Taxi Final Report

Christoph, Chris, Carol

4/30/2017

Introduction

This project explores yellow taxi cab data in NYC. The full dataset online includes all trips completed in yellow taxis in NYC from January to June 2015, and records 18 variables described below in the “Data” section. This dataset is huge (146,087,462 rows and 5GB), and too big for our computers to work with, so we decided to load in data for 1 day where payment type was credit card because these were the only rides that recorded tips. We chose May 6, for no reason in particular, except that it is a non-holiday weekday. This one day has 288k observtions, which was still burdonsome to work with so we took a random subset of 10% of the rows. This left us with 28,857 rows.

From NYC Open Data (https://data.cityofnewyork.us/Transportation/2015-Yellow-Taxi-Trip-Data/ba8s-jw6u (https://data.cityofnewyork.us/Transportation/2015-Yellow-Taxi-Trip-Data/ba8s-jw6u)): “This dataset includes trip records from all trips completed in yellow taxis in NYC from January to June in 2015. Records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. The data used in the attached datasets were collected and provided to the NYC Taxi and Limousine Commission (TLC) by technology providers authorized under the Taxicab Passenger Enhancement Program (TPEP). The trip data was not created by the TLC, and TLC makes no representations as to the accuracy of these data.”

We were particularly interested in exploring 2 features of this taxi trips dataset: tipping behavior and ride availability at different times and locations. First, we will explore what factors predict how much (as a percentage of their total bill) a rider tips. It is standard to tip 20%, which we see in the data, but we wanted to explore exceptions to this norm. We will use a wide variety of variables to try to predict tip percentage. This knowledge could be very helpful to NYC taxi drivers willing to change their driving behavior to get bigger tips. Next, we will examine the most popular pickup and drop off locations in New York and at what times they are most popular, as well as in which direction taxis travel at different times of day. This knowledge would be useful to the taxi cab companies, or even city planners, informing them were more taxis or more public transportation options are needed.

Data

Each row of the dataset corresponds to a taxi ride in NYC on May 5, 2015. Each ride has 18 associated variables: `dropoff_datetime` : date and time when meter was disengaged; `dropoff_latitude` : latitude where the meter was disengaged; `dropoff_longitude` : longitude where the meter was disengaged; `extra` : Miscellaneous extras and surcharges (0 = no charge, 0.5 = 50 cent rush hour charge, 1 = 1 dollar overnight charge); `fare_amount` : time-and-distance fare calculated by the meter; `imp_surcharge` : 30 cent improvement charge on all rides; `mta_tax` : 50 cent tax on all rides; `passenger_count` : number of passengers in the vehicle; `payment_type` : how the passenger paid for the trip. Always 1 (credit card) in the subset of the data we are analyzing; `pickup_datetime` : date and time when meter was engaged; `pickup_latitude` : latitude where the meter was engaged; `pickup_longitude` : longitude where the meter was engaged; `rate_code` : final destination (1 = Standard rate, 2 = JFK, 3 = Newark 4 = Nassau or Westchester, 5 = Negotiated Fare, 6 = Group Ride); `store_and_fwd_flag` : whether the trip record was held in vehicle memory before sending to the vendor; `tip_amount` : tip paid. Note: Only calculated for credit card tips; `tolls_amount` : total amount of all tolls paid in

the trip; `total_amount` : total amount charged to passengers (fare + tip + tolls + extra + mta_tax); `trip_distance` : elapsed trip distance in miles; `vendor_id` : technology vendor that provided the record (1 = Creative Mobile Technologies, 2 = VeriFone)

We also created several variables including `avg_speed` , `cent_pickup` , `cent_dropoff` , `dropoff_hour` , `dropoff_region` , `dropoff_time` , `outbound` , `pickup_hour` , `pickup_region` , `pickup_time` , `tip_pct` , and `trip_duration` .

We did a decent amount of cleaning the dataset before we started our analysis, to remove outliers and missing values. Everything can be seen in the Code Appendix, but briefly, we only included longitudes and latitudes actually in NYC and removed rides where pickup_longitude/latitude equaled dropoff_longitude/latitude (meaning the trip didn't go anywhere), where trip_distance and/or trip_duration equaled 0, where fare_amount equaled 0, where passenger_count equaled 0, where tip percents were above 100%, and where average speed was above 50 mph. After all the cleaning, we worked with 26,049 rides and 35 variables.

Analysis

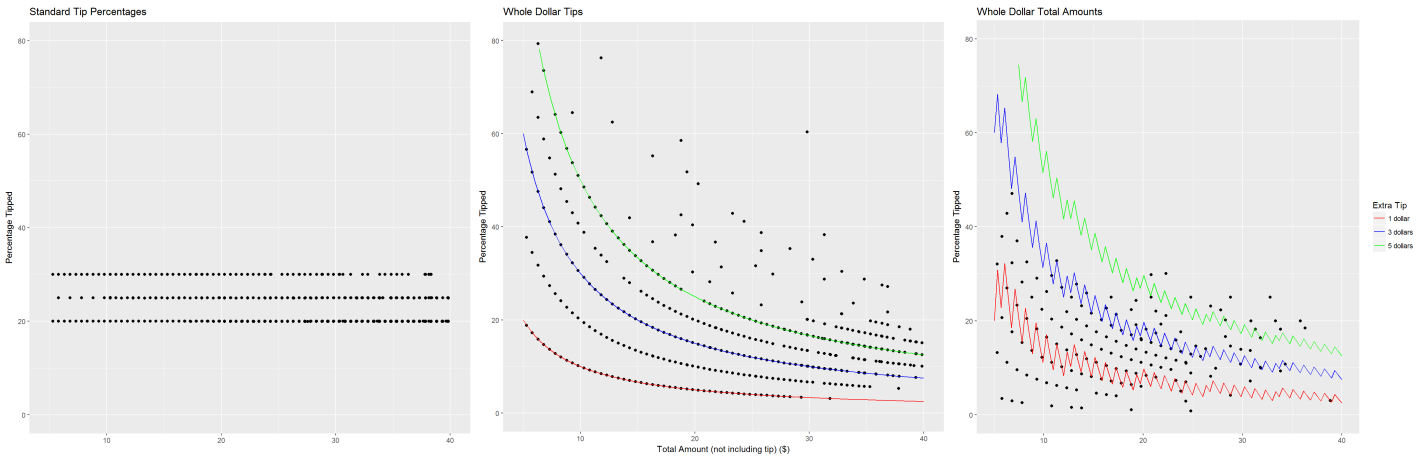
1) What factors predict how much (as a percent of total fare) a person tips?

- Do people tip more, as a percentage, if their fare is lower?

For our first attempt to explain `tip_pct` , we decided to use `fare_amount` as a predictor. We hypothesized that people would give larger tips, in percent terms, the smaller their fare amount since these tips on small amounts are smaller in absolute terms. For example, it would not be uncommon for someone to give a 40% tip on a 5 dollar fare, because that only amounts to 2 dollars, but a 40% tip on a 30 dollar fare (12 dollars) would be less likely. The data do lend support to this hypothesis, as most of the high percentage tips are given on lower bills, and there is a downward curving pattern to the data points, and the coefficient of `fare_amount` is negative, though small (-0.062), and highly signifcant. More interestingly, however, are other patterns that appear in the data.

First, you can see the ubiquitousness of standard tip amounts in the horizantal line of points at 20%, 25%, and 30%. 35% of riders tip approximately 20%, while another 8% tip 25% and 3% tip 30%.

Moreover, in this plot we also see interesting, downward sloping, convex lines of points. These are due to people giving whole dollar tips or giving a tip so that their total bill is a whole number. 23% of riders tip a whole dollar amount while 4% have a whole number total amount, most likely due to a "rounding" tip.



The plot on the right shows just the riders who tipped 20%, 25%, or 30%, and the two other plots show the riders who tipped a whole dollar amount or used their tip to round their total bill to a whole number. The overlaid lines represent the expected tip amount based on the fare if people were to tip a whole dollar amount or use a

rounding tip, and then add 1, 3, or 5 dollars to their tip. As can be seen in the plot they fit the points perfectly, and we can fit the other curves of points by adding other constant extras like 2, 4, 6, 7, 8… dollars.

Since these 3 tipping behaviors (standard percent, whole dollar tip, whole dollar bill) are so common, we decided to run a multinomial logistic regression to predict when people tip using one of these

other	round_tip	standard_pct	whole_tip
7500	704	11589	5845

As fare, average speed, passenger count and duration increases, people are more likely to use the standard tip percentage, then a whole tip, then a rounding tip and then some other behavoir. As tip amount or percent increases, tipping a standard amount is more common, while giving a rounding tip or a whole dollar tip are less common than some other behavoir. In terms of time of day, people are most likely to give a whole or rounding tip at night, and are more likely to give a standard percent in the afternoon or evening.

For the further analysis, we decided to remove these observations because the riders tipping these salient amounts are most likely choosing tips regardless of the other variables, and our goal is to explore how these other factors affect `tip_pct`. We can already explain this tipping behavior, and in fact have a plots that fit the points perfectly. Therefore, it wll be more interesting to focus our analysis on the points we haven't yet explained. Note that we also removed 0% tips, about 3% of the original data, for the same reason, and because these riders could have tipped in cash, which we cannot determine from the data we have.

Now we can continue with our analysis of `tip_pct`.

After removing these observations, the coefficient on `fare_amount` drops from -0.062 to -0.310, and stays highly significant, even with fewer observations. So we now expect a 1 dollar increase in fare to be associated with a 0.3% smaller tip. Moreover, the R^2 of this new model jumps from 0.5% to 6.8%.

When running this model, we assume the LINE assumptions, which we will now check.

