# Peak Usage of Traditional New York City Taxi Services:

## How taxi drivers can remain competitive by maximizing rides and fare amounts

Siyao Ma
Sudeshna Barman
Dannver Wu

Stats 133
Spring 2016
Professor Adam Lucas
Group Name: Team Taxi

# I. Introduction

The increasing availability of peer-to-peer ridesharing services such as Uber (launched in June 2010) and Lyft (beginning in June 2012) have led many analysts, economists, and journalists to conclude that traditional taxi services will soon go out of business. These experts rightly point out that services like Uber and Lyft are far cheaper than taxis. And because these on-demand services operate through convenient mobile applications, they also save customers the time and hassle of calling or flagging down a taxi. Moreover, traditional taxis face much higher operating costs, such as licensures, certifications, safety checks, and other regulatory fees. In some areas, such as New York City, a taxi driver must have a "medallion" permitting him or her to pick up passengers. Until recently, New York City taxi medallions regularly auctioned for over $1 million. In such cities, taxi drivers must pay "leases" on their medallions, further bloating their costs.

With consumers switching from traditional taxis to ridesharing services, taxis in New York City face declining market shares. This has led many New York City taxi drivers to "jump ship," leaving their taxis to drive for Uber and Lyft. Last November, *ArsTechnica* reported that White & Blue Group, which manages the largest fleet of leased taxicabs in New York, had at one point been forced to idle as much as 20 percent of its fleet each day. CNN further reports that the situation might lead to a localized financial crisis:

> *The value of [taxi] medallions [in New York City] is taking a huge hit as Uber cars flood the city. The most recent sale price was just $740,000; down nearly 40% from last year...The plunge in medallion values has created a financial crisis for taxi owners. Many owners take out loans to buy the medallions, much as a home buyer would finance a mortgage...That was easy enough to do when the value of the medallions kept going up. But lenders big and small – from Citigroup to credit unions – are [now] refusing to refinance loans for medallion owners...[some are] looking to get out of the medallion lending business entirely. If taxi owners can't refinance their short-term loans, the lenders can seize the medallions that were put up as collateral. That will force many out of business, and the foreclosure sales of their medallions will drive down prices further.*

Taxi drivers, leasing companies, and others affiliated with the traditional taxi industry have made their displeasure with ride-sharing services very clear. Some have lobbied for local, state, and federal governments to impose the same regulations on Uber and Lyft drivers that taxis are required to uphold. In several areas, such as France, violence broke out as taxi drivers took to the streets to protest Uber's business model. In response to such heated opposition, lawmakers have in many cases bowed down to the pressure and called for ride-sharing services to be banned altogether. Uber has been declared illegal in Thailand, Nevada, and several cities in India. There is a long and growing list of cities where Uber has been partially banned, forced to suspend operations, or banned but operating illicitly.

On the other hand, these types of laws are often criticized as protecting traditional taxi industries at the expense of dearly-held capitalist principles. To many champions of the free market, it seems unreasonable to illegalize a faster, cheaper, more convenient service that offers so many benefits to consumers. Others see such bans as a burden to customers who may not be able to afford taxis.

Both sides have valid points, but the outcome of the ongoing struggle between taxis and ride-sharing apps does not have to be winner-take-all. What if there was a way to keep taxis functioning and profitable, while still ensuring the existence of convenient apps like Uber and Lyft? Due to the comparatively high costs of operating a taxi, it is unlikely that taxi fares will decrease in the future. We propose that rather than participating in direct price competition, taxis can remain a viable option in a changing market by making better, data-driven decisions about when and where to offer their services.

This idea was the motivation behind our project, which seeks to examine and explain some of the trends behind taxi use in New York City from 2009 to 2015. By determining peak usage statistics, we hope to formulate a plan allowing New York City taxi drivers to maximize revenue. Assuming constant costs, such a plan will maximize driver profits, incentivizing drivers to stay in the taxi business rather than turning to Uber or Lyft.

**Where did we get our data?** This data was downloaded from the website of the New York City Taxi and Limousine Commission, the agency responsible for regulating taxis in the city. The Commission published a historical dataset covering over 1.1 billion individual taxi trips in the city from January 2009 through December 2015. Each individual trip record contains precise location coordinates for where the trip started and ended, timestamps for when the trip started and ended, fare amount, payment method, number of passengers, and distance traveled.

