Bike Sharing Data: Detailed Write-Up

Mier Chen(mc5w x) / Katherine Qian (kq3zm) / Samuel Beadles(sbb5ur) / Zhiwei Zhang(zz3px)

I. Description of data and sources:

We used two main sources of data: bike sharing data and distance data:

The bike share data focuses on Pronto Bike Share in Seattle. This system consists of 500 bikes and 54 stations located in Seattle. We used outside libraries to calculate distance, which we use to supplement the bike share data.

Bike Sharing Data Set

Source: We sourced our data from Kaggle.
Content: The data comes as 3 datasets that provide data on the stations, trips, and weather from 2014-2016. The data came in three separate Excel files. The files, which we named data/trips, data/stations, and data/weather, contained information on individual trips, stations, and daily weather conditions, respectively.
Size: station.csv: 5.19 KB / trip.csv: 45.5 MB / w eather.csv: 55.1 KB. Trip.csv is the largest set: it has approximately 300,000 observations.
Columns of interest:
Data/trips: [“starttime”,”stoptime”, “tripduration”, “from_station_id”, “to_station_id” , “usertype”, “birthyear”]
Data/weather: [“Max_Temp”, “Min_Temp” , “Max_Gust_Wind_Speed” , “Events”]
Data/stations: [“station_id” , “lat” , “long” , “name”]

Distance data

We calculated distance by combining information from data/trips and data/stations: data/trips gave us origins and destinations and data/stations gave us coordinates. We think that the reason it is not included is that the bikes do not have real-time GPS trackers precise enough to take this data. Additionally, bike users will not reliably go directly from one station to the next. We still think, however, that these data will help us understand how users interact with the system. We developed two approaches to calculating distances between routes, described in the relevant data analysis section.

II. Data Cleaning

Deleting hidden duplicates in “trip_id”:

The unique “trip_id” identifiers in the data/trip had duplicates. We located and deleted the extra rows. We also checked data/station in order to confirm that the same data duplication had not occurred there. The original set had 286,858 observations: we eliminated ~50,000

Converting times to datetime:

Both start-and-stoptime came to us in separate DataFrame columns in the format “10/13/14 10:48”. In order to better compare the dates across multiple years, we decided to change the format into datetime.

In smaller datasets (<300,000 entries) this process worked fine. With larger datasets, e.g. from New York City’s Citi-bike share set of about 9 million entries, we have to explore other options or have prohibitively long run times.

Cleaning names of events: e.g. Rain, Snow → Rain-Snow:

Only unusual events, such as rain, had entries. Nan entries indicate sunny temperatures.

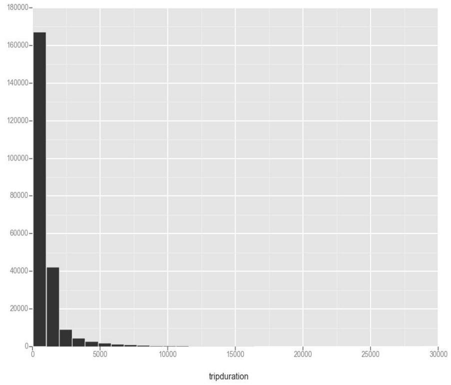
Creating 6 buckets of temperature, spaced by 10 degrees:

Using max and min temperatures, we found reasonable degree categories for weather.

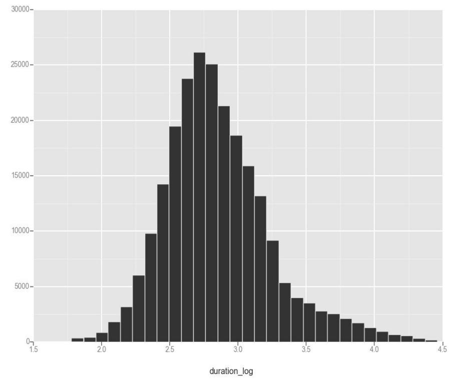
Eliminating outliers in the “tripduration” column of data/trips:

The data have wide variation: the maximum trip duration is approximately 29,000 seconds (almost an 8 hour ride), and the minimum is just 60 seconds. The data also show a serious rightward bias (FIG 1): that is, most trips are short and sweet. Mean trip length is 1178 seconds and median is 624 seconds (including outliers). For this dataset, the median of approximately 10 minutes will be more useful.