Linearity - the plot appears to have some curvature suggesting that the variables may not be linearly associated. To control for this possible non-linearity we included a squared term for `fare_amount`. In this model, the coefficient drops further to -1.003, the coefficient on `fare_amount` squared is 0.022 and significant, and the R^2 rises further to 10.9%. In context, this means that tips, percent wise, are highest at low fares, drop slightly as the fare increases for a while, and then rise back up. **Indendence** - the obersvations are independent because each point is a different ride. **Normality** - to check for this we plotted a histogram of the residuals from the model. The residuals seem somewhat normally distributed with a few large outliers. We tried running a regression using the log transformed values for `tip_pct` and `fare_amount` just in case. The p-value was again very significant and the coefficient was negative, but the R^2 dropped back down to 4.3% **Equal Variance** - The plot of the residuals against the fitted values shows that we may be slightly underpredicting for lower values of y.

While, we have some concerns about these assumptions, overall the model fits them well enough to continue with our analysis. Addionally, our p values were extremely signifigant and most likely wouldn't be altered by further calculations.

- Do people tip more if the taxi drives more quickly?

Moving on, we decided to test if people would tip more the higher the average taxi speed, because people generally don't like waiting in traffic, so if they get to where they are going more quickly, relative to how far they are traveling, we hypothesized that they would give a higher tip to thank the driver. However the model gives a

very small, not very significant coefficient on `avg_speed` and the scatterplot shows almost no correlation. Perhaps some people tip less the faster they go because they feel unsafe manuevering quickly through NYC traffic. If this is true, it could offset our hypothesis and result in this overall lack of correlation.

- Do people tip more if there are more passengers in the car?

Here, our hypothesis was that riders would tip more as the number of passengers increased, due to a social pressure exerted from the other passengers to tip more. However, most riders tip the same (median of 20%) regardless of how many people were in the taxi Further, the regression gives a small, negative coefficient on `passenger_count`, and it is not very significant. Finally, the pairwise t-test shows that no two levels of `passenger_count` are different from one another.

- Do people tip more at different times of day?

Next, we looked at how tip giving bahvior changes throughout the day. In terms of `dropoff_time`, the p-values were not significant, but there is some indidcation that people tip less at night (midnight to 5 AM) than in the morning. This result would make sense because people riding that late are most likely either tired from a long day or work or returning from a night out and thus more likely to get mad at a driver, and tip less (if we assume people are worse at controlling their emotions when they are tired and/or inebriated). Note: We get a similar result for `pickup_time`, and for a combined regression with both `pickup_time` and `drop_off` time.

Note: on all of the above regressions, we checked the LINE assumptions, and concluded that there were satisfied.

- Which variables are most important for predicting `tip_pct`?

Finally, we ran an all-subsets regression only incuding a subset of the variables. We eliminated some predictors (ie: `vendor_id`, `store_and_fwd_flag`, `rate_code`) because the rider did not know them and they could not have affected `tip_pct`, others (ie: `payment_type`, `mta_tax`, `imp_surcharge`) because they were the same for every ride, others because they were obviously correlated with other predictors or `tip_pct` itself and would have violated the independence assumption (ie: `tip_amount` with `tip_pct`, `trip_duration` with `trip_distance`, `total_amount` with `fare_amount`, and `pickup_time`, `pickup_hour`, `pickup_datetime`, `dropoff_hour`, `dropoff_datetime` with `dropoff_time`). That leaves 9 predictors: `extra`, `fare_amount`, `passenger_count`, `tolls_amount`, `trip_distance`, `avg_speed`, `dropoff_time`, and `outbound`.

The best model includes `extra`, `fare_amount`, `tolls_amount`, `trip_distance`, and `outbound`, as well as the Afternoon and Evening levels of `dropoff_time`. Running this “best” regression, we get the following results:

`extra` has the largest effect on `tip_pct`, as the data suggest that a rider would tip 1.9% less if charged 1 dollar more in extra fees. This makes sense because people generally don’t want to pay as large of an additional tip on top of the extra amount they’re already being charged. `fare_amount` has a similar effect on `tip_pct`, most likely for the same reason, but interestingly, a $1 increase in `tolls_amount` is associated with a 0.7% increase in tip. We believe this is because it’s a hastle for drivers to go through tollbooths and riders feel sympathetic so tip more. Finally, people tip more on longer distance trips, which again makes sense if riders build a rapport with thier driver due to the extended length. These are all the significant predictors from the regression.

However, even in this “best” regression, the R^2 value is still only 8.4%, meaning that we are only explaining 8.4% of the variability in `tip_pct`. This is not bad, but in the end, we conclude that while we did gain some interesting insights, it is very difficult to accurately predict `tip_pct`, partly because there is a lot of variability, and partly because of the ubiquitousness of standard tips like 20% and whole dollar tips.

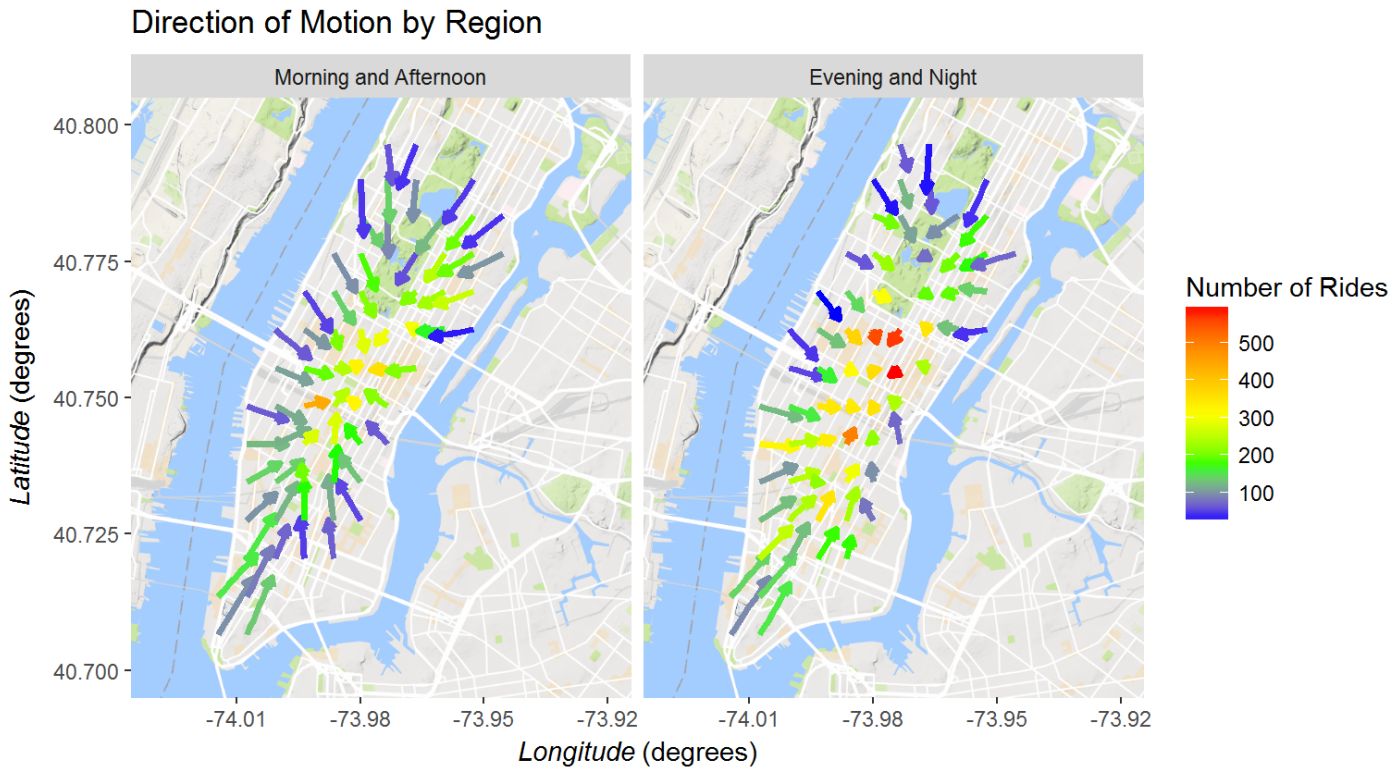
- Finally, we ran one last regression to determine what affects whether or not people tip at all.

All else equal, riders are more likely to tip 0% with a higher fare, when there are more passengers in the taxi, at night, or they are heading outbound, and less likely to tip 0% on longer trips.

2. Direction of Taxi Flow and Public Transit

Taxi rides and subway rides are interchangeable modes of transportation. Routes that taxis most consistently drive may thus indicate routes with sub-optimal public transit connections. In this section, we explore the relationship between public transit routes and taxi rides. We hypothesize that taxis are used as substitutes for public transit rather than compliments i.e. that many taxi rides cover routes also possible through public transit.

To begin this analysis, we first investigate the direction of flow of taxis within Manhattan. To do this, we first divided Manhattan and the surrounding area into 625 regions. Then, we calculated the average normalized direction vector for all taxis starting within each region. The plot below shows these arrows superimposed over a map of Manhattan. Each arrow begins at the midpoint of the region it represents. The longer the arrow, the more rides starting within a region agree on direction. The color of each arrow indicates number of rides starting with in a region. Arrows representing less than 30 rides were removed from the plot.



As can be seen in the plot, the largest number of taxi pickups during both the morning & afternoon (between 4am and 2pm, inclusive) and the evening & night(between 3 pm and 3am, inclusive) occur in central Manhattan, around Times Square/Grand Central Station. The arrows in this region are short, which suggests that rides starting in this region go equally in all directions. On the other hand, taxis starting around the periphery of the city tend to have a fairly uniform inward direction of motion. This makes sense since Manhattan is an island and therefore outward motion along the periphery is not possible (assuming taxis stay in Manhattan).

