Dataset Observations

**Note**: This file is meant to keep a list of observations on the dataset as the project unfolds.

Link IDs and Link Points:

* 153 link IDs
* 152 link points
* 2 link IDs have two link points or one link ID has three link points?
* Remove link points with much fewer (typically 2) observations to clean out outliers
* View on map the scope of the project for clustering
  + Multiple Polylines may be on top of each other but going in opposite directions
  + How to see multiple Polylines
  + For the same link points, there could be multiple Polyline levels
* Filter data by link points, get encoded polyline with lowest number of characters
* Segregate data into files by link points, find the one with the lowest number of characters, replace the encoded polyline with that
* Extract date from dataset
  + Minute, hour, month
* Machine learning models based on time itself
  + Filter data by link points, get encoded polyline with lowest number of characters
  + Segregate data into files by link points, find the one with the lowest number of characters, replace the encoded polyline with that

Data Cleaning:

* Keep ID, Speed, Data\_as\_of, Link\_ID, and possibly travel\_time
* Extract hour, minute, second to run the daily linear regression model
* Remove null speeds
* Outliers?
  + Average speed outliers (maybe too low or too high?)
  + Create a histogram to get an idea of the average speed’s distribution
* Correct Encoded Polylines
* Remove status and Transcon ID (stated to not be useful)
  + Remove any other potentially useless columns
* Explore “owner” column
  + Potentially remove records by owner if they have minimal records
* Explore Distribution of data by DATE\_AS\_OF column.
  + By Year, Hour of the day, etc.
* Remove data points by link name with few counts (< 45,000)

Linear Regression:

* Instantaneous Speed wrt to time (Variables: Speed, time/date) - Speed of traffic
* Change in Speed wrt to time interval (30 mins, 1 hour, 1 day) (Change in speed/average of speed for every time interval (e.g. negative speed means slowing down, meaning heavier traffic), time/date) - Flow of traffic/congestion
* Time Intervals:
  + For every (moment of time) or (period of time) how is speed/travel time/ related
* Link ID
* Maybe figure out inflow and outflow between localities (boroughs?) for another linear regression
* Visualization: Link Points
* Use datetime functions to get seconds, minutes, hours, days, etc. and put them into different columns in R. Then study the shift in speed between various time intervals, and see where these shifts occur in great quantities (e.g. at New Years there may be a lot of congestion due to the New Year celebration)
* Filter out rows, divide into four different times of day (0-6, 6-12, 12-18, 18-24)
  + Average speed throughout the day
  + Analyze change of speed throughout day
  + Create linear regression model for the four quarters of the day

Time-Series Analysis:

* For Daily Analysis: Before 2019
* For Annual Analysis: After 2019
* Learn about Time-Series.
* Identify what we can do for the data we have
* Group by year, average by speed
* X-Axis: Time
* Y-Axis: Speed, what we’re trying to predict
* Divide day into number of timeslots,
  + For every timeslot create a linear model
  + Observe variation throughout day
* Random Forest:
  + Get a small, random sample (1,000 records)
  + Base graph on speed and hour
  + Get coefficient
  + Use that to predict the traffic/speed throughout the day
* Everyday there is some sort of change, observe changes over the year
  + Observe changes with holidays and construction amongst other events
* One function for the change throughout the day, week, month, year

Clustering:

* May or may not work with it

Visualization:

* Animate in Tableau for average speed for LinkID through the time
* Find ways to show geographic distribution of data over time
* Intervals: Second, minute, hour?
* Line graph for major Link IDs to show change over hours
* Tableau dashboard for posting the predictions

Link ID Dataframe:

* Potentially download the data in a different format due to cut off columns/rows of data
* Store data in MongoDB due to cut off columns/rows of data
* Filter Link ID, Link Points, Encoded Poly Line, Encoded Poly Line Levels, Borough, and Link Name
  + Only columns in this dataframe, remove all others
* For each Link ID, we need Link Points
  + Link Points will be stored as an array
  + Get encoded polyline and poly line level for those link points
    - Potentially download the data in a different format due to cut off columns/rows of data
  + For every Link ID there should be a borough and link name
  + Get poly line and poly line level with the least number of characters
    - Lowest number of poly lines shows data, rest is junk
* Filtering records:
  + Remove all Link IDs with less data
  + Link ID may have multiple link points, link points may go to multiple link IDs, remove these records
* Get direction of Link ID from Link Name (e.g. South to West)