In order to eliminate outliers, we took the log of the “tripduration” observations. We selected the middle 50% of observations (FIG 2).This process eliminated 8111 outliers, approximately 2.8% of our data set. We decided that this was an appropriate action: the possibility that bikers forget to dock or incorrectly dock their bikes may have been skewing our observations.



(Fig 1)



(Fig 2)

Cleaning station names (E.g. Burke Museum / E Stevens Way NE & Memorial Way NE => E Harrison St & Broadway Ave E):

Cleaning the station names made the later analysis more it simpler and easier to manipulate, especially in calculating distances.

Creating merged DataFrame of data.trip, data.station, data.weather: named data.cycle:

We began by using ensuring that ‘starttime’ and ‘Date’ were both the same type: datetime. We merged the trip DataFrame and weather DataFrame on ‘starttime’ and renamed it. We then made two copies of station_df. Using those two copies, we left merged twice on ‘from_station_id’ and then ‘to_station_id’ in order to create our final DataFrame, data.cycle.

Optimizing runtime:

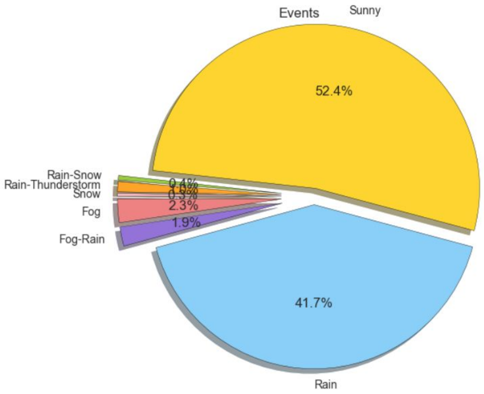
Because we were dealing with a relatively large dataset, several optimizations were done to allow for faster/more efficient run-time. Two practices that helped reduce runtime was making a deep copy of the dataset, such that the dataset is now stored in the memory location (for example, `station_df = station_df_raw.copy(deep=True)`), and ssing lambda (`trip_df['stop_year'] = trip_df['stoptime'].apply(lambda dt: dt.year)`), reducing the use of functions/methods, instead creating one-time use methods taking up less memory space and reducing runtime.

III. Data Analysis

We analyze the data down two main lines: by membership status, and by weather. We will begin by discussing our findings on weather patterns and bike sharing, and then continue on to the analysis of usage by user type.

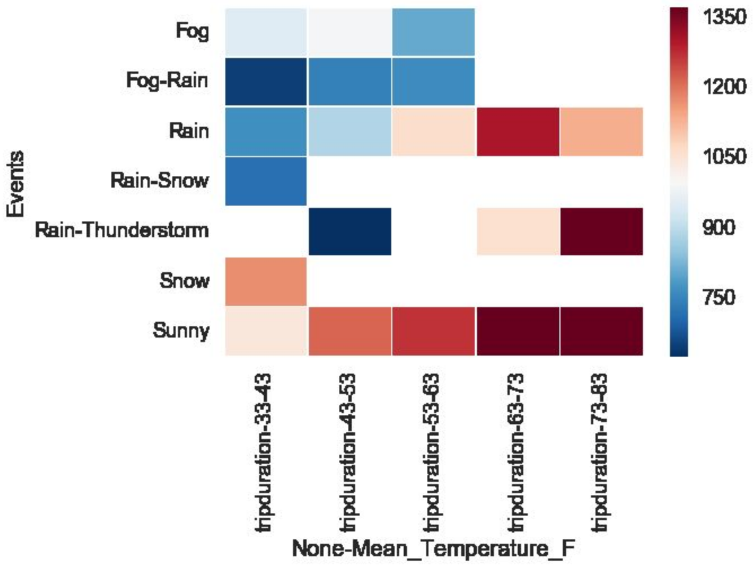
Weather

Our data shows serious bias toward two weather events in particular:



More than 94% of all events fall into two of the categories: rainy and sunny. This bias alters our analysis of the other events, since their sample sizes are so low. Some of the later analysis will not be useful for such a small sample size. However, we can look at the data in a broad sense.