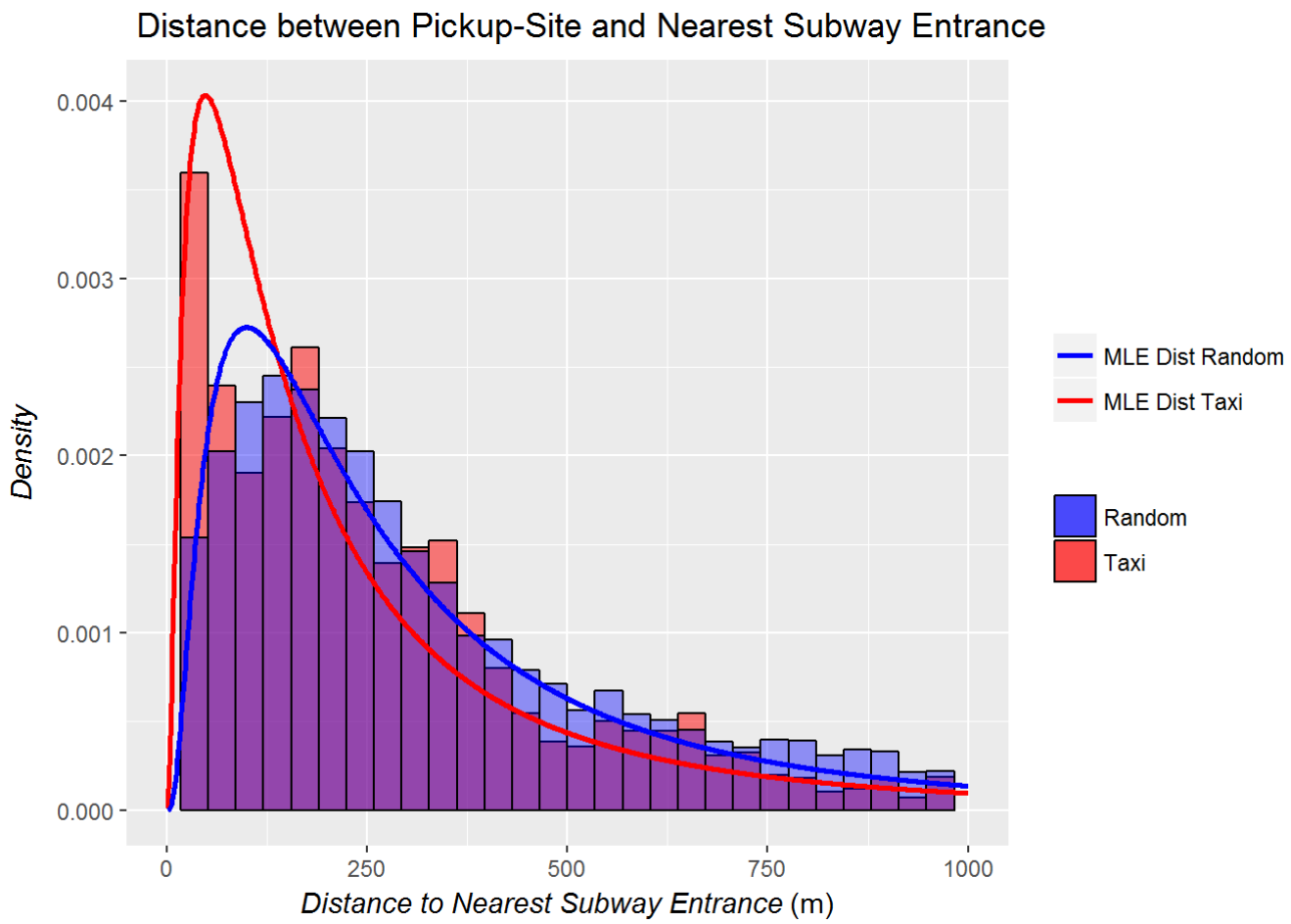
When comparing the above plot to a subway map of NYC (see below), it appears that regions around the periphery with higher number of pickups are those with worse Subway connections to central Manhattan. For instance, in both at both times of day a large number of people taxi from the southwest edge of Manhattan

toward central Manhattan (as shown by the green-yellow arrows on the plot above). This may be explained by the poor subway connection along this route.

To further investigate the relationship between taxi rides and public transit we investigated how far the pickup locations of the taxi rides were from subway stations. To do this, we downloaded the GPS coordinates of all subway entrances in NCY from New York State government website (source: https://data.ny.gov/Transportation/NYC-Transit-Subway-Entrance-And-Exit-Data/i9wp-a4ja/data (https://data.ny.gov/Transportation/NYC-Transit-Subway-Entrance-And-Exit-Data/i9wp-a4ja/data)). For taxi ride, we then calculated the straight-line distance between pickup location and the nearest subway station.

The plot below shows a histogram of straight-line distances between pickup site and nearest subway station (red). Shown in blue is the sampling distribution for randomly generated coordinates in Manhattan. In the Taxi histogram, there was originally a cluster of data points at distance 2.5 km, however we removed these points for this analysis because we assumed they corresponded to pickup locations outside of Manhattan (our region of interest).

If taxi pickup locations were independent of station locations, we would expect the red and the blue histograms be from the same distribution. To test whether the two samples are from the same distribution, we fitted the maximum-likelihood estimated (MLE) lognormal distributions to each sample (solid lines in plot below). Lognormal distributions were chosen since the histograms show clear right-skew and distance must always be positive. We then used a likelihood ratio test to test whether the two samples are from the same distribution. First, we calculated the likelihood ratio using $\Lambda = 2(\log L(a) + \log L(b) - L(a \cup b))$, where $L(a)$ is the likelihood of the MLE distribution for taxi rides, $L(b)$ is the likelihood of the MLE distribution for random positions, and $L(a \cup b)$ is the likelihood of the MLE distribution of both samples combined. We then calculated the probability of the two sample coming from the same distribution by integrating the chi squared distribution with two degrees of freedom (since two parameters were estimated per distribution) from Λ to ∞. The p-value is $5 \cdot 10^{-268}$, which is below the significance level of 5% used in this analysis. This confirms that there are significant differences between the two distributions.



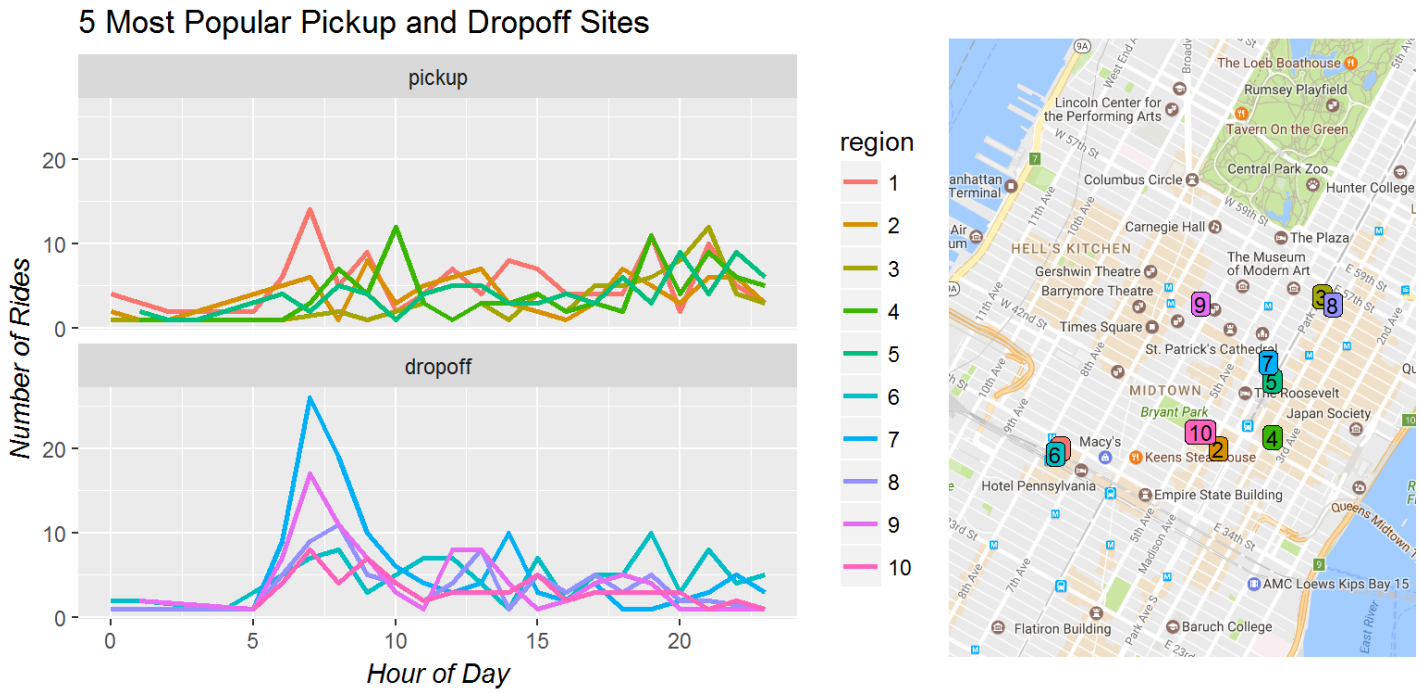
From the above plot, the histogram for `taxi` differs from the `random` one primarily at low values of distance, where there are more taxi pickups than would be expected from random positioning. At first, this seems counterintuitive as it suggests that people are more likely to call a taxi when closer to a subway station. One possible explanation for this, however, is that people use taxis to supplement the last leg of a journey that cannot be done by subway. If this were the case, we would expect rides beginning near subways to be shorter than rides beginning far away.

