Project proposal

Group members:
- Jacqueline Gutman, jg3862
- Nasser Zalmout, nz658

Github: https://github.com/nasser-zalmout/big_data_project
Chosen project type: Explore Taxis
Option: Option #4 Yellow taxis, green taxis, Uber and bikes

Abstract:

We aim at exploring and analyzing the effect of introducing the recent public transportation means of yellow taxis, green taxis, and Uber. We will be studying the impact these new means have brought about to the overall urban movement in NYC.

We will begin by exploring the different datasets at our disposal covering the different transportation means, and potentially spanning the different NYC boroughs. At this stage, our goal is to achieve an understanding of the current patterns outlined in the data, whether in terms of geographic location, frequency of trips, trip durations, fare amounts, among others.

Our next goal is to run regression models to approximate future trends based on the available historical data, and to estimate the future market shares of these different new means, and potential growth rate aggregated by the different variables we obtain from our earlier analysis.

Data sources:

Yellow Taxis: April to September 2014
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-04.csv
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-05.csv
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-06.csv
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-07.csv
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-08.csv
https://storage.googleapis.com/tlc-trip-data/2014/yellow_tripdata_2014-09.csv

Green Taxis: April to September 2014
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-04.csv
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-05.csv
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-06.csv
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-07.csv
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-08.csv
https://storage.googleapis.com/tlc-trip-data/2014/green_tripdata_2014-09.csv

Uber pickups: April to September 2014
https://github.com/fivethirtyeight/uber-tlc-foil-response/tree/master/uber-trip-data

Subway entrances shapefiles
https://nycopendata.socrata.com/Transportation/Subway-Entrances/drex-xx56

NYC zipcodes shapefiles
http://catalog.opendata.city/dataset/nyc-neighborhood-tabulation-areas-polygon

NYC neighborhoods definitions
https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoods.htm

Status Report

We have run some preliminary investigation of the taxi files, to better estimate the workload and how we will distribute the effort among the team members and throughout the remaining time period. Some of the issues that we are able to observe include:

- There's a limited dataset shared across all taxi services. This period includes April-September 2014, and January-June 2015, so we thought of limiting the analysis to this period.
- The dataset misses the neighbourhood tag for each trip, where each trip record has the coordinates only, so we need a reverse geocoding tool to map coordinates to locations.
- We started investigating suitable geocoding techniques that we can use to eliminate the trips with coordinates outside the NYC boundaries.
- Collected additional datasets that we thought will be useful in the analysis process; including shapefiles for NYC, dataset for Subway entrances, and a dataset mapping zip codes to neighbourhoods.

Status Report and Preliminary Issues

Some of the milestones achieved at this stage of the project include:

- Implementing and applying the a tool to eliminate points beyond the NYC borough's boundaries. The tool eliminated **2.54% rows of the yellow taxi dataset, 0.272% of the green taxi dataset, and 2.71% of the uber dataset**.
- Applying map/reduce tasks to parse and generate intermediate/aggregated files at which the bulk of the analysis will be applied at.
- Implemented a codebase for calculating the distance between each pickup location and the nearest subway station, using a dataset of the Subway entrances mentioned above, indexed with KDtree.
- Updated the existing tool to also map the coordinates and zip codes, from the dataset and shapefiles, to the affiliate neighbourhood names using the zip codes mapping dataset.

Remaining tasks:

- Aggregating and grouping the datasets by various time and location clusters.
- Applying regression analysis models to capture future trends and growth/decline expectations of each taxi service.
- Visualizing and analyzing the results using graph charts and maps.

Project Report

1. Abstract

This project aims at analysing the trends of the public taxi services in New York City, and investigating the dynamics of the market shares of each taxi service, including yellow cabs, green cabs and Uber. As a new yet very strong player in the taxi industry, Uber is changing the previous norms of the industry at a rapid phase, so comparing Uber's adoption rates in various neighborhoods to the localized use of more traditional yellow and green taxis should be of great interest. The project will try to analyse the trips data of the three public transportation means as a function of time and location, with a balance of fine/coarse grain views depending on the observed patterns. We will also run several regression models to capture the growth patterns throughout various locations and weekly cycles.