Heat map:

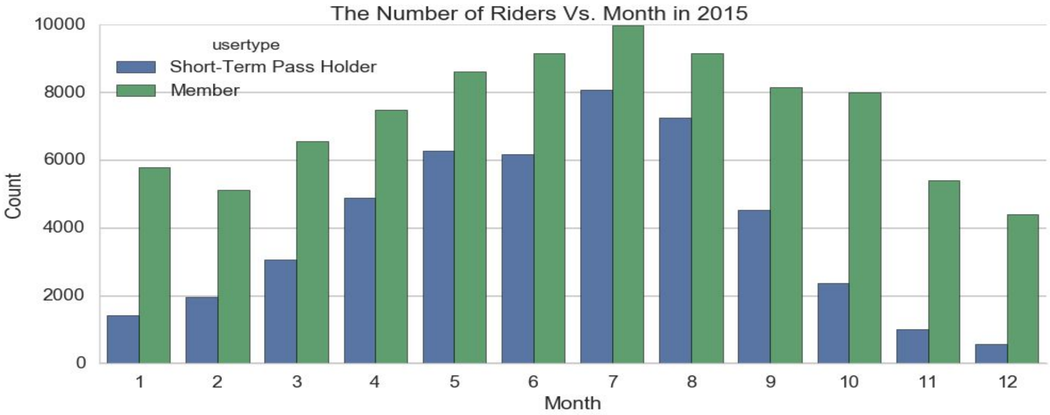


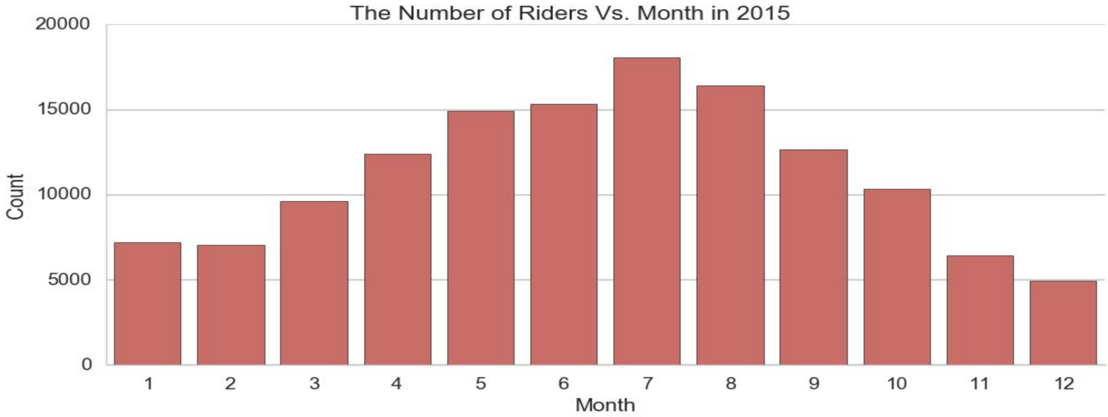
The heat map illustrates differences in duration in different temperatures and weather conditions. The white areas represent the conditions in which there was not any data for that conditions (for example, there will not be any snowy conditions when the temperature is higher).

Generally, people tend to use the bikes for longer periods of time when the weather is warmer and sunny, as shown by the dark red squares. At a lesser duration, warm rainy days also take the bikes out for a relatively long time.

An abnormality that emerged with the heat map was that there was a high number of usage when it was warm and raining/thunderstorming. This could possibly explained by people waiting out the thunderstorm, unable to return bikes in such conditions. In addition, since rain-thunderstorm only consisted of 1% of the entire dataset, so certain entries can heavily skew the results**.**

Users show a pronounced preference for biking in the summer months of June, July, and August:





From the first graph, we see the general trend in use of the bike share program; as the weather gets warmer in the summer months, usage increases. (This trend confirms the observation above given by the heat map)

We decided to analyze this graph further. We group it by pass-holders and members: the sum of both columns in the grouped graph equals the ungrouped graph. We observe that the member variation across months is much less pronounced. The standard deviation of members is almost double that of short term pass users. This observation makes sense: short term pass holders most likely tend to flock to Seattle, and Pronto, during the warmer summer months. The range of pass holders fluctuates by a factor of almost 15. The range of annual members ranges closer to 41% of the highest amount.

Analysis by user-type:

In order to complement the bike share data and make use of trip duration, we calculated trip distances in two ways:

Google API (Application Program Interface): Google API allows programmers to access information available in Google Maps in forms more conducive to using with code. Using this API requires several steps. This process requires two outside libraries, *requests* and *json*.