To test this, we split that data into two groups: taxi closer than 100 m to a subway station and taxi rides farther than 100 m to a subway station. We then performed a t-test on the 'trip_distance' of these two groups. The resulting p-value is 0.75, suggesting that there is no difference between the two means; however, histograms of `trip_distance` for each group suggest that the normality assumption is violated. To fix this, we did a t-test on log(`trip_distance`), realizing that we are now testing for differences in geometric means rather than arithmetic means. The p-value of 0.08 still suggests that, however, the differences in the means are not significant (at the 5% significance level).

The t-test shows that there is no significant difference in ride length for people closer to subways. We therefore reject the hypothesis that people use taxi rides to complement subway travels. Instead, it seems more likely that people gravitate toward subway stops when looking for taxis.

What are the most popular taxi pickup and dropoff sites?

The fact that so many taxi pickups occur right next to subway stops suggests that people do not view the subway as an alternative to taking the taxi. This alludes to potential deficiencies in the subway system. Subway deficiencies are likely to be greatest in regions with the most taxi pickup and dropoffs. As a final part our project, we thus investigate which pickup/dropoff sites within NYC are the most popular.



The map above (right) shows the most popular pickup and dropoff regions in Manhattan. Regions 1-5 are the most popular taxi pickup sites (1 = most popular, 5 = least) and regions 6-10 are the most popular dropoff sites (6 = most popular, 10 = least popular). Both the most popular pickup and dropoff site was Penn Station (region 1,6). This suggests that Penn Station is the location in NYC where public transportation is most deficient relative to travel demand. Thus, if a new subway line were to be built, it should probably connect with Penn Station. Also included in the most popular pickup/dropoff sites are is Grand Central Station (regions 2,4, 10) and Park Avenue, near Trup Tower (regions 7,5). All of the most popular pickup and dropoff regions are located in central Manhattan, suggesting that this is the area where improved public transit is most important.

The left part of the figure above shows how the number of pickups/drop-offs changes at each location with time of day. While the number of pickups/dropoffs stays fairly constant throughout the day, there appear to be slightly more pickups and droppoffs in the mornings (around 7am) and evenings (around 8pm)/While speculative, this may be due people commuting to work. There are slightly more pickups and droppoffs in the mornings (around 7am) and evenings (around 8pm), which may be from people commuting to work. This is particularly noticeable at for dropoffs around 7am. These are the times where subways need to be running most frequently.

Conlclusion

The taxi industry is shifting dramatically with companies such as Uber and Lyft starting to dominate the transportation industry. We could've really benefited from incorporating data from these companies into our final analysis. Further exploration should compare the different destinations, passenger counts, and distances that Lyft and Uber take their passengers as compared to taxis. It is important to note that Lyft and Uber don't allow passengers to tip, so comparing the total fare amount is important in this case, especially in trying to predict total earnings of drivers. Our project explores both tipping tendencies and direction of taxi flow and public transit.

The reliance on tipping separates taxicabs from other transportation industries. Our analysis is useful for taxi drivers in determining the best scenario for earning a higher tip. First, generally, in terms of tip percentage, we've found most people tip the standard 20%, 35% of riders tip approximately 20%, while another 8% tip 25% and 3% tip 30%. However, there are also many people who tip whole dollar amounts or use their tips to round

their total to a whole dollar amount: 23% of riders tip a whole dollar amount while 4% have a whole number total amount, most likely due to a “rounding” tip. The frequency of these behaviors is, however, altered by different variables.

People do not appear more to tip more if there are more passengers in the car, however, although the p-values were not significant, there was some indication that people tip less at night (midnight to 5 AM) than in the morning. Finally, we found that riders are more likely to tip 0% with a higher fare, at night or they are heading outbound, and less likely to tip 0% on longer trips. Although our LINE assumptions were satisfied to a certain degree, perhaps being able to utilize the full dataset would’ve allowed us to more fully satisfy these assumptions.

In order to continue to explore factors affecting tip, further analysis should compare different days of the year. For example, we might predict that people would tip more around the holiday times, or during the summer months where there are possibly more tourists.

In the second part of our analysis we investigated the relationship between public transportation and taxi rides. To do this, we added a dataset `subway` containing the the GPS coordinates of all subway entrances in NYC. Using this dataset, we found that taxi ride pickup locations are significantly closer to subway stations than would be expected if they were randomly distributed throughout Manhattan. Furthermore, the mean distances traveled are the same for taxis starting near subway stations and those starting farther from subway stations. People thus do now view the subway as a substitute for taxi travel, suggesting deficiencies in the subway system. By analyzing the most popular pickup and dropoff locations in Manhattan, we found that these subway deficiencies are greatest in Central Manhattan, particularly near Penn Station and Grand Central. New subway lines being constructed should ideally connect these regions.

To strengthen these conclusions, the actual routes traveled by taxis could be analyzed in more detail. For example, the taxi travel time for a given route could have been compared to the estimated time travel time using public transportation. It could then be tested whether taxi rides occur more frequently along routes with greater time difference the two modes of transportation. Similarly, the number of stopovers required to complete a taxi route with public transportation could be investigated in more detail. We expect routes with more stopovers to be frequented more by taxis. This analysis was beyond the scope of the experiment, however, because such time estimates for public transit are not readily available.

Finally, if we had to conduct an experiment testing the relationship between public transit and taxi rides, we survey taxi customers on their choices with questions like (1) What routes do they most frequently travel by taxi? (2) Why do they not they not take the subway. Such a survey would provide meaningful insight into consumer behavior and help focus the analysis of the taxi data.

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(fitdistrplus)
library(GGally)
library(ggplot2)
library(ggmap)
library(grid)
library(gridExtra)
library(knitr)
library(leaps)
library(lubridate)
library(mapproj)
library(MASS)
library(nnet)
library(pander)
library(readr)
ggpairs(taxi[,c(7, 16, 17, 20, 22, 35)])
# library("RSocrata")
# taxi <- read.socrata("https://data.cityofnewyork.us/resource/2yzn-sicd.json?$where=pickup_date
time between '2015-05-06T00:00:00.000' and '2015-05-07T00:00:00.000'&payment_type=1")
#write.csv(taxi, "taxi_data_full.csv", row.names = FALSE)
setwd("~/Yale/Semester 4/STAT 230 Intro Data Analysis/NCAA Final Project_GitHub/ncaa-academics")
taxi <- read.csv("taxi_data.csv", as.is = TRUE)
###CLEANING

# Create `imp_surcharge`
taxi$imp_surcharge <- rep(0.3, nrow(taxi))
###CLEANING

# Remove 2 observations where `fare_amount` == 0
taxi <- taxi[taxi$fare_amount > 0,]
# Create `tip_pct`
taxi <- taxi %>% mutate(tip_pct = (tip_amount / (total_amount - tip_amount)) * 100)
# Remove observation where tip_pct > 100 or = 0
taxi <- taxi[taxi$tip_pct <= 100,]
# Create `pickup_hour` and `dropoff_hour`
taxi$pickup_hour <- as.numeric(substr(taxi$pickup_datetime, 12, 13))
taxi$dropoff_hour <- as.numeric(substr(taxi$dropoff_datetime, 12, 13))

# Create `pickup_time` and `dropoff_time`
taxi$pickup_time <- factor(ifelse(taxi$pickup_hour %in% 6:11, "Morning",
                    ifelse(taxi$pickup_hour %in% 12:17, "Afternoon",
                    ifelse(taxi$pickup_hour %in% 18:23, "Evening",
                    ifelse(taxi$pickup_hour %in% 0:5, "Night", NA)))),
                    levels = c("Morning", "Afternoon", "Evening", "Night"))
taxi$dropoff_time <- factor(ifelse(taxi$dropoff_hour %in% 6:11, "Morning",
                     ifelse(taxi$dropoff_hour %in% 12:17, "Afternoon",
                     ifelse(taxi$dropoff_hour %in% 18:23, "Evening",
                     ifelse(taxi$dropoff_hour %in% 0:5, "Night", NA)))),
                     levels = c("Morning", "Afternoon", "Evening", "Night"))
# Only consider longitudes and latitudes in Manhatten
MAXLAT <- 40.85; MINLAT <- 40.675; MAXLONG <- -73.85; MINLONG <- -74.1
taxi <- taxi %>% filter(pickup_longitude < MAXLONG, pickup_longitude > MINLONG,
```

