

# Predicting Citi Bike Availability in NYC

DATA 698 Research Project  
CUNY Fall 2022

Philip Tanofsky

2022-11-22

# Citi Bike Availability

Predict the count to avoid the empty racks



# Problem and Objective

v3: started 11:30A on Nov 22, 2022

Introduce yourselves and describe your problem. Explain your objectives, challenges of your work, proposed methodologies, and the assumptions you made while conducting modeling and/or analysis. Provide an overview of your approach and/or conceptual model (please do not present your code directly). Describe the results you obtain and summarize the current achievements and possibility of future works.

# Previous Work

- bullet 1
- bullet 2
- bullet 3
- bullet 4

# Challenges and Assumptions

Large volume of data

- Over 10 million trips in 3 months under consideration
- Over 1650 docking stations in NYC

Rebalancing

- System action of moving bikes to restock docking stations
- Identification and inconsistency

User-friendly availability prediction

- Independent variables to prediction model
- End-user input values

# Data Collection

## Citi Bike System Data

- Download CSV file
- API call every 15 minutes for two weeks

## NYC Open Data

- Borough shapes via GeoJSON files

## Docking Station Elevation

- R library elevatr

## Surplus/Shortage Timetable

- Convert trip data to dataframe of bike departure and arrival counts by docking station for a given time interval

# Data Analysis

## Ridership patterns

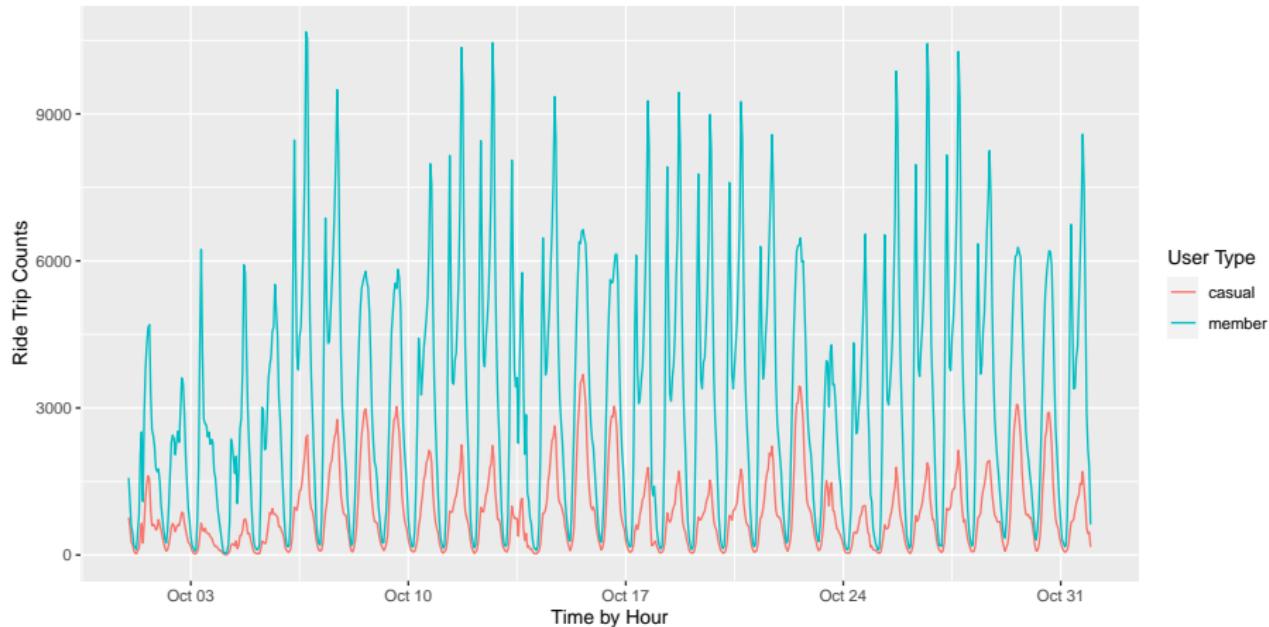
- Timeframe: August - October 2022 (92 days)
- Location: New York City departing trips
- Number of trips: 10,210,102
- Stations: 1,717
- Abandoned trips: 24,887