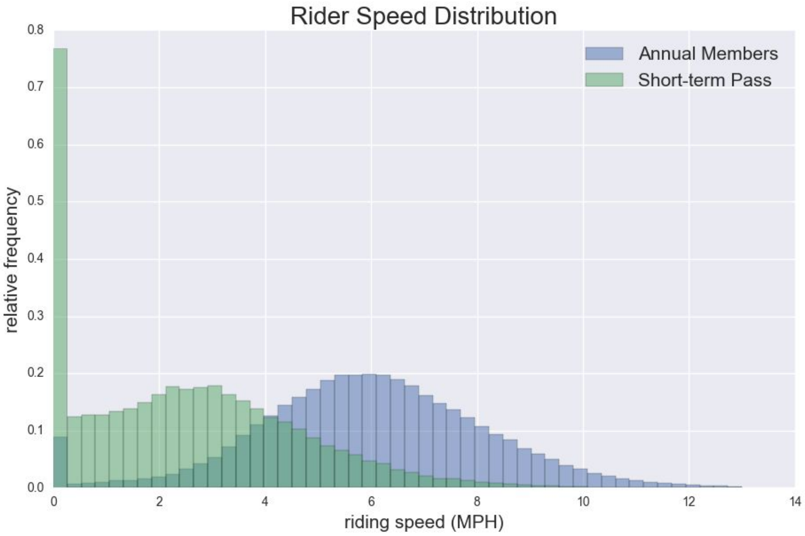
First, we used the data.stations information on latitude and longitude to populate the data.trips dataframe. With that information, we requested Google API's Distance Matrix between these two locations. The request occurs through a URL: we inputted each specific location in a particular format and the Distance Matrix returned the data in JSON format. The second part of this calculation was parsing the JSON file, which comes in a very nested format of dicts/lists. Given the distance element in meters, we populated data.trips with distances.

Geopy: We used a built-in library in order to calculate the direct distance. The library is geopy.distance, and the function is great_circle. We created a matrix for the 58 options and combined that matrix with the data/trip dataset. This library calculates distances as the crow flies.

In the end, we chose to use the less accurate geopy method due to a lack of efficiency from the Google API method. With a dataset of 300,000 observations, querying a URL became overburdensome.

Observations:

Rider speed data, while imperfect, gives us insight into member/pass holder behavior:



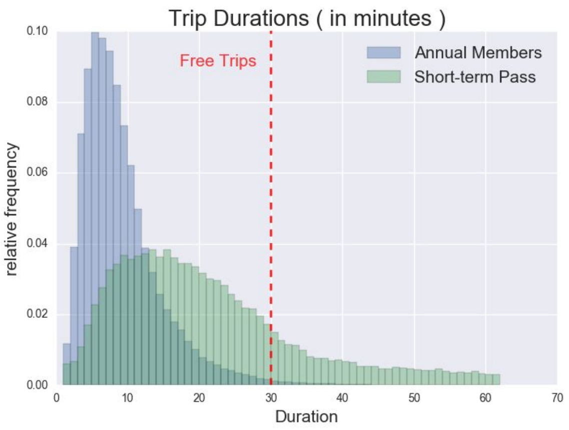
The graph of rider speed distribution, at first glance, suggests that annual members ride at speeds more than twice that of short-term pass members-- 5.9 MPH vs 2.5 MPH, on average. This conclusion seems suspect: what could explain such a significant difference? Our understanding of the origin of this graph, however, may suggest that speed will vary by purpose.

That is to say, this graph indicates that short term pass members take longer to get to their destinations. Instead of slower biking speeds, this probably means that short-term pass holders tend to use the bikes for tourism, rather than utility.

We also observed a difference between average distance for the types: on average, an annual member rides .85 miles, whereas pass holders ride .779 miles. This data does not necessarily contradict the previous paragraph. It suggests that members ride farther, but also more directly. Pass holders take a longer time to go a shorter distance.

These complicated observations come from our inability to directly link distance to duration: we have to guess at the relationship between the two. Distance is not an extremely useful instrument to predict speed because it is confounded by user demographic differences.