```
               pickup_latitude < MAXLAT, pickup_latitude > MINLAT)
taxi <- taxi %>% filter(dropoff_longitude < MAXLONG, dropoff_longitude > MINLONG,
               dropoff_latitude < MAXLAT, dropoff_latitude > MINLAT)
# Remove rows where dropoff = pickup
taxi <- taxi %>% filter(dropoff_longitude != pickup_longitude,
              dropoff_latitude != pickup_latitude)
# Create geographic regions
taxi$dropoff_lat_region <- cut(taxi$dropoff_latitude, 3)
taxi$dropoff_long_region <- cut(taxi$dropoff_longitude, 3)
taxi$pickup_lat_region <- cut(taxi$pickup_latitude, 3)
taxi$pickup_long_region <- cut(taxi$pickup_longitude, 3)

taxi$pickup_region <- as.factor(3 * as.numeric(taxi$pickup_lat_region) +
                                  as.numeric(taxi$pickup_long_region) - 3)
taxi$dropoff_region <- as.factor(3 * as.numeric(taxi$dropoff_lat_region) +
                                   as.numeric(taxi$dropoff_long_region) - 3)
# Create outbound, cent_pickup, and cent_dropoff
norm <- function(x) {
  sqrt(rowSums(x^2))
}
# Coordinates of Bryant Park taken as Downtown Manhattan
cent_lat <- 40.753889; cent_long <- -73.983490
cent_pickup <- data.frame(x = rep(cent_lat,nrow(taxi)) - taxi$pickup_latitude,
                          y = rep(cent_long, nrow(taxi)) - taxi$pickup_longitude)
cent_dropoff <- data.frame(x = rep(cent_lat,nrow(taxi)) - taxi$dropoff_latitude,
                           y = rep(cent_long, nrow(taxi)) - taxi$dropoff_longitude)
taxi$cent_pickup <- norm(cent_pickup)
taxi$cent_dropoff <- norm(cent_dropoff)
taxi$outbound <- as.numeric(norm(cent_dropoff) > norm(cent_pickup))
# Remove `trip_distance` = 0
taxi <- taxi[taxi$trip_distance > 0,]
# Remove `passenger_count` = 0
taxi <- taxi[taxi$passenger_count > 0,]
# Create `trip_duration`
taxi$pickup_datetime <- ymd_hms(taxi$pickup_datetime)
taxi$dropoff_datetime <- ymd_hms(taxi$dropoff_datetime)
taxi$trip_duration <- as.numeric((taxi$dropoff_datetime - taxi$pickup_datetime) / 60)
taxi <- taxi[taxi$trip_duration > 0,]
# Create `avg_speed`
taxi$avg_speed <- taxi$trip_distance / (taxi$trip_duration / 60)
# Remove 0% tip
taxi0 <- taxi
taxi <- taxi[taxi$tip_pct > 0,]

# Produces heat plot of NYC showing average of desired variable for each region.
  # data: data frame from which data is drawn
  # variable: vector of the variable under consideration
  # lat_type: either "pickup" or "dropoff"
  # GRID_RESOLUTION: integer representing number of divisions of longitude/latitude
  # title: string containing plot title
  # varname: string containing variable name
  # MINLAT, MAXLAT, MINLONG, MAXLONG: specify area covered by plot

locationPlot <- function(data, variable, lat_type = "pickup", GRID_RESOLUTION = 100,
```

```
                        title = "Add Title", varname = "Variable", MINLAT = 40.675,
                        MAXLAT = 40.85, MINLONG = -74.1, MAXLONG = -73.85) {

  data$variable <- variable

  if(lat_type == "pickup") {
    # Select all taxi rides whose pickup was within specified region. Stores
    # these rides to a new data frame 'pos'
    pos <- filter(data, pickup_longitude < MAXLONG, pickup_longitude > MINLONG,
                  pickup_latitude < MAXLAT, pickup_latitude > MINLAT)
    # Remove all rows with dropoff longitude or latitude equal to 0
    pos <- filter(pos, dropoff_longitude != 0, dropoff_latitude!= 0)
    # Remove all rows where pickup and dropoff locations are the same
    pos <- filter(pos, dropoff_longitude != pickup_longitude,
                  dropoff_latitude != pickup_latitude)
    # Cut latitude into bins
    pos$lat_region <- cut(pos$pickup_latitude,GRID_RESOLUTION)
    # Cut longitude into bins
    pos$long_region <- cut(pos$pickup_longitude,GRID_RESOLUTION)
  }

  else if (lat_type == "dropoff") {
    pos <- filter(data, dropoff_longitude < MAXLONG, dropoff_longitude > MINLONG,
                  dropoff_latitude < MAXLAT, dropoff_latitude > MINLAT)
    pos <- filter(pos, dropoff_longitude != 0, dropoff_latitude!= 0)
    pos <- filter(pos, dropoff_longitude != pickup_longitude, dropoff_latitude != pickup_latitud
e)
    pos$lat_region <- cut(pos$dropoff_latitude,GRID_RESOLUTION)
    pos$long_region <- cut(pos$dropoff_longitude,GRID_RESOLUTION)
  }

  # Groups data in 'pos' by 'lat_region' and 'long_region' (region on the grid)
  # Calculates number of rides being picked up in each region ('count') and the
  # average direction traveled for each taxi rides beginning in each grid
  # (long_dir_avg, lat_dir_avg)
  grouped <- group_by(pos, lat_region, long_region) %>%
    summarize(count = n(), var = mean(variable))

  temp <-  grouped$lat_region
  temp <- gsub("\\(", "", temp)
  temp <- gsub("\\]", "", temp)
  temp <- strsplit(temp, ",")
  temp <- unlist(temp)
  temp <- as.numeric(temp)
  grouped$max_lat <- temp[2*1:nrow(grouped)] # list of latitude upper bounds
  grouped$min_lat <- temp[2*1:nrow(grouped)-1] # listof latitude lower bounds
  grouped$avg_lat <- 0.5*(grouped$max_lat + grouped$min_lat)

  temp <-  grouped$long_region
  temp <- gsub("\\(", "", temp)
  temp <- gsub("\\]", "", temp)
  temp <- strsplit(temp, ",")
  temp <- unlist(temp)
  temp <- as.numeric(temp)
```

```
  grouped$max_long <- temp[2*1:nrow(grouped)] # list of longitude upper bounds
  grouped$min_long <- temp[2*1:nrow(grouped)-1] # listof longitude lower bounds
  grouped$avg_long <- 0.5*(grouped$max_long + grouped$min_long)

  # Load Google Map of desired region
  map <- get_map(location = c(MINLONG,MINLAT, MAXLONG, MAXLAT), source = "stamen",
                 maptype = "watercolor")

  ggmap(map) +
    geom_rect(aes(x = min_long, y = max_long, xmin = min_long, xmax = max_long,
                  ymin = min_lat, ymax = max_lat, fill = var), data = grouped) +
    scale_fill_gradientn(varname, colors = c("purple","blue","green","yellow","orange","red")) +

    labs(title = title, x = expression(italic("Longitude") ~ "(degrees)"), y =
           expression(italic("Latitude") ~ "(degrees)"))
}

####1) What factors predict how much (as a percentage of their total fare) a person tips?
ggplot(subset(taxi, total_amount < 50), aes(x = total_amount - tip_amount, y = tip_pct)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Percentage Tipped by Total Amount",
       x = "Total Amount (not including tip) ($)", y = "Percentage Tipped")
mfare <- lm(tip_pct ~ fare_amount, data = taxi)
summary(mfare)
standard_tip <- taxi[round(taxi$tip_pct, 1) %in% c(20.0, 25.0, 30.0),]
standard_pct_plot <- ggplot(subset(standard_tip, total_amount < 50),
                            aes(x = total_amount - tip_amount, y = tip_pct)) +
  geom_point() +
  scale_x_continuous(limits = c(5, 40)) +
  scale_y_continuous(limits = c(0, 80)) +
  labs(title = "Standard Tip Percentages", y = "Percentage Tipped", x = "")
whole_tip <- taxi[taxi$tip_amount %% 1 == 0,]
whole_num1 <- function(x) {(((x + 1)/x) - 1) * 100}
whole_num3 <- function(x) {(((x + 3)/x) - 1) * 100}
whole_num5 <- function(x) {(((x + 5)/x) - 1) * 100}
whole_tip_plot <- ggplot(whole_tip, aes(total_amount - tip_amount, tip_pct)) +
  geom_point() +
  stat_function(fun = whole_num1, aes(color = "1 dollar")) +
  stat_function(fun = whole_num3, aes(color = "3 dollars")) +
  stat_function(fun = whole_num5, aes(color = "5 dollars")) +
  scale_color_manual(name = "", values = c("red", "blue", "green")) +
  theme(legend.position = "none") +
  scale_x_continuous(limits = c(5, 40)) +
  scale_y_continuous(limits = c(0, 80)) +
  labs(title = "Whole Dollar Tips",
       x = "Total Amount (not including tip) ($)", y = "Percentage Tipped")
whole_bill <- taxi[taxi$total_amount %% 1 == 0,]
whole_num1 <- function(x) {((ceiling(x) + 1)/x - 1) * 100}
whole_num3 <- function(x) {((ceiling(x) + 3)/x - 1) * 100}
whole_num5 <- function(x) {((ceiling(x) + 5)/x - 1) * 100}
whole_bill_plot <- ggplot(subset(whole_bill, total_amount < 50), aes(x = total_amount - tip_amou
nt, y = tip_pct)) +   geom_point() +
  stat_function(fun = whole_num1, aes(color = "1 dollar")) +
```