## Bike Availability

- Timeframe: Oct. 31 - Nov. 13 (14 days)
- Location: Brooklyn

# EDA: Bike Trips by Hour

Bike Trips by Hour by User Type for Aug. - Oct. 2022

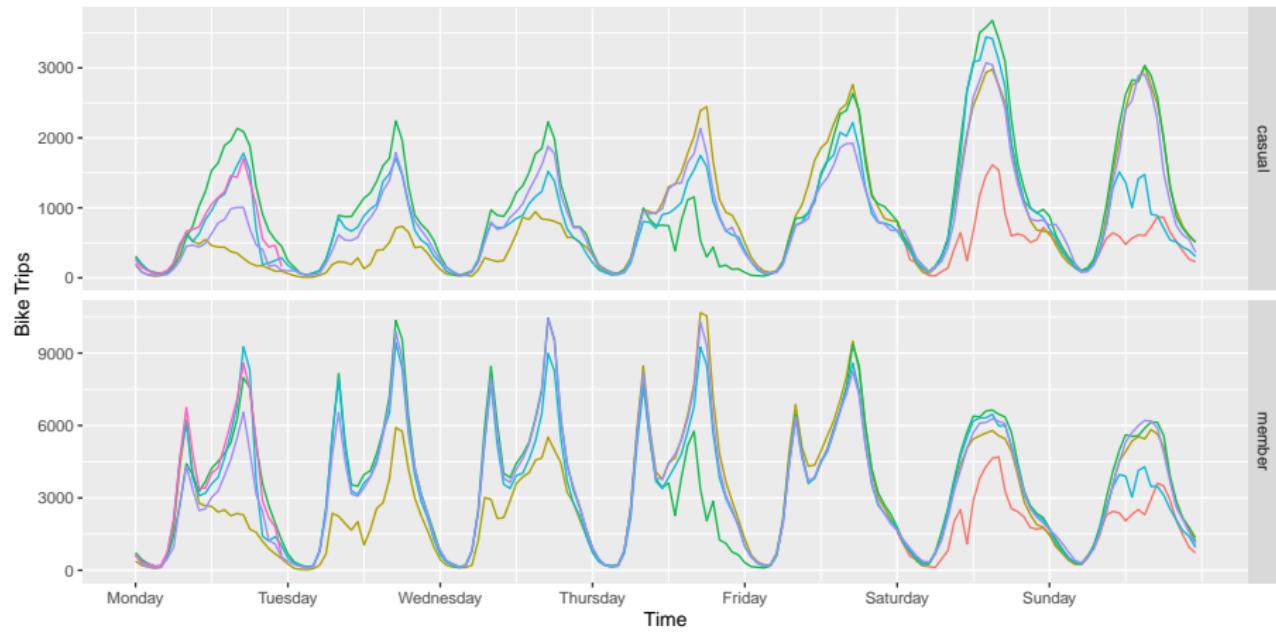


STL decomposition

cnt = trend + season\_week + season\_day + remainder

# EDA: Weekly Seasonal Pattern

Weekly seasonal plot of the bike trips for Aug. - Oct. 2022

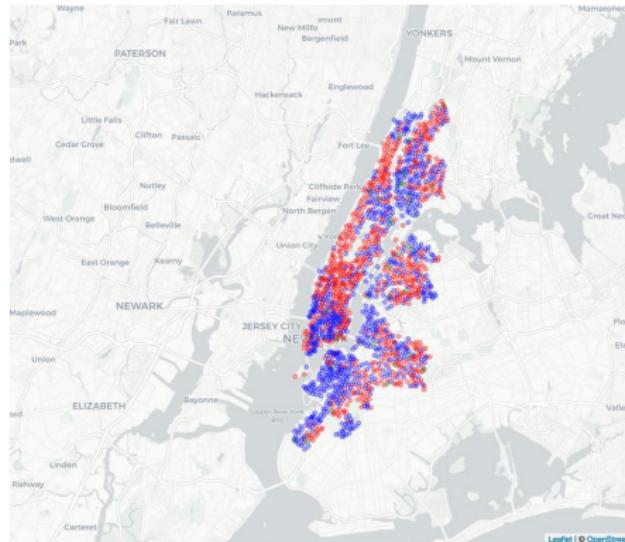


# EDA: Three-Month Summary

System-wide view of surplus or shortage for every NYC docking station

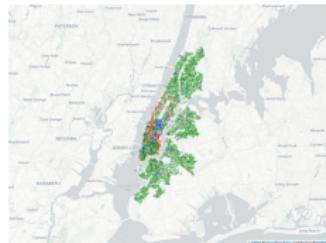
## Docking Station Color

- Blue: Surplus
- Red: Shortage
- Green: Even