The program's fee structure affects pass-holders much more significantly than annual members:

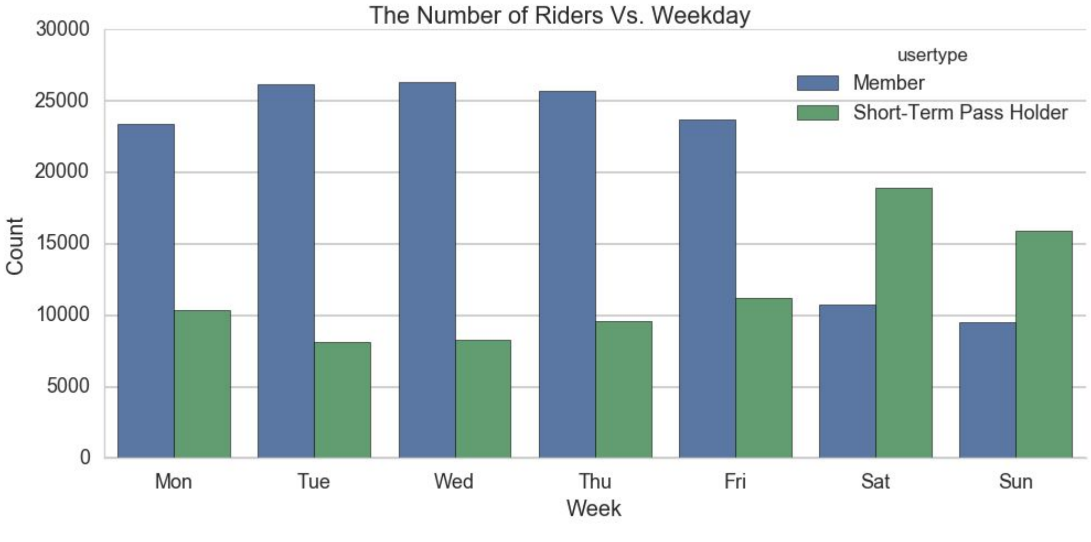


This bike sharing program charges short-term users a fee for bikes used longer than 30 minutes (shown by the dashed red line). Annual members pay a fee after the first 45 minutes. Annual members, for the most part, do not exceed their fee boundary of 45 minutes (only 0.58% of annual observations are after 45 minutes), with a mean and median of 36.50 minutes and 20.83 minutes, respectively.

Pass holders, however, are less sensitive to paying time-based fees. Most still spend less than 30 minutes: pass holders has a mean and median of 9.88 minutes and 7.98 minutes, respectively. We also observe that short-term holders do not have a cliff at the 30 minute mark, as 30.91% of short-term pass holders still continue using the bikes beyond the first 30 minutes, which indicates a relative inelasticity of demand. 12.6% of pass-holders ride longer than 30 minutes. In other words, pass holders are willing to pay for the extra minutes.

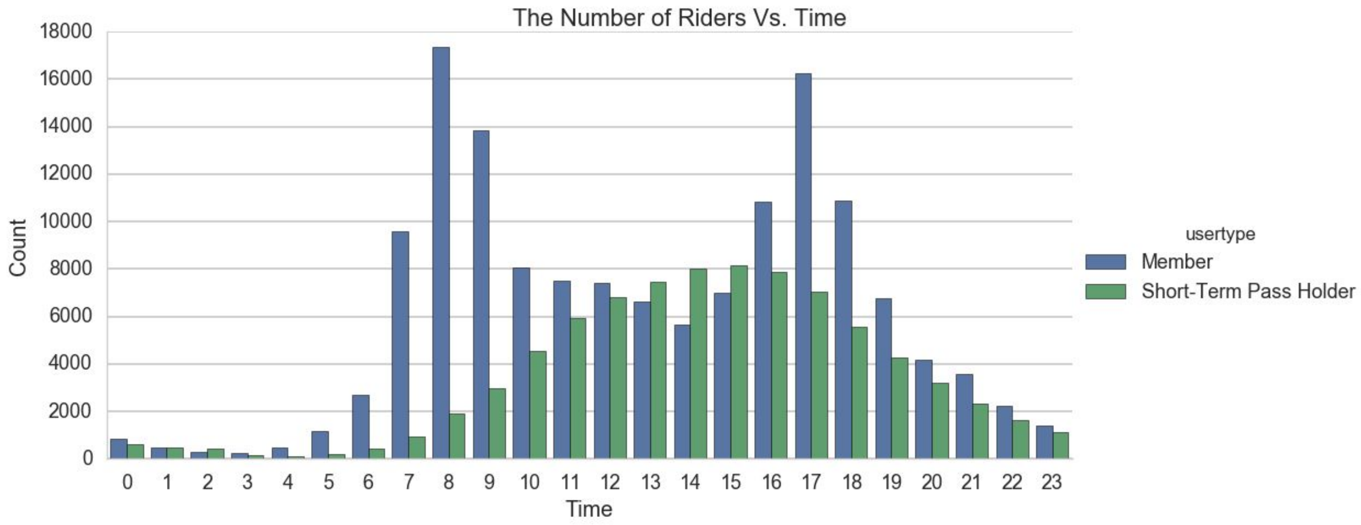
One potential recommendation would be for the program to move the fee-incurring time for pass holders to earlier in the ride. This change may increase revenue. Given the low elasticity, the program should not expect serious changes in short-term pass holder usage. However, if competitors will take advantage of this price increase, this may not be a useful change.