```
  stat_function(fun = whole_num3, aes(color = "3 dollars")) +
  stat_function(fun = whole_num5, aes(color = "5 dollars"))  +
  scale_color_manual(name = "Extra Tip", values = c("red", "blue", "green")) +
  scale_x_continuous(limits = c(5, 40)) +
  scale_y_continuous(limits = c(0, 80)) +
  labs(title = "Whole Dollar Total Amounts", y = "Percentage Tipped", x = "")
grid.arrange(standard_pct_plot, whole_tip_plot, whole_bill_plot, ncol = 3)
taxi$const_tip_pct <- rep(NA, nrow(taxi))
taxi$const_tip_pct <- as.numeric(round(taxi$tip_pct, 1) %in% c(20.0, 25.0, 30.0))

taxi$const_tip <- rep(NA, nrow(taxi))
taxi$const_tip <- as.numeric(taxi$tip_amount %% 1 == 0)

taxi$round_tip <- rep(NA, nrow(taxi))
taxi$round_tip <- as.numeric(taxi$total_amount %% 1 == 0)
table("Standard Pct" = taxi$const_tip_pct, "Whole Tip" = taxi$const_tip)
table("Standard Pct" = taxi$const_tip_pct, "Round Tip" = taxi$round_tip)
table("Whole Tip" = taxi$const_tip, "Round Tip" = taxi$round_tip)
taxi$tiptype <- rep(NA, nrow(taxi))
taxi$tiptype[(taxi$const_tip == 1 & taxi$round_tip == 0 & taxi$const_tip_pct == 0)] <- "whole_ti
p"
taxi$tiptype[(taxi$const_tip == 0 & taxi$round_tip == 0 & taxi$const_tip_pct == 1)] <- "standard
_pct"
taxi$tiptype[(taxi$const_tip == 0 & taxi$round_tip == 1 & taxi$const_tip_pct == 0)] <- "round_ti
p"
taxi$tiptype[(taxi$const_tip == 0 & taxi$round_tip == 0 & taxi$const_tip_pct == 0)] <- "other"
pander(table(taxi$tiptype))
mlog_none <- multinom(tiptype ~ 1, data = taxi)

mlog_fare <- multinom(tiptype ~ fare_amount, data = taxi)
summary(mlog_fare)
anova(mlog_fare, mlog_none)

mlog_duration <- multinom(tiptype ~ trip_duration, data = taxi)
summary(mlog_duration)
anova(mlog_duration, mlog_none)

mlog_tipamount <- multinom(tiptype ~ tip_amount, data = taxi)
summary(mlog_tipamount)
anova(mlog_tipamount, mlog_none)

mlog_tippct <- multinom(tiptype ~ tip_pct, data = taxi)
summary(mlog_tippct)
anova(mlog_tippct, mlog_none)

mlog_passenger <- multinom(tiptype ~ passenger_count, data = taxi)
summary(mlog_passenger)
anova(mlog_passenger, mlog_none)

mlog_avgspeed <- multinom(tiptype ~ avg_speed, data = taxi)
summary(mlog_avgspeed)
anova(mlog_avgspeed, mlog_none)

mlog_dropofftime <- multinom(tiptype ~ dropoff_time, data = taxi)
```

```
summary(mlog_dropofftime)
anova(mlog_dropofftime, mlog_none)
# Remove standard tip percentages
taxi2 <- taxi
taxi2 <- taxi2[round(taxi2$tip_pct) != 20,]
taxi2 <- taxi2[round(taxi2$tip_pct) != 25,]
taxi2 <- taxi2[round(taxi2$tip_pct) != 30,]

# Remove whole dollar tips
taxi3 <- taxi2[taxi2$tip_amount %% 1 != 0,]

# Remove rounding tips
taxi4 <- taxi3[taxi3$total_amount %% 1 != 0,]
ggplot(subset(taxi4, total_amount < 50), aes(x = total_amount - tip_amount, y = tip_pct)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Percentage Tipped by Total Amount",
       x = "Total Amount (not including tip) ($)", y = "Percentage Tipped")

mfare2 <- lm(tip_pct ~ fare_amount, data = taxi4)
summary(mfare2)
mfare3 <- lm(tip_pct ~ fare_amount + I(fare_amount^2), data = taxi4)
summary(mfare3)
hist(mfare3$residuals)
mfare4 <- lm(log(tip_pct) ~ log(fare_amount), data = taxi4)
summary(mfare4)
plot(resid(mfare3) ~ fitted(mfare3), pch=16)
before <- ggplot(subset(taxi, total_amount < 50), aes(x = total_amount - tip_amount, y =
tip_pct)) +
  geom_point() +
  geom_point(data = taxi[taxi$tip_amount %% 1 == 0,], color = "red") +
  geom_point(data = taxi[round(taxi$tip_pct) %in% c(0, 20, 25, 30),], color = "green") +
  labs(title = "Percentage Tipped by Total Amount",
       x = "Total Amount (not including tip) ($)", y = "Percentage Tipped")
after <- ggplot(subset(taxi4, total_amount < 50), aes(x = total_amount - tip_amount, y =
tip_pct)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Percentage Tipped by Total Amount (Without Standard Tips)",
       x = "Total Amount (not including tip) ($)", y = "Percentage Tipped")
grid.arrange(before, after, ncol = 2)
ggplot(subset(taxi4, tip_pct <= 50), aes(x = avg_speed, y = tip_pct)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Tip % by Avg Speed", x = "Average Speed (mph)",
       y = "Percentage Tipped")

mspeed <- lm(tip_pct ~ avg_speed, data = taxi4)
summary(mspeed)
hist(mspeed$residuals)
plot(resid(mspeed) ~ fitted(mspeed), pch=16)
ggplot(taxi4, aes(x = factor(passenger_count), y = tip_pct)) +
  geom_boxplot() +
  labs(title = "Tip % by # of Passengers", x = "Number of Passengers",
```

```
       y = "Percentage Tipped")
mpassengers <- lm(tip_pct ~ passenger_count, data = taxi4)
summary(mpassengers)
anova(mpassengers)
pairwise.t.test(taxi$tip_pct, taxi$passenger_count)

mpassengers <- lm(tip_pct ~ passenger_count + I(passenger_count^2), data = taxi4)
summary(mpassengers)
anova(mpassengers)
hist(mpassengers$residuals)
plot(resid(mpassengers) ~ fitted(mpassengers), pch=16)
ggplot(taxi4[taxi4$tip_pct < 50,], aes(x = dropoff_time, y = tip_pct)) +
  geom_boxplot() +
  labs(title = "Percentage Tipped by Time of Day", x = "Time of Day",
       y = "Percentage Tipped")
ggplot(taxi4, aes(x = pickup_hour, y = tip_pct)) +
  geom_jitter() +
  labs(title = "Percentage Tipped by Time of Day", x = "Time of Day",
       y = "Percentage Tipped")
mdropoff_time <- lm(tip_pct ~ dropoff_time, data = taxi4)
summary(mdropoff_time)
anova(mdropoff_time)
pairwise.t.test(taxi$tip_pct, taxi$dropoff_time)

mpickup_time <- lm(tip_pct ~ pickup_time, data = taxi4)
summary(mpickup_time)
anova(mpickup_time)
pairwise.t.test(taxi$tip_pct, taxi$pickup_time)

mtime <- lm(tip_pct ~ dropoff_time + pickup_time, data = taxi4)
summary(mtime)
anova(mpickup_time)
hist(mtime$residuals)
plot(resid(mtime) ~ fitted(mtime), pch=16)
r1 = regsubsets(tip_pct ~ extra + fare_amount + passenger_count + tolls_amount +
                trip_distance + avg_speed + dropoff_time + outbound, data = taxi4)
r1s <- summary(r1)
bestwhich <- r1s$which[which.min(r1s$cp),]
best_vars <- names(bestwhich[bestwhich == TRUE])
best_vars <- best_vars[-1]
mbest <- lm(tip_pct ~ extra + fare_amount + tolls_amount + trip_distance +
            outbound + dropoff_time, data = taxi3)
summary(mbest)
taxi0$tip0 <- factor(ifelse(round(taxi0$tip_pct) == 0, "Yes", "No"))
m0all <- glm(tip0 ~ extra + fare_amount + passenger_count + tolls_amount +
             trip_distance + avg_speed + dropoff_time + outbound, data = taxi0,
             family = binomial)
m0step <- step(m0all, direction = "backward", trace = FALSE)
summary(m0step)


rm(grouped)
MAXLAT <- 40.85
MINLAT <- 40.675
```

```
MAXLONG <- -73.85
MINLONG <- -74.1
SIZEFACTOR  <-  0.015 # Desired Magnitude of Direction Arrows
GRID_RESOLUTION <- 25



taxi$lat_region <- cut(taxi$pickup_latitude,GRID_RESOLUTION)
taxi$long_region <- cut(taxi$pickup_longitude,GRID_RESOLUTION)
# Add rows 'lat_dir' and 'long_dir' which respectively contain the change in latitude and longit
ude for each ride.
taxi <- mutate(taxi, lat_dir = (dropoff_latitude - pickup_latitude),
               long_dir = (dropoff_longitude - pickup_longitude))

# Create columns 'lat_dir_scaled' and 'long_dir_scaled', which contain 'lat_dir' and 'long_dir'
 normalized to magnitude 'SIZEFACTOR'
taxi<- mutate(taxi, lat_dir_scaled = lat_dir * SIZEFACTOR/sqrt(lat_dir^2 + long_dir^2),
               long_dir_scaled = long_dir * SIZEFACTOR /sqrt(lat_dir^2 + long_dir^2))

# Note: NaN results for rides where pickup and dropoff locations are equal.

taxi$daytime <- rep(NA, nrow(taxi))

taxi$daytime <- ifelse(taxi$pickup_hour<=14 & taxi$pickup_hour >= 4 , "Morning and Afternoon",
"Evening and Night")
taxi$daytime <- factor(taxi$daytime, levels = c("Morning and Afternoon", "Evening and Night"))
# Group data by 'lat_region' and 'long_region'. Calculates number of rides being picked up ('cou
nt') and the average direction traveled for each region
grouped <- group_by(taxi, lat_region, long_region, daytime) %>%
  summarize(lat_dir_avg = mean(lat_dir_scaled, na.rm = TRUE),
            long_dir_avg = mean(long_dir_scaled, na.rm =TRUE), count = n())

# Split latitude region boundaries into upper, lower, and average longitudes
temp <-  grouped$lat_region
temp <- gsub("\\(", "", temp)
temp <- gsub("\\]", "", temp)
temp <- strsplit(temp, ",")
temp <- unlist(temp)
temp <- as.numeric(temp)
grouped$max_lat <- temp[2*1:nrow(grouped)]
grouped$min_lat <- temp[2*1:nrow(grouped)-1]
grouped$avg_lat <- 0.5*(grouped$max_lat + grouped$min_lat)

