Capstone Project 1: Data Wrangling

Weather Data Cleaning

The 2016 NYC weather data was originally acquired from the Wunderkind API, and provided here.
https://www.kaggle.com/meinertsen/new-york-city-taxi-trip-hourly-weather-data

First we dropped all the columns that we do not think that we'll use.

As we check the data, we notice that it's fairly well organized. The rain and snow columns are are separate boolean columns. The data would be easier to read if it was a single column with a categorical values, so we combine the two columns into one precip_type column with values 'rain', 'snow', or an empty string for clear weather.

The simplified weather data is saved as weather_edit.csv.

Taxi Data Cleaning

The taxi data in NYC was acquired from NYC OpenData
https://data.cityofnewyork.us/dataset/2016-Yellow-Taxi-Trip-Data/uacg-pexx
With this set being 131M rows, some bandwidth issues were experience so we decided to download only the taxi trips where VendorID = 1, giving us 61M rows of taxi trip data.

First we drop the columns that we do not need, mainly leaving us with the pickup and dropoff datetimes and the trip distance. With the goal of exploring the relationship between weather conditions and automobile travel times in NYC, we calculate the average speed for each trip. To get the time for each entry, we first convert the pickup and dropoff columns to datetime objects, and subtract them to find the time difference. Then we take the trip_distance and divide by the time to get the average speed.

As we check the speed column, some values make no sense. The max is inf and the min is negative. So we drop the rows where the speed is less than 0, as well as greater than 60.

The simplified taxi data is saved as taxi_edit.csv.

Merging the Data

The big problem we had was the sheer size of the taxi data set. Any sort of manipulation or processing would run into incredibly length run times due to some hardware limitations. So we decided to approach the data in two ways, one was to aggregate the weather by day, and match

it with a sample of the taxi data. The other was to choose a few specific days where it either rained or snowed and compare the taxi trip data with the weather data as recorded.

Selecting Months 1 and 9

With the weather data, we made the assumption that if it rained for more than 20% of the day then we would consider that day as having rained. We group the weather data by the month and day, and aggregate temperature by average and the precipitation as a total. Then we assigned the grouped data a categorical value of ‘rain’ or ‘snow’ or empty string.

With some rough data exploration, we can see which months had how many clear days, rainy days, and snow days. We can use this information to choose a sample with a variety of characteristics. We choose January and September.

Using months 1, and 9, we then group the taxi data by month and filter for the months we want. We also filter the weather data for months 1 and 9.

Finally, we merge the two data frames on the date column.
The resulting merged data is saved as data_merged.csv

Selecting Dates 2/24 and 3/21

From the exploratory analysis above, we have some basic visualizations showing the instances of rain/snow for each day. We chose 2/24 and 3/21 since they held a reasonable proportion of rain and snow to clear instances recorded.

First we grouped the weather data by month and day, and returned the desired grouping (2/24, and 3/21). We did the same for the taxi data to get the trips that occur on the dates 2/24 and 3/21.

Each weather pickup datetime is assigned a value from the index. For each taxi data pickup datetime, see which weather pickup time is closest to the taxi pickup time, and assign it the value ID from the weather data. The two data frames are then merged on this ID column.

The resulting data is then saved as taxi_rain.csv and taxi_snow.csv.