2. Datasets

The datasets used in the project include the trip records of the biggest groups of taxi operators in NYC, including the yellow cabs, green cabs[1] and Uber services[2]. The records for the green/yellow taxi trips span a time period of 2009-2015. The Uber dataset that is available for public use, however, include April-September 2014, and January-June 2015. Moreover, the 2015 Uber dataset doesn't have the latitude/longitude details of the pickup locations, rather a pointer to the Uber zone affiliate with that location.

To be able to make concrete observations about the taxi usage patterns across all the taxi services, we have to limit the analysis to taxi service datasets that span the same timeframe as all the others. The limitations we covered at the beginning of the section force us to limit the analysis to the datasets covering the periods of April-September 2014.

The tables below present the statistics of the various used datasets:

	Yellow cabs	Green cabs	Uber
April	14,618,761	1,309,157	564,516
May	14,774,043	1,421,505	652,436
June	13,813,031	1,337,761	663,845
July	13,106,367	1,273,975	796,122
August	12,688,879	1,344,943	829,276
September	13,374,018	1,361,895	1,028,137
TOTAL	**82,375,099**	**8,049,236**	**4,534,332**

[1] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
[2] https://github.com/fivethirtyeight/uber-tlc-foil-response

Before running the bulk of the analysis/modelling, we investigated the data quality and suitability for the intended analysis, and run the following data cleansing processes.

2.1. Eliminating trips outside of NYC boroughs

The provided datasets from the TLC websites are intended to cover the NYC boroughs only, especially for the pickup locations. However, and due to various human and technical errors, the dataset included a minority of trips that had pickup locations outside the borders of the city's boroughs. Including these points in our analysis would result in bias in the produced results, depending on how these points end up aggregated at.

To be able to differentiate the points within NYC from the others, we developed a small tool that builds a KDtree-based index over all NYC boroughs, using a shapefile of the NYC zipcodes and boroughs. The shapefile contains the zip codes, neighbourhoods and boroghs affiliated with all latitude/longitude ranges in the city. Figure shows these neighbourhoods as presented at the file.

After building the KDtree index, the tool iterates through the entire datasets to eliminate trips that don't fall within these borders. The Table below shows the percentage of eliminated records for each file of those included in the dataset. The percentage of eliminated records, averaging about 2.5%, don't pose any severe reduction in the data size, and can be considered negligible to some extent.

Taxi service	Percentage eliminated
Yellow cabs	2.54%
Green cabs	0.272%
Uber	2.71%

2.2. Missing neighbourhood data

The files that we got through the TLC website; whether for the yellow cab, green cab or Uber, provide the location information in terms of longitude/latitude only. To be able to provide a more coherent coarse-grained analysis of the data, we need to have a more coarse-grained clustering of the locations, preferably in terms of neighbourhoods or cluster of blocks.

The most obvious solution for this issue would be to turn again to the NYC shapefile to extract the neighbourhood information. However, as explained at the previous section, the shapefile for

NYC contains the zip codes and borough information only. Using the zip codes alone would be too fine-grain for our analysis, where each zip code covers a few blocks only. Using the borough information would be too coarse-grained! So we still need some additional information to map the zip codes to neighbourhood information.

We were able to complement the shapefile data using an additional dictionary that maps the zip codes to the relevant neighbourhood information[3].

Zip codes	Neighbourhood
10453,10457,10460,...	Central Bronx
10458,10467,10468,...	Bronx Park and Fordham
10451,10452,10456,...	High Bridge and Morrisania
...	...

We included this mapping as an extension to the locations cleaning tool mentioned before. The tool eventually cleans the dataset, and appends the neighbourhood for each record in the file.

2.3. Calculating distance to the nearest subway station

As part of our breakdown of the analysis required for the taxi trips, we thought that the distance to the nearest subway station might have a great influence over the pattern of taxi/Uber usage. For certain distances from the subway stations, using a cab or an Uber might be more suitable for the passengers, especially at certain times of the day, like night hours, or days of the week, like the weekend when the subway routes become more sparse with delayed times between trains. We can use the location information from the data files to calculate the distances to the surrounding subway stations. The exact entrance locations (longitude/latitude) for all the subway stations in NYC can be found at the NYC Open Data [4].