# Split longitude region boundaries into upper, lower, and average longitudes
temp <-  grouped$long_region
temp <- gsub("\\(", "", temp)
temp <- gsub("\\]", "", temp)
temp <- strsplit(temp, ",")
temp <- unlist(temp)
temp <- as.numeric(temp)
grouped$max_long <- temp[2*1:nrow(grouped)]
grouped$min_long <- temp[2*1:nrow(grouped)-1]
grouped$avg_long <- 0.5*(grouped$max_long + grouped$min_long)
# creates heat plot with colors reprenting number of pickups in the region
```

```
heatplot <- ggplot(data = grouped, aes(xmin = min_long, xmax = max_long, ymin = min_lat, ymax =
max_lat, fill = count)) +
  geom_rect() +
  scale_fill_gradientn("Number of Pickups", colors = c("purple","blue","green","yellow", "orang
e", "red")) +
  facet_wrap(~daytime)
MAXLAT1 <- 40.8
MINLAT1 <- 40.7
MAXLONG1 <- -73.92
MINLONG1<- -74.03
MINRIDES <-  30 # minimum number of rides for arrow to show up


#loads in google map of Manhattan
map <- get_googlemap(center = c(lon = cent_long, lat = cent_lat), zoom = 12, maptype =
"terrain", style = 'feature:all|element:labels|visibility:off')

#plots direction of motion by region
motion_plot <- ggmap(map) +
  geom_segment(aes(x = (avg_long + long_dir_avg), xend = avg_long, y = (avg_lat + lat_dir_avg),
yend = avg_lat, color = count), data = filter(grouped, count > MINRIDES), arrow = arrow(length=u
nit(0.15, "cm"), ends="first", type = "closed"), size = 1.3) +
  labs(title = "Direction of Motion by Region", x = expression(italic("Longitude")~"(degrees)"),
 y = expression(italic("Latitude")~"(degrees)"))+
  scale_color_gradientn("Number of Rides", colors = c("blue","green","yellow","orange","red")) +
  scale_y_continuous(limits=c(MINLAT1, MAXLAT1)) +
  scale_x_continuous(limits = c(MINLONG1, MAXLONG1)) +
  facet_wrap( ~ daytime)

motion_plot
ggmap(map) +
  geom_segment(aes(x = (avg_long + long_dir_avg), xend = avg_long, y = (avg_lat + lat_dir_avg),
yend = avg_lat, color = daytime), data = filter(grouped, count > MINRIDES), arrow =
arrow(length=unit(0.15, "cm"), ends="first", type = "closed"), size = 0.9) +
  labs(title = "Direction of Motion by Region", x = expression(italic("Longitude")~"(degrees)"),
 y = expression(italic("Latitude")~"(degrees)")) +
  scale_y_continuous(limits=c(MINLAT1, MAXLAT1)) +
  scale_x_continuous(limits = c(MINLONG1, MAXLONG1))
knitr::include_graphics('NYC_Subway_Map.jpg')
# proportion of taxis heading away from downtown
sum(taxi$outbound)/nrow(taxi)
plot(jitter(lat_dir_scaled, factor =1000) ~ jitter(long_dir_scaled, factor = 1000), data = taxi,
 col = rgb(0,0,0, 0.01), pch =16, asp = 1)
points(x = mean(taxi$long_dir_scaled), y = mean(taxi$lat_dir_scaled), col = "red", pch = 16, cex
 = 3)

# calculate direction angle of a vector
toPolar <- function(x,y) {
  angle <- atan(y/x)*180/pi
  if (x>0 & y<0) {
    angle <-  angle + 360}
  if (x<0 & y < 0) {
    angle <-  angle + 180}
  if (x<0 & y > 0) {
```

```
    angle <- angle + 180}

  angle
}

x <-  taxi$long_dir_scaled
y <- taxi$lat_dir_scaled

#cacluclates direction angle of each taxi ride
taxi$angle <- rep(NA,nrow(taxi))
for (i in 1:nrow(taxi)) {
  taxi$angle[i] <- toPolar(x[i],y[i])
}

ggplot(data = taxi, aes(x = angle)) + geom_histogram()



#colnames(taxi)
#plot(taxi[,c(16,17,19,20,31, 32,33,34, 35, 36)])
#Source "https://data.ny.gov/Transportation/NYC-Transit-Subway-Entrance-And-Exit-Data/i9wp-a4ja/
data"
# loads in coordinates of NCY subway entrances
subway <- read_csv("~/Yale/Semester 4/STAT 230 Intro Data Analysis/NCAA Final Project_GitHub/nca
a-academics/NYC_Transit_Subway_Entrance_And_Exit_Data.csv")
subway <- filter(subway, Entry == "YES")
subway <- subway[,c(29,30)]
colnames(subway) <- c("Ent_lat", "Ent_long")

# calculate distance between pickup site and nearest subway entrance for each taxi ride
taxi$station_dist <- rep(NA, nrow(taxi))
for (i in 1:nrow(taxi)) {
taxi$station_dist[i] <- min(norm(data.frame(x = subway[,"Ent_lat"]-rep(taxi$pickup_latitude[i],
nrow(subway)), y = subway[,"Ent_long"]-rep(taxi$pickup_longitude[i], nrow(subway)))))
}

# convert distance from degrees to meters
convert <- 40075/360*1000
taxi$station_dist <- taxi$station_dist*convert




# randomly generate positions in Manhattan and find distance to nearest subway station
set.seed(230)
MINLAT2  <-  40.71
MAXLAT2 <-  40.760
MINLONG2  <-  -74.01
MAXLONG2 <-  -73.975
MINRIDES  <-  30
rand_lat <- runif(10000, min = MINLAT2, max = MAXLAT2)
rand_long <- runif(10000, min = MINLONG2, max = MAXLONG2)
rand_pos <- data.frame(pickup_latitude = rand_lat, pickup_longitude = rand_long)
```

```
rand_pos$station_dist <- rep(NA, nrow(rand_pos))
for (i in 1:nrow(rand_pos)) {
rand_pos$station_dist[i] <- min(norm(data.frame(x = subway[,"Ent_lat"]-rep(rand_pos$pickup_latit
ude[i], nrow(subway)), y = subway[,"Ent_long"]-rep(rand_pos$pickup_longitude[i],
nrow(subway)))))
}
rand_pos$station_dist <- rand_pos$station_dist*convert
# fit maximum likelihood estimated log-normal distributions to the two distance distributions
dist1 <- fitdist(taxi$station_dist[taxi$station_dist < 1500], "lnorm")
mean1 <- summary(dist1)$estimate[1]
sd1 <- summary(dist1)$estimate[2]
dist2 <- fitdist(rand_pos$station_dist, "lnorm")
mean2 <- summary(dist2)$estimate[1]
sd2 <- summary(dist2)$estimate[2]
dist3 <-  fitdist(c(rand_pos$station_dist,taxi$station_dist[taxi$station_dist < 1500]), "lnorm")
mean3 <- summary(dist3)$estimate[1]
sd3 <- summary(dist3)$estimate[2]

L1 <- logLik(dist1)
L2 <- logLik(dist2)
L3 <- logLik(dist3)
lambda <-  2*(L1 +L2 - L3)


f <- function(x) dchisq(x,2)
p_value <- integrate(f, lower = lambda, upper = Inf)
p_value
# histogram of distance distributions with superimposed MLE-estimated distributions
station_hist <- ggplot() +
  geom_histogram(data = taxi, aes(x = station_dist, y =..density.., fill = "Taxi"), alpha  =
0.5, col = "black") +
  geom_histogram(data = rand_pos, aes(x = station_dist, y = ..density..,  fill = "Random"), alph
a = 0.4, col = "black")+
  labs(title = " Distance between Pickup-Site and Nearest Subway Entrance ", x = expression(ital
ic("Distance to Nearest Subway Entrance")~"(m)"), y = expression(italic("Density"))) +
  scale_x_continuous(limits = c(0, 1000)) + geom_line(aes(x = 1:1500, y =dlnorm(1:1500, mean1, s
d1), color = "MLE Dist Taxi"), size = 1) + geom_line(aes(x = 1:1500, y =dlnorm(1:1500, mean2, sd
2), color = "MLE Dist Random"), size = 1) + scale_fill_manual(values = c("Random" = "blue","Tax
i" = "red"))  + theme(legend.title=element_blank()) + scale_color_manual(values = c("MLE Dist Ta
xi" = "red", "MLE Dist Random" = "blue"))
station_hist

#Note several outliers at 2500 m for Taxi
plot(log(taxi$trip_distance) ~ log(taxi$station_dist))

# test assumption for t-test
var(taxi$trip_distance[taxi$station_dist<=100])
var(taxi$trip_distance[taxi$station_dist > 100 & taxi$station_dist <1500 ])
hist(taxi$trip_distance[taxi$station_dist<=100])
hist(taxi$trip_distance[taxi$station_dist > 100 & taxi$station_dist <1500 ])
t.test(taxi$trip_distance[taxi$station_dist<=100], (taxi$trip_distance[taxi$station_dist > 100 &
 taxi$station_dist <1500 ]))
t.test(log(taxi$trip_distance[taxi$station_dist<=100]), log(taxi$trip_distance[taxi$station_dist
 > 100 & taxi$station_dist <1500 ]))
```

```
ggplot() + geom_histogram(data = taxi[taxi$station_dist<=100,], aes(trip_distance, y =
..density..)) + geom_histogram(data = taxi[taxi$station_dist > 100 & taxi$station_dist <1500,],
aes(trip_distance, y = ..density..), fill = "red", alpha = 0.2)

mod <- lm(trip_distance~station_dist, data = taxi[taxi$station_dist < 1500,])
summary(mod)