Bike sharing behavior depends on days of the week:



Approximately twice as many short-term pass holders ride bikes on the weekends than during the week. This tendency is reversed for annual members, who tend to ride during the week. The program may be able to take advantage of this trend by customizing rates on weekends and during the week.

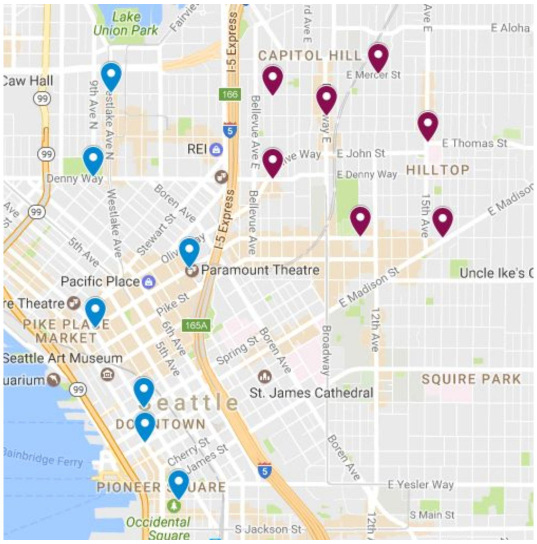
Bike sharing behavior by time of day:

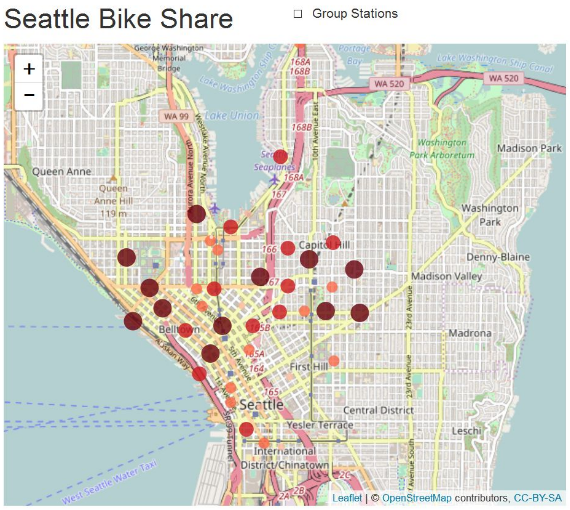


Further breakdown by day shows the huge difference in peaks throughout the day between members and short-term pass holders.

This trend could be explained in combination the graph of breakdown by week as the annual members could be using the bike share system to commute to work, and therefore have the two peaks in the morning and the afternoon. Short-term pass holds would not need to commute to work everyday/ have higher use on the weekends, where there are no structured schedules, other than restrictions such as daylight, which is shown by the peak during the daytime.

The bike origination/destination distribution is not even around Seattle:





We wanted to explore where bikes end up most often. To calculate this number, we found the net amount of bikes that ended up at particular stations over this period of time. On the above graph blue indicates a negative net (more taken than dropped off) and purple indicates a positive net. Bikers tend to originate in the east and drop off bikes in the west.

The graph on the right displays the stations based on usage (darker colors indicate more usage). Usage has a more even distribution, especially in terms of east to west.

Limitations and further analysis:

While there are multiple data sets on other bike share programs in other cities, because of the amount of data entries, long run-times limited the amount of analysis that could be done. For example, New York City and Washington D.C.'s bike share data sets both had over three million entries for just one year. Future work with cross-city comparisons could help with strategies to increase use of bike shares in certain cities and analysis practices in other cities that could maximize efficiency.

As we mentioned in the analysis, our distance calculation runs into two problems. One, it calculates linearly. Two, we have no information on what the riders do in between stops: we can only make educated guesses.

Deeper analysis would be useful to quantify the impact of changes in pricing structure on Pronto's top line. This line of inquiry is outside the scope of our skill set, and project.