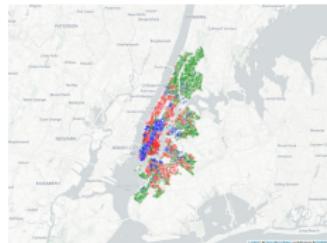


# EDA: Daily Pattern

Time lapse of surplus or shortage for every NYC docking station



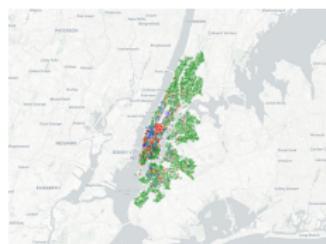
(a) 5 a.m.



(b) 8 a.m.



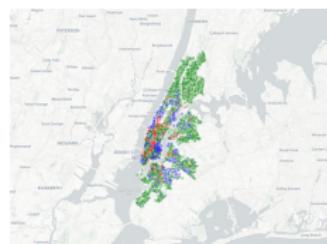
(c) 11 a.m.



(d) 2 p.m.



(e) 5 p.m.



(f) 8 p.m.

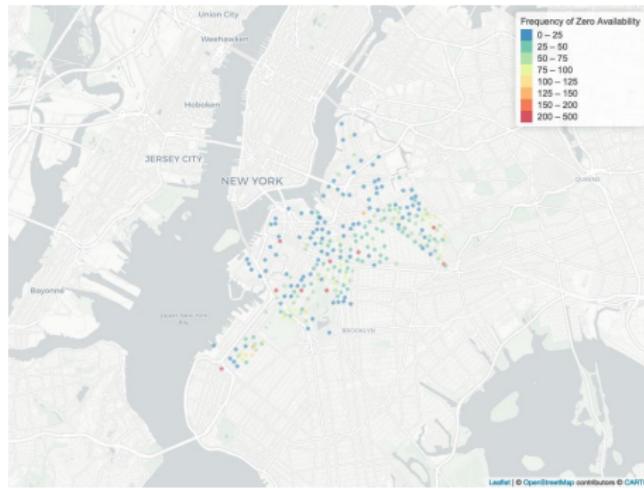
# Overview of Algorithm Approach

- Data from API call every 15 minutes for two weeks
  - Citi bike availability at each station
  - Two weeks is small interval to predict
    - Valid limitation of model
- Cluster all the Brooklyn docking stations
  - Distance: 400m (Quarter mile)
- Apply Generalized Linear Models for Count Data