We created a KDtree index over all subway entrance locations, and used the index to query the closest subway entrances using the K Nearest Neighbours algorithm, setting K value to 1 to get the top nearest entrance only.

We have also integrated the codebase to calculate the distance to the nearest subway station with the previous tool, so the same tool now can eliminate the erroneous records, adds the neighbourhood that the pickup location belongs to, and finally calculates the distance to the nearest subway station. The KDtree index is calculated to all the subway stations in the system, rather than for each individual neighbourhood, so even if the nearest subway station is assigned to a different neighbourhood, the distance will be calculated based on it.

---

[3] https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoods.htm
[4] https://nycopendata.socrata.com/Transportation/Subway-Entrances/drex-xx56

3. Parsing and Aggregating the Data in Hadoop MapReduce

For all yellow and green cab trips in the six-month period, the location tool was used to determine the neighborhood and distance to nearest subway station information for both sets of longitude and latitude coordinates available in the data: the coordinates of the pickup, and the coordinates of the dropoff. For the Uber data, because dropoff information was not included in the available data, the location tool was used on the geospatial coordinates of the pickup only. A chain of two successive Map-Reduce processes was used to parse, clean and aggregate the data, with the output of the first Reduce task provided as input to the second Mapper task.

3.1. Joining the pickup, dropoff, and original taxi data

The first MapReduce process joined the corresponding trip records from the pickup and dropoff records output by the location tool for the yellow and green cab data, and projected the relevant attributes of each record into the desired order and format. For the uber data, no dropoff data existed to be joined, so the Mapper task merely projected the relevant attributes in the same format as used for the taxi data, and the Reducer output these projections with no processing. The Mapper task determined the type of taxi for each record as well as whether each record came from the original, pickup, or dropoff relation by searching for these identifier substrings in the input filename. All blank records were skipped and headers were removed.

The Mapper task parsed the date-time string for the pickup time and dropoff time for each record to extract a number from 0-23 corresponding to the hour of the event, a number from 0-6 corresponding to the weekday of the event (from 0: Sunday to 6: Saturday), and a number from 0-53 for the week number of the calendar year 2014. The original date-time strings were removed from the projection and replaced with these values. Additional information, including trip distance, fare amount, total amount, and number of passengers, were retained in the projected record after appropriate rounding. The Reduce task joined this trip information with the neighborhood and distance to subway from both the pickup and dropoff relations for all records in the original relation, regardless of whether pickup and/or dropoff information was missing (due to pickups or dropoffs outside of the region of interest defined by the location tool shapefile). Where these values were missing, they were replaced by empty strings to ensure all records for a particular taxi type had an equal number of attributes.

The input files provided to this Mapper task were chunked by month of the year and type of taxi (yellow, green, or Uber). One Reducer task was used for each month/taxi type combination, for a total of 18 Reducers. The key for each record output by the mapper contained the month and taxi type, and this value was provided to the custom partitioner to ensure, for example, all records from green cabs in April, whether from the original data, the pickup data, or the dropoff data, were assigned to the same Reducer, and the output of each Reducer contained an entire month of joined data for one of the three taxi types. These intermediate data files were saved to an S3 bucket to enable further processing and eventual chaining of MapReduce tasks.

3.2. Aggregating the joined taxi data by weekday, hour, and neighborhood

The joined records output by the first MapReduce tasks were then provided as inputs to the Mappers for the second process. Keys were defined on each record by the hour of the day, the day of the week, the pickup neighborhood (for example, LaGuardia airport), and the type of cab. A count of 1 pickup for each record was output by the mapper, as well as each record's values of distance traveled, fare amount, total amount, number of passengers, and both the pickup location's distance to the subway and the dropoff neighborhood's distance to the subway. For the uber trips, keys were output by the Mapper in the same format, but only the pickup location's distance to the subway was output as all other values were missing. Because aggregation was done over neighborhoods of the pickup point, if the pickup neighborhood was unknown (either invalid coordinates or coordinates outside of NYC), the record was ignored by the Mapper.

A custom partitioner ensured that all keys for a single type of cab went to a single Reducer, and a total of 3 Reducers were used. A custom combiner aggregated the counts and summed the values of each attribute for all records output by a single Mapper. The Reducer functioned quite similarly to the combiner, but after summing all values for all attributes, attributes other than count of pickups were averaged together, according to the total count of pickups in this bin.

