by distance travelled in descending order and ouput it to .csv.

## Challenges Faced

A challenge I faced early on in this analysis was a simple merge of the years’ worth of ride data in Alteryx. While I must admit that viewing the workflow now the solution seems sleek it is not an intuitive solution (I wouldn’t have thought to combine a directory with a dynamic input tool without explicitly being told to do so). A challenge I faced (that I did not end up implementing a solution for) was to dynamically normalize my residential area score between the bounds of 0 and 1. In data cleaning tools such as numpy or pandas this is fairly simple – just divide all values of that column by the maximum value of said column – but I could not find a sleek solution to this in Alteryx. I did see a couple suggestions incorporating multiple tools to complete this goal but for the simplicity of the workflow I didn’t deem it worthwhile (not having this score normalized didn’t seem like a big deal at this juncture). I figured the formula tool would help me here but I wasn’t able to utilize it this way.

Given more time I would like to have simplified my workflow and devised my own rules-of-thumb for how to address common junctions. For example I use 3 joins statements in my workflow when my suspicion tells me I only really need 2 (I use a third to regain information lost in a groupby and my intuition tells me there’s a better way keep this data). An example of a rule-of-thumb I would like to consider is where and when do I rename my columns: should I always perform this activity at joins? Or should I wait until the end of the workflow to clean it up before export? I suspect that there are common practices that people recommend following for certain actions that I have not implemented.

All things considered I do believe that my solution is very scalable – all the user would need to do would be to add more trip data to the same directory and the workflow would merge that in with the dataset to add even more data to this scoring solution. One of the reasons I avoided hard-coding dividing the residential area score by the current maximum of the set was for this reason: if I received another batch of monthly trip data perhaps this maximum would change and sully my workflow.

To continue with this investigation I would like to have normalized my data between 0 and 1 and include some visualization to represent the score on a station map. I believe the ideal representation of my scoring system would be to interpolate the scores between stations on a heat map which would layer on top of a map of the city of Montreal. This way the user will be able to extrapolate the “residential-ness” of the Bixi station to other parts of the city that maybe lack Bixi stations (say an address between two stations). With this sort of interpolation one could argue that this score doesn’t only apply to addresses near Bixi stations but to any address in the city of Montreal (within a given proximity of the downtown core). I was looking for how to implement this type of visualization in Alteryx and was falling short of my goal – Tableau definitely seems like a superior solution for these sorts of visualizations.