# Stations with Zero Bike Availability

Frequency of zero bike availability per docking station in Brooklyn

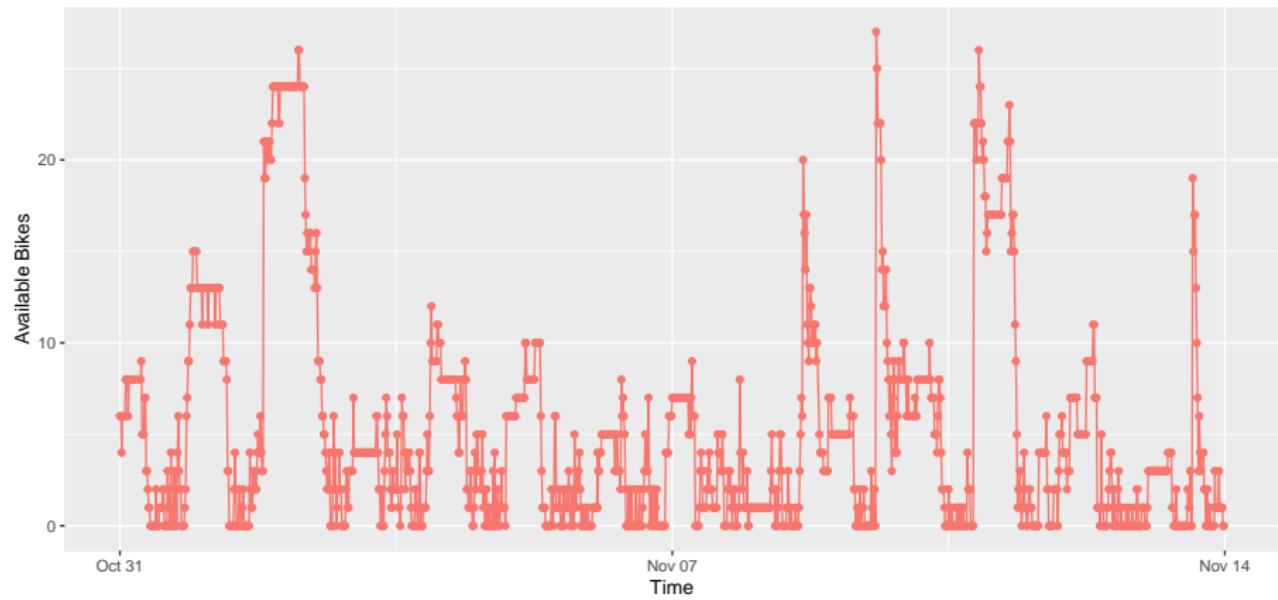
- Span: Two weeks
- Instance: 15-minute interval



# Rebalancing Example

Example of inconsistent nature of rebalancing

Docking Station 3582 in Brooklyn

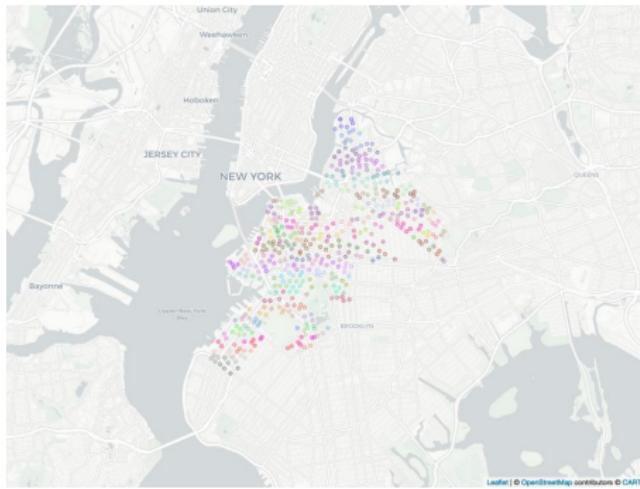


# Proposed Methodologies

- Inputs
  - Latitude and longitude
    - Convert to cluster
  - Day of the Week
    - Convert to Weekday or Weekend day
  - Time of Day (15M and 1H)
- Citi Bike offers live map of availability
- Lyft provides real-time availability
- Time series model not used explanation (or really, just don't include)
- Poisson distribution and Negative Binomial given the over-dispersion