However, because missing values of individual attributes, such as fare amount, were summed as 0, while still incrementing the total count of pickups in a particular bucket, these averages were biased. In particular, because the green cabs contained far more missing data compared to the yellow cabs, average values for green cabs tended to be more skewed towards zero compared to the relatively complete records used for the yellow cabs. Because of this discrepancy, caution was taken in using these skewed averages as potential predictors in the subsequent regression tasks.

4. Analyzing the Aggregated Count Data with Regressions

The output of the chained MapReduce task yielded, for each type of taxi, a spatiotemporal bucket of taxi pickups defined by the hour-long interval (e.g. 10:00am - 10:59 am), the day of the week, and the neighborhood of the pickup. Below, we summarize these results with the total pickup count across all neighborhoods by aggregating by hour of the day, and day of the week.

Day of the week	***Yellow cabs***	***Green cabs***	***Uber***
Sunday	10,602,423	1,237,063	470,988
Monday	10,307,896	919,565	525,777
Tuesday	11,699,072	992,684	648,668
Wednesday	11,702,437	1,023,029	681,205
Thursday	11,978,915	1,097,963	737,303
Friday	12,073,152	1,278,011	722,123

Saturday	11,948,311	1,473,385	625,254
total	80,312,206	8,021,700	4,411,318

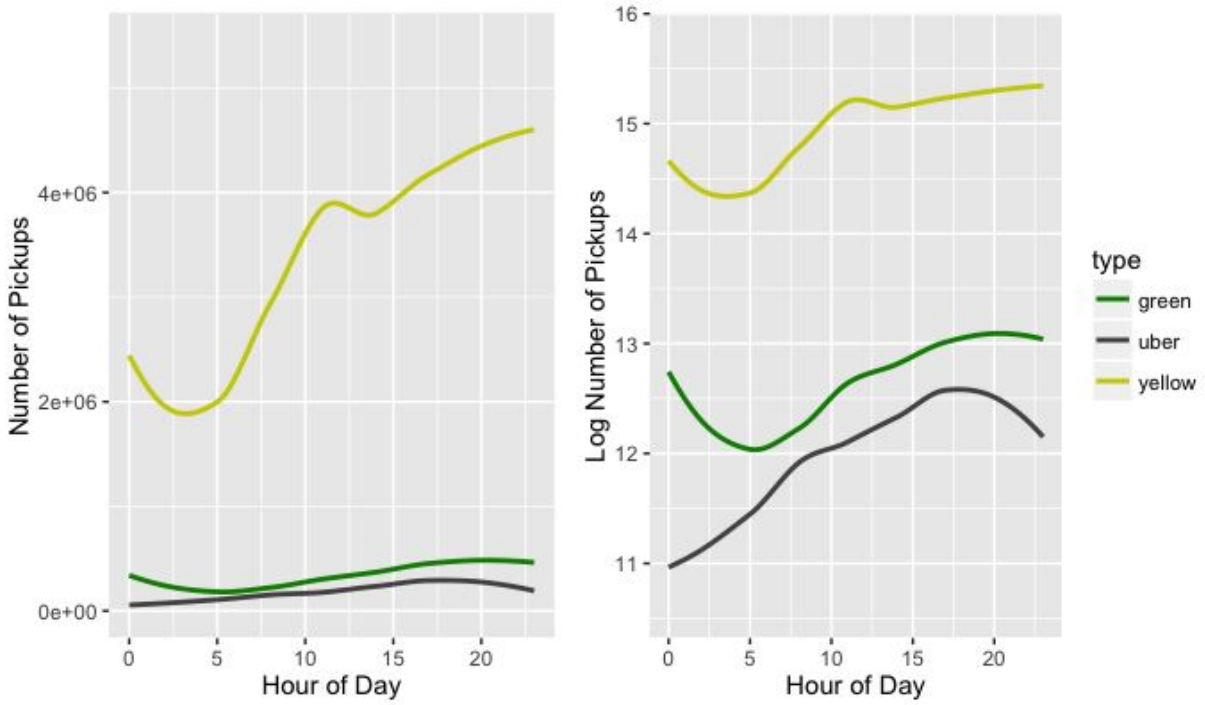


Figure 1: Number of pickups per hour of the day, on standard scale (left) and log scale (right)