GRID_RESOLUTION <- 40
N_LOCATIONS <- 5

taxi$lat_region <- cut(taxi$pickup_latitude, GRID_RESOLUTION)
taxi$long_region <- cut(taxi$pickup_longitude, GRID_RESOLUTION)

# Groups data by geographic region and calculates number of rides picked up and the average dire
ction traveled for each ride beginning in each grid
grouped <- group_by(taxi, lat_region, long_region) %>%
  summarize(count = n())

# Create new data frame 'max' containing the regions with the most pickups
max <- grouped[order(grouped$count, decreasing =TRUE),][1:N_LOCATIONS,]
max <- filter(taxi, lat_region == max$lat_region & long_region == max$long_region)

# Groups data by hour, latitude, longitude and count number rides in each group
grouped_pickup <- group_by(max, pickup_hour, lat_region, long_region) %>% summarize(count = n())

# Add column identifying rides as "pickup"
grouped_pickup$type <- rep("pickup", nrow(grouped_pickup))

taxi$lat_region <- cut(taxi$dropoff_latitude, GRID_RESOLUTION)
taxi$long_region <- cut(taxi$dropoff_longitude, GRID_RESOLUTION)

grouped <- group_by(taxi, lat_region, long_region) %>%
  summarize(count = n())

# Create new data frame 'max' containing the regions with the most dropoffs
max <- grouped[order(grouped$count, decreasing =TRUE),][1:N_LOCATIONS,]

# Creates data frame of all taxi rides going to these region
max <- filter(taxi, lat_region == max$lat_region & long_region == max$long_region)

# Groups data by pickup_hour, latitude, and longitude and counts how many rides are in each grou
p
grouped_dropoff <- group_by(max, pickup_hour, lat_region, long_region) %>%
  summarize(count = n())

# Identifies data as "dropoff"
grouped_dropoff$type <- rep("dropoff", nrow(grouped_dropoff))
# Combine data tables grouped_pickup, grouped_dropoff" into dat
dat <- bind_rows(ungroup(grouped_pickup), ungroup(grouped_dropoff))

# Create data frame of latitudes and longitudes, without duplicates
grid <- data.frame(latitude = dat$lat_region, longitude = dat$long_region)
grid <- unique(grid)
```

```
# Classify rides in dat by region
grid$regioncode <- 1:nrow(grid)
dat$region <- rep(NA, nrow(dat))
for(i in 1: nrow(grid)) {
  dat$region[(as.character(dat$lat_region) == as.character(grid$latitude[i])) &
       (as.character(dat$long_region) == as.character(grid$longitude[i]))] <- i
}
dat$region <- factor(dat$region)
dat$type <- factor(dat$type, levels = c("pickup", "dropoff"))
# Split region boundaries into upper, lower, and average latitudes
temp <-  grid$latitude
temp <- gsub("\\(", "", temp)
temp <- gsub("\\]", "", temp)
temp <- strsplit(temp, ",")
temp <- unlist(temp)
temp <- as.numeric(temp)

grid$max_lat <- temp[2 * 1:nrow(grid)]
grid$min_lat <- temp[2 * 1:nrow(grid) - 1]
grid$avg_lat <- 0.5 * (grid$max_lat + grid$min_lat)

# Split region boundaries into upper, lower, and average longitudes
temp <-  grid$longitude
temp <- gsub("\\(", "", temp)
temp <- gsub("\\]", "", temp)
temp <- strsplit(temp, ",")
temp <- unlist(temp)
temp <- as.numeric(temp)

grid$max_long <- temp[2 * 1:nrow(grid)]
grid$min_long <- temp[2 * 1:nrow(grid)-1]
grid$avg_long <- 0.5 * (grid$max_long + grid$min_long)
# create line graph of number of pickup/dropoffs per hour at the 5 most popular sites
timeplot <- ggplot(data = dat, aes(x = pickup_hour, y = count, col = region)) +
  geom_line(size = 1) +
  labs(title = "5 Most Popular Pickup and Dropoff Sites",
       x = expression(italic("Hour of Day")),
       y = expression(italic("Number of Rides"))) +
  facet_wrap(~ type, ncol =1)
  grid$regioncode <- factor(grid$regioncode, levels = c("1", "2", "3", "4", "5", "6", "7", "8",
"9", "10"))
#Plots location of most popular pickup and dropoff sites
MAXLAT1 <- 40.775; MINLAT1 <- 40.74; MAXLONG1 <- -73.965; MINLONG1<- -74.0#73.995
map <- get_map(location = c(MINLONG1 - 0.01 ,MINLAT1 - 0.01, MAXLONG1 + 0.01, MAXLAT1 + 0.01), s
ource = "google", maptype = "roadmap")

pop_map <- ggmap(map) +
  labs(x = expression(italic("Longitude") ~ "(degrees)"),
       y = expression(italic("Latitude") ~ "(degrees)")) +
  geom_label(aes(jitter(avg_long), jitter(avg_lat),label = regioncode, fill = regioncode), data
= grid, size = 3, label.padding = unit(0.1, "lines")) +
  theme(axis.line = element_blank(),
    axis.text.x = element_blank(),
```

```
    axis.text.y = element_blank(),
    axis.ticks = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),plot.margin=unit(c(0,0,0,0),"mm")) +
  scale_y_continuous(limits=c(MINLAT1, MAXLAT1)) +
  scale_x_continuous(limits = c(MINLONG1, MAXLONG1))+
  theme(legend.position="none")
grid.arrange(timeplot, pop_map, ncol = 2, widths = c(2,1))
# add to tips section

data  <-  taxi
variable <- taxi$tip_pct
lat_type <- "pickup"
data$variable <- variable
GRID_RESOLUTION <-  2
title  <-  "Average Tip Percentage by Region"
varname  <-  "Tip Percentage"
MINLAT  <-  40.71
MAXLAT  <-  40.760
MINLONG  <-  -74.01
MAXLONG <-  -73.975
MINRIDES  <-  30


  if(lat_type == "pickup") {
    # Select all taxi rides whose pickup was within specified region. Stores
    # these rides to a new data frame 'pos'
    pos <- filter(data, pickup_longitude < MAXLONG, pickup_longitude > MINLONG,
                  pickup_latitude < MAXLAT, pickup_latitude > MINLAT)
    # Remove all rows with dropoff longitude or latitude equal to 0
    pos <- filter(pos, dropoff_longitude != 0, dropoff_latitude!= 0)
    # Remove all rows where pickup and dropoff locations are the same
    pos <- filter(pos, dropoff_longitude != pickup_longitude,
                  dropoff_latitude != pickup_latitude)
    # Cut latitude into bins
    pos$lat_region <- cut(pos$pickup_latitude,GRID_RESOLUTION)
    # Cut longitude into bins
    pos$long_region <- cut(pos$pickup_longitude,GRID_RESOLUTION)
  }

  if (lat_type == "dropoff") {
    pos <- filter(data, dropoff_longitude < MAXLONG, dropoff_longitude > MINLONG,
                  dropoff_latitude < MAXLAT, dropoff_latitude > MINLAT)
    pos <- filter(pos, dropoff_longitude != 0, dropoff_latitude!= 0)
    pos <- filter(pos, dropoff_longitude != pickup_longitude, dropoff_latitude != pickup_latitud
e)
    pos$lat_region <- cut(pos$dropoff_latitude,GRID_RESOLUTION)
    pos$long_region <- cut(pos$dropoff_longitude,GRID_RESOLUTION)
  }

  # Groups data in 'pos' by 'lat_region' and 'long_region' (region on the grid)
  # Calculates number of rides being picked up in each region ('count') and the
  # average direction traveled for each taxi rides beginning in each grid
  # (long_dir_avg, lat_dir_avg)
```

```
  grouped <- group_by(pos, lat_region, long_region) %>%
    summarize(count = n(), var = mean(variable))

  temp <-  grouped$lat_region
  temp <- gsub("\\(", "", temp)
  temp <- gsub("\\]", "", temp)
  temp <- strsplit(temp, ",")
  temp <- unlist(temp)
  temp <- as.numeric(temp)
  grouped$max_lat <- temp[2*1:nrow(grouped)] # list of latitude upper bounds
  grouped$min_lat <- temp[2*1:nrow(grouped)-1] # listof latitude lower bounds
  grouped$avg_lat <- 0.5*(grouped$max_lat + grouped$min_lat)

  temp <-  grouped$long_region
  temp <- gsub("\\(", "", temp)
  temp <- gsub("\\]", "", temp)
  temp <- strsplit(temp, ",")
  temp <- unlist(temp)
  temp <- as.numeric(temp)
  grouped$max_long <- temp[2*1:nrow(grouped)] # list of longitude upper bounds
  grouped$min_long <- temp[2*1:nrow(grouped)-1] # listof longitude lower bounds
  grouped$avg_long <- 0.5*(grouped$max_long + grouped$min_long)
  grouped$var <- as.factor(ceiling(grouped$var))
   # Load Google Map of desired region
 map <- get_googlemap(center = c(lon = cent_long, lat = cent_lat), zoom = 12, maptype = "terrai
n", style = 'feature:all|element:labels|visibility:off')

  ggmap(map) +
    geom_rect(aes(x = min_long, y = max_long, xmin = min_long, xmax = max_long,
                  ymin = min_lat, ymax = max_lat, fill = var), data = filter(grouped, count > MI
NRIDES), alpha = 0.2) +
    labs(title = title, x = expression(italic("Longitude") ~ "(degrees)"), y =
           expression(italic("Latitude") ~ "(degrees)")) +
    geom_point(aes(x=cent_long, y = cent_lat), colour = "black", size = 3)
```