# Modeling Step 1: Clustering Model

- Borough: Brooklyn
- Docking Stations: 474
- Clusters: 2XX

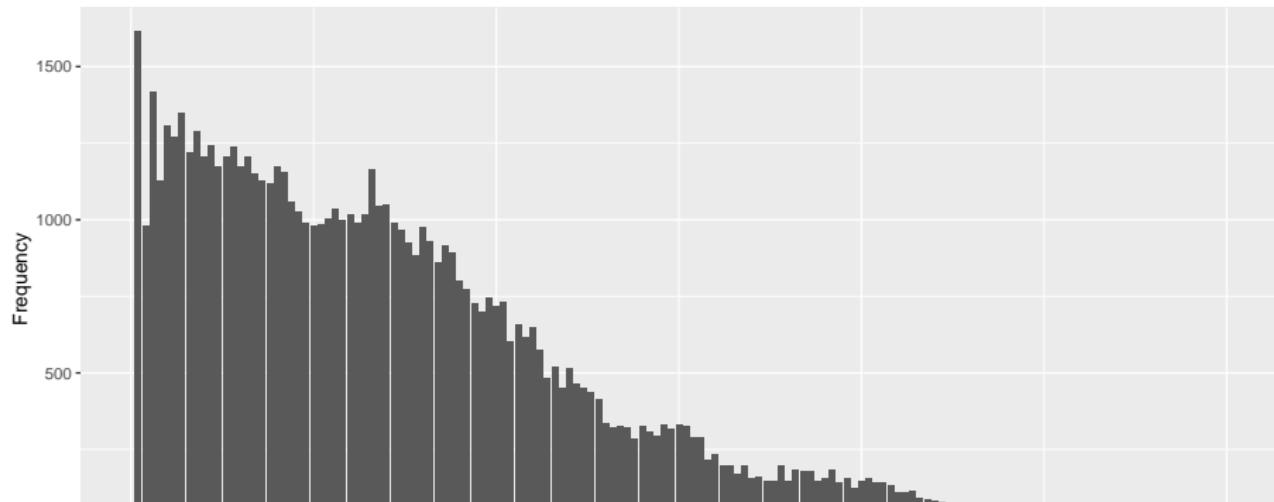


# Bike Availability Count Frequency by Cluster (Brooklyn)

Histogram of availability based on 1HR intervals

- High frequency near 0 and 2 and plateaus at docking station capacities

Frequency of Hourly Available Bike Counts for Brooklyn Clusters



## Modeling Step 2: Count Models

Generalized Linear Models for Count Data

- Poisson
- Quasi-Poisson
- Negative Binomial
- Hurdle
  - Poisson and Negative Binomial
- Zero-Inflated
  - Poisson and Negative Binomial

Zero-Inflated

$$Pr(Y = 0) = \pi + (1 - \pi)e^{-\lambda} \quad (1)$$

$$Pr(Y = y_i) = (1 - \pi) \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}, y_i = 1, 2, 3, \dots \quad (2)$$

Poisson Probability mass function

# REMOVE THIS SLIDE !!! THIS IS FOR 15 MINUTES

- Certain input to model based on 15 minute intervals above
- Results table

## Model Results 1-Hour Interval

- Certain input to model based on 1 hour intervals below
- Results table

# Model Results Visualization

- Plot of the preds and actuals for best scoring model

# Availability Prediction

- Model prediction
  - Show how it works

```
predict_num_bikes_avail(-73.960859, 40.67355, "08:30:00", "Sat")
```

TODO

```
predict_num_bikes_avail(-73.960859, 40.67355, "08:30:00", "Sat")
```

# Current Achievements

- Prediction model for Brooklyn
  - Clustering
  - Zero-Inflated XXX (Pois/NB) Model
- Capture patterns of Citi Bike usage

## Future Works

- Weather ... actually, can I predict weather? would that really work?
- Subway stations: Citi Bike offers valet
- Model of all NYC
- Real-time clustering would be better
- Greater amount of availability data
- Deployed web application or smartphone app