Because the counts of pickups in these spatiotemporal buckets is a Poisson-distributed random variable, we chose to fit a generalized linear regression with Poisson link to see if we could predict the expected number of pickups for a particular neighborhood in a one-hour interval, conditional on the type of taxi, the time of day, the day of the week, and potentially other predictors including the average distance to the subway for trips originating in that bucket. In addition to the 52 neighborhood categories, we computed which of the 5 boroughs each neighborhood belonged to. Hour of the day was transformed into two cyclical variables using the sine and cosine transformations. We first performed three simple Poisson regression on each of the taxi trip types using the following variables as predictors of number of pickups.

Yellow and **Green** pickup count models	Day of the week, borough of pickup, sine of hour, cosine of hour, average number of passengers, average trip distance, average distance to subway at pickup point, average distance to subway at dropoff point, indicator for weekends
Uber pickup count models	Day of the week, borough of pickup, sine of hour, cosine of hour, indicator for weekends

4.1. Stepwise Poisson regressions

A stepwise regression using AIC as the feature selection criterion retained all variables in the model, and a likelihood ratio test showed that the average distance to subway at both the pickup point and the dropoff point (for green and yellow cabs only) were significant in predicting the expected number of pickups for a given spatiotemporal bucket.

Yellow	**Model 1:** count ~ weekday + borough + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	**P-value (models compared)**
	Model 2: count ~ weekday + borough + avg.pickup.dist.subway + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	p < 2.2e-16 ***
	Model 3: count ~ weekday + borough + avg.pickup.dist.subway + avg.dropoff.dist.subway + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	p < 2.2e-16 ***
Green	**Model 1:** count ~ weekday + borough + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	**P-value (models compared)**
	Model 2: count ~ weekday + borough + avg.pickup.dist.subway + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	p < 2.2e-16 ***
	Model 3: count ~ weekday + borough + avg.pickup.dist.subway + avg.dropoff.dist.subway + hour.sin + hour.cos + avg.num.passengers + avg.distance.traveled + is.weekend	p < 2.2e-16 ***
Uber	**Model 1:** count ~ weekday + borough + hour.sin + hour.cos + is.weekend	**P-value (models compared)**
	Model 2: count ~ weekday + borough + avg.pickup.dist.subway + hour.sin + hour.cos + is.weekend	p < 2.2e-16 ***

4.2. Cross-validated Lasso penalized Poisson regressions

With this information in mind, we then decided to perform a more robust analysis of the Poisson regression model using cross-validation and residual analysis to identify unusual buckets where the number of pickups observed was drastically higher or lower than expected given the main effects of neighborhood, borough, average distance to subway, day of the week, hour of the day, and type of taxi. This technique allows us to identify spatiotemporal cells where the confluence of these effects reveals a surprising interaction. For example, we shall see that LaGuardia airport is unusually busy at 9am on Mondays, much more so than we would expect from the typical taxi activity at 9am, and the typical taxi activity on Mondays, and the typical taxi activity at LaGuardia airport. Identifying these anomalies allows us to suggest “hot spots” and “dead zones” for every hour of the day-week cycle, and dispatch a larger number of taxis to wait nearby in these unusually busy neighborhoods.

Rather than fitting three separate Poisson regressions as we did in the stepwise procedure, we now include as an additional predictor the taxi type variable, and interact this predictor with all other predictors included in the model. The predictors included in the model are as follows:

