# Data Processing - short summary

## 1 Data Processing

### 1.1 Preprocessing Trip Records

**Python Script:** `preprocess_data.py`

**Data Source:** Monthly Trip Records (2014-2019) for Yellow Cabs, Green Cabs, For-Hire Vehicles (FHV), and High Volume FHV from `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`. Each dataset is stored in separate folders : `SourceData/Year/.parquet`

**Steps:**

1. Trip records including date, time, pickup location, trip distance, and total fare were aggregated into date-pickup location pairs.
2. For each pair, the number of recorded trips was counted (`trip_count`).
3. Averages for distance and fares were calculated for all trips within each pickup location-date pair.

**Notes:**

- Trips starting or ending at New York's airports (Zones 1, 132, 138) were excluded.
- Records with negative or unrealistic values for distance (> 200 miles) or fares (>= $1000) were discarded.

**Output:** Daily summaries were concatenated for each dataset, resulting in `merged_grouped.csv` for each data source. The csv includes columns for date, pickup location ID, average trip distance, average total amount, and trip number.

### 1.2 Pooling Datasets

**Python Script:** `pool_taxi_data.py`

**Input data:** `merged_grouped.csv`

**Objective:** Combine daily summaries from all four datasets, with options to pool only medallion datasets or FHV datasets.

**Steps:**

1. Concatenation of daily aggregations
2. Aggregation at the date and pickup location level, with trip counts summed and trip distance and total amount averaged, weighted by trip numbers.

**Output:** `data_grouped_PU.csv` with the same columns as before.

### 1.3 Prepare for Regression

**Python Script:** `prepare_for_regression.py`

**Input Data:**

- data_grouped_PU.csv
- weather_NYC_2014_2019.csv (Central Park Station, excluding days with missing wind measurements)

**Steps:**

1. Imputation of zeros for trip count in zone-day pairs with no records
2. Merging taxi data with weather data based on date.
3. Addition of variables such as year, month factors, weekday dummies, holiday dummies, and logged trip count.
4. Addition of time trends using Chebyshev polynomials (1st to 5th order).
5. Yearly outlier filtering: Iteration over each unique weekday and taxi zone for ridership distribution. Outliers identified based on the median $\pm$ 1.5 $*$ interquantile range adjusted by the square root of N, as standard in matplotlib.fliers package. Days where more than one third of taxi zones were marked as outliers were then dropped.

**Output:**

- Pooled_data/PU/final/final_data_subset_PU.csv : Dataset used for regression analysis: Aggregated by Day, merged with weather data, outlier filtered for subsets YG (Medallion taxis) and FHV (Ridesharing companies).

# 2 Matching ACS, Park and weather data to taxi zones

**Python Script** : weight_socioeconomic_data.py

**Procedure:** Since socioeconomic covariates from the ACS represent 5-year estimates on the ZCTA (ZIP Code Tabulation Area) level, I match them to the taxi zone level by creating intersection share and population weighted averages. Park and beach coverage of each zone was calculated from https://nycopendata.socrata.com/Recreation/Parks-Properties/enfh-gkve and https://data.cityofnewyork.us/dataset/Beaches/ijwa-mn2v which includes all parks and beaches managed by NYC's Park Agency. The park/beach coverage was defined as the proportion of a taxi zone which was covered by park facilities/beach spaces.