EXPLORATORY DATA ANALYSIS

NYC Metra subway data cleaning and analysis

*Finding the best place to sell flavored milk after a marathon*

By: Scott Turransky

**Abstract**

The goal of this project was to perform data cleaning and data assessment for a fictional scenario to solve a specific problem. The goal of the project is by using actual data, identify a solution to a proposed issue. This solution should be supplemented by charts, graphs and other visual aids.

This project was designed with a fictional customer of Nesquik. Nesquik was interested in setting up a pop up booth at a metra station near the finish line of the New York City Marathon. They are targeting the marathon runners and would like to know best stations to set up their popup booth. The station they select would be based on which station sees the most gains on race days.

**Design**

This analysis focused on several stations closest to the finish line. Seeing how the marathon finishers already ran 26.2 miles, it is reasonable to assume they would want to get on subway at one of the nearest stations to the finish line.

The marathon occurs the first Sunday of November every year (with 2020 being an exception due to the pandemic.) This assessment compares subway entries on race day with the 2 Sundays before the race, and the 2 Sundays after the race. This ensures the sample data occur during the same portion of the year (ie no summertime spikes) and are not affected by holidays, or M-F commuter traffic.

**Data**

MTA train data is downloadable by week. For this assessment 25 weeks were originally downloaded (2 weeks before race day, race day and 2 weeks post race for 2016,2017, 2018, 2019 and 2021.) If all the MTA station data downloaded was imported for the analysis it would create 8,660,638 rows of data. However by strategically devising the SQL query to select only Sunday data, it reduced the data frame down to 726,322 rows which is significantly more manageable.

Information was also taken from the New York Marathon website, including a map of the finish line showing close MTA location (see below.)

Map

Description automatically generated

This assessment looked at 10 stations near the finish line, and 1 station nowhere near the course as a comparison. Of the 10 stations, 4 saw a visible spike. Those stations were assessed further to see which station saw the greatest impact.

**Algorithms**

The following algorithms were utilized in cleaning the data:

* A function was developed to find first Sunday of November (race day)
* Timedelta addition to find Sundays before and after race day.
* Strings in the database were stripped of excess white space, and the replace method was used to replace symbols with a more pythonic replacement (“ “, “\” and “-“ were replaced with “\_”
* Data masks were used to reduce the dataframe to smaller subsets.
* String concatenation was used to merge C/A, Unit and SCP columns into a single column (listing each turnstile as the concatenation)
* The number of people at a station can vary year to year, so it was helpful to have a line plot that compared entries on race days to the weeks prior/after the race. To effectively manage this, a function entries \_to\_dict was created to add the data to a dictionary where each year was a key.
* The data for all 11 stations were plotted for all 5 days on a the same graph for all 5 years. All 5 years showed a spike compared to the baseline of subway entries for that year.
* The entries\_to\_dict function was modified so that it was a dictionary of dictionaries. This now included stations as keys. This allowed plotting each station over the 5 years.
* Using a for loop, the data for each station was plotted looking for spikes on race days.
* 5 stations consistently showed a spike on race day, and therefore were impacted by the race.
* Aggregations were performed on each station. From this it was identified how many extra people showed up on race day at each station, and what the percentage increase was. This helped to identify the stations that should be recommended.

**Tools**

The following tools were used in this analysis:

* The analysis was performed in python. NumPy and Pandas were used in Python for the data manipulation.
* SQLAlchemy was used to import the results of a SQL query into the Python file.
* Matplotlib was used for plotting the results.

**Communication**

In addition to the slides and visuals presented, the results will be posted on my personal github account.

**Final Results**

While 86 ST had the largest average percentage increase in entries every year, it is a slower station that sees less foot traffic in general. 59 ST Columbus had a large increase in people but it already is a very crowded station, and it only sees about a 40% increase in entries to the station.

Our recommendation is for the Chocolate Milk Popup to be at 66 St Lincoln Station, which sees an average increase of over 3500 people per year, which is a 73% increase when compared with the other entry points in the year.