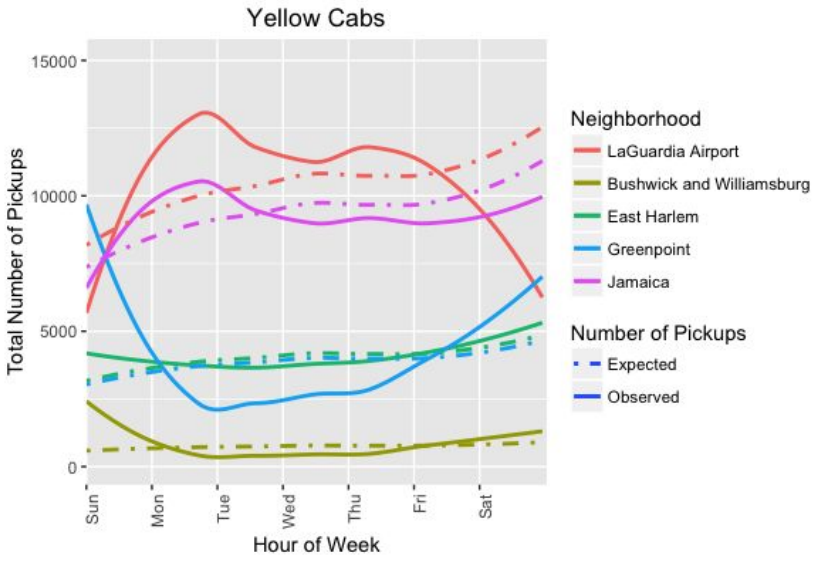
- Day of the week (7 levels, 6 dummy predictors)
- Neighborhood (52 levels, 51 dummy predictors)
- Taxi type (3 levels, 2 dummy predictors
- Weekend indicator (2 levels, 1 dummy predictor)
- Borough (5 levels, 4 dummy predictors)
- Average distance to subway at pickup location (continuous)
- Hour of the day - Sine transformation (continuous)
- Hour of the day - Cosine transformation (continuous)
- Interaction terms (65 predictors * 2 levels, 130 interaction terms)

We use 10-fold cross-validation, and search over 100 values of the regularization penalty term, lambda, using an L1 lasso penalty to induce sparsity in the regression coefficients. We select the value of lambda that minimizes the average cross-validated error across all 10 folds, using the deviance statistic of the generalized linear regression fit as our error measure. We then use the optimal model selected by this procedure to predict expected counts for all spatiotemporal buckets in the data given the set of predictors and interaction terms. Including interaction terms and dummy variables, we include a set of 197 potential predictors in the scope of the model. In the final model, 157 of these predictors have non-zero coefficients. Finally, we compare these expected counts to our observed counts using the Pearson residual for the Poisson distribution, which is simply the observed minus the expected counts, divided by the standard deviation of the expected counts.

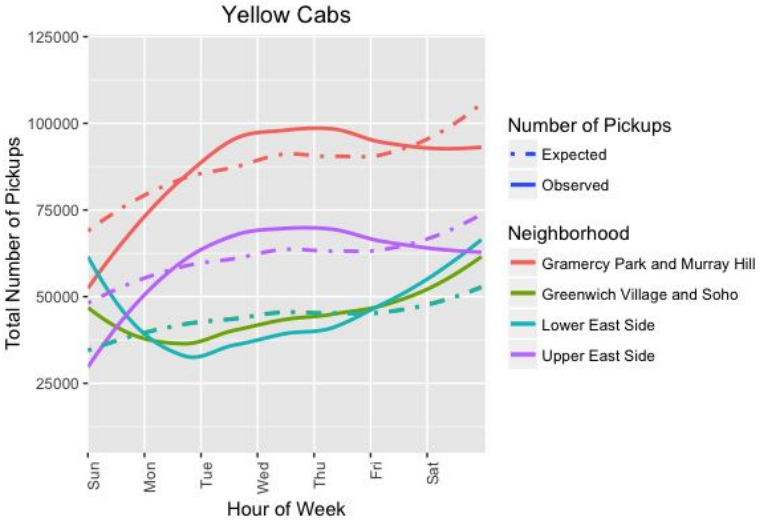
For each of the three types of taxis, we sort these residuals and examine the highest 0.5 percent and lowest 0.5 percent of all spatiotemporal buckets for that taxi type, in order to find the most unusual interactions of neighborhood and time in terms of their observed number of pickups. This method provides very good outlier detection for identifying the “hot spots” (large positive standardized residuals) and “dead zones” (large negative standardized residuals). For example, we see that for yellow cabs at LaGuardia Airport, the most unexpectedly busy times are 9-11am on Mondays and 10pm-midnight on Sundays. This reflects the fact that a large number of business travelers tend to travel Monday mornings, and we could not have recovered this information simply from knowing the average number of taxi pickups at LaGuardia, and the average number of pickups on Mondays, and the average number of pickups at 9am. Below, we summarize a select number of the most unexpectedly busy neighborhoods and times.

Neighborhood	Time	Day	Type	Observed	Expected
Bushwick and Williamsburg	Midnight - 3 am	Sun	green	12am: 4497, 1am: 4803, 2am: 4482	12am: 1001, 1am: 889, 2am: 796
Gramercy Park	8am	Tue, Wed	yellow	Tue: 134368, Wed: 134459	Tue: 66320, Wed: 66333
Gramercy Park	5pm	Tue, Wed	Uber	Tue: 12791, Wed: 12581	Tue: 6350, Wed: 6668
Greenpoint	11pm - 3am	Fri - Sat	green	11pm: 18279, 12am: 19749, 1am: 20337, 2am: 19620	11pm: 8531, 12am: 6348, 1am: 5593, 2am: 4925
Upper East Side	6am - 9am	Mon, Tue, Wed	Uber	Tue 6am: 3769, Tue 7am: 5502, Tue 8am: 4547	Tue 6am: 1055, Tue 7am: 1141, Tue 8am: 1276

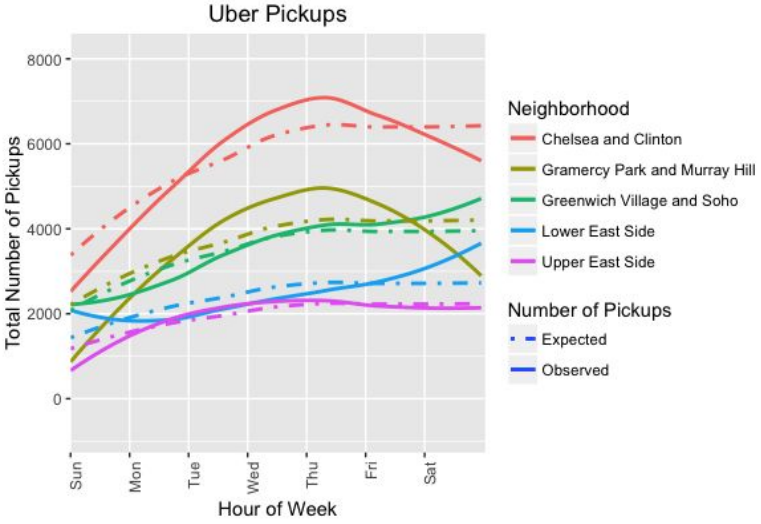
Below we present some interesting examples of neighborhoods where the observed distribution of number of pickups deviates from the expected pattern for certain points in time.



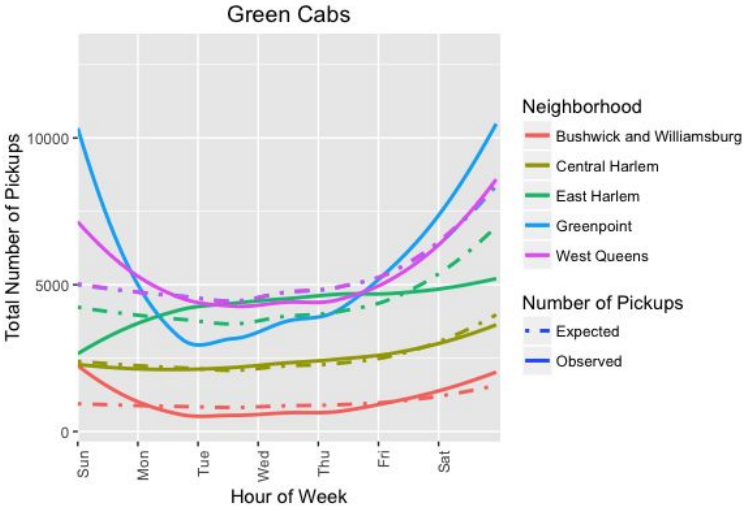
In this plot, we see a huge peak for LaGuardia airport throughout the day on Monday and continuing into Tuesday morning, as arriving travelers hail cabs from the airport to their destination. These times are typically relatively slow for many neighborhoods, so the large number of pickups is unexpected. In contrast, for Greenpoint, we see that very few yellow cabs are taken during the week, and we expect this number of pickups to stay relatively steady, but in fact, this area is incredibly active on Friday and Saturday nights, with a large unexpected boost in the number of taxi pickups beginning late Friday night and continuing into the early hours of Sunday morning.



In contrast, for two of these four neighborhoods in Manhattan, we expect a reasonably high number of pickups on the weekends, but the observed number of pickups is considerably lower than expected, with a large peak in the number of taxi pickups during the weekdays. Thus, we see that the Gramercy Park and Upper East Side neighborhoods seem to be areas where people work during the week, while the Lower East Side and Greenwich Village/Soho neighborhoods are areas where people tend to go out and take taxis more on weekend evenings more so than during the week.



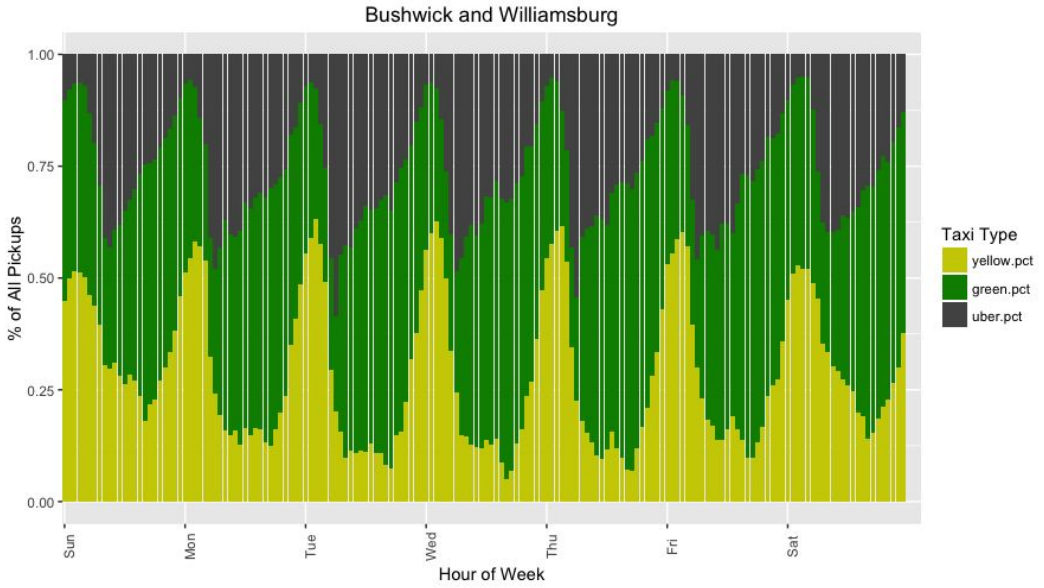
We can observe similar trends in the Uber pickup data, although the overall number of pickups by Uber is much smaller than the number of yellow cab pickups for nearly all neighborhoods. In particular, notice that once again there are more Uber pickups than expected during weekday rush hours in Chelsea and Gramercy Park, and more Uber pickups on weekends in the Lower East Side and Greenwich Village.



Selected outliers in the green cab data show these trends as well. By comparing neighborhoods in these figures, we can easily identify neighborhoods which are weekend nightlife hotspots, and which neighborhoods are more often frequent by working professionals commuting during peak rush hours of the week. The majority of neighborhoods do not demonstrate

such severe discrepancies, but rather have a distribution of pickup frequency that is predicted quite well by our model throughout the week.

We can also use this tool to visualize, for any given neighborhood, the relative distribution of taxi pickups between the three services over the course of the entire week. Typically, yellow cab pickups dominate, but the size of these ratios shift during the course of the week. For example, in Bushwick and Williamsburg, green cab pickups tend to comprise a much larger proportion of the total pickups during the late afternoons and early evenings, and are relatively much less available during the early morning hours.



In East Harlem, by contrast, these dips in yellow taxi availability relative to green cabs seem to occur closer to midday, with increasing yellow cab availability in the early to late evenings. This may be due to a shortage in green cab availability during the early evenings, and if this is the case, dispatchers of green cab companies or Uber may be interested in increasing the availability of green cabs during these times to draw potential business.