# II. Methods

New York City taxicabs come in two forms: “yellow,” or medallion taxis, and “green,” or boro taxis. Yellow taxis can pick up passengers anywhere in the five boroughs of the Bronx, Brooklyn, Manhattan, Queens and Staten Island. Beginning in August 2013, green taxis were introduced in order to “improve access to street-hail transportation...for people who live or spend time in areas of New York City historically underserved by the yellow taxi industry” (NYCTLC). These green taxis can drop off passengers anywhere in New York, but are only authorized to pick up passengers in Upper Manhattan, the Bronx, Brooklyn, Queens (excluding LaGuardia Airport and John F. Kennedy International Airport), and Staten Island. Because green taxis face restrictions on where they can pick up passengers, they served to confound the location variable. It should therefore be noted that our project is limited *only* to those taxis which face no pickup restrictions – that is to say, yellow taxis.

We originally intended to examine all available yellow taxi data from January 2009 through December 2015, but we were limited by enormous file sizes. Each csv, representing one month of data, averaged roughly 1.9 GB in size and contained about 14 million cases. Instead of downloading 84 of these massive files, we wrote a bash script that would download the data in a more manageable way. This script removed the first row (containing column names), sampled 20,000 random cases, shuffled them, and removed columns with extraneous information.

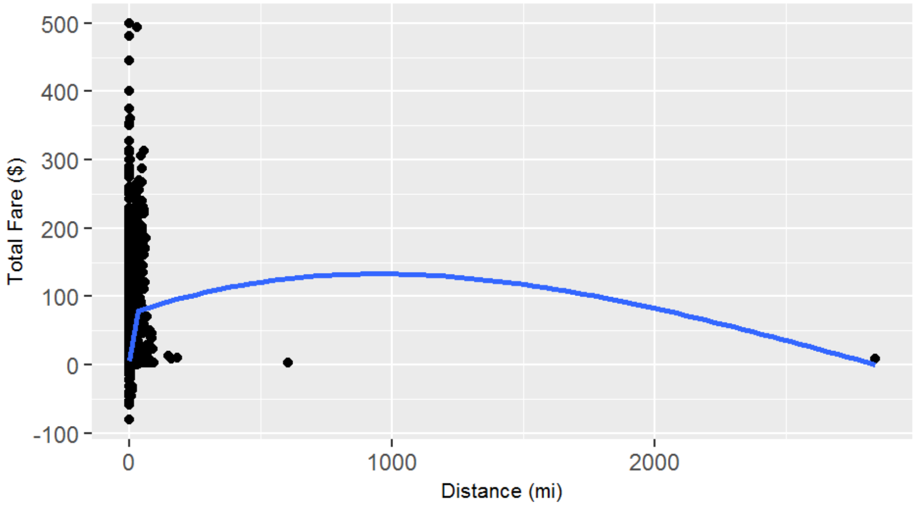
After importing the clean data into R, we combined the monthly data sets into an aggregate combined dataset with 1.68 million cases. Though the data contained several different variables, we didn’t necessarily include all of them in our analysis. For example, New York taxis do not charge extra for multiple passengers. Thus, the number of passengers does not have any effect on revenue. We therefore excluded this variable from our analysis. Furthermore, all New York taxis are equipped with credit card-reading equipment. Payment method, then, also has no effect on revenue, and it is accordingly not considered in our analysis. We instead focus on four areas: average fare (in dollars), average trip length (in miles), number of rides, and pickup locations. We further break several of these factors down by an added time dimension.

### Trip Length and Fare

As a result of regulation by the New York Taxi and Limousine Commission, all New York City taxis have the same pricing structure. The initial flat charge is $2.50, plus $0.50 per 1/5 mile or $0.50 per minute in slow traffic or when the vehicle is stopped. There is a flat $0.50 MTA state surcharge and $0.30 improvement surcharge on every trip. There is an additional charge based on time of day: $0.50 from 8pm to 6am, and $1 from 4pm to 8pm on weekdays. Passengers are also responsible for all bridge and tunnel tolls.

In our dataset, the variable “fare” represents the initial $2.50 flat fare plus $0.50 per mile. The variable “total” is a summation of the flat fare, per mile fare, $0.50 MTA surcharge, $0.30 improvement surcharge, time of day surcharge, and tolls. “Total” more accurately represents a passenger’s true out-of-pocket cost. On the other hand, this variable does not reflect a taxi driver’s profit very well, because these fees and surcharges go straight to the MTA; drivers see no benefit. Considering our objective of maximizing revenue, though, it makes no real difference which measure of fare we use. In any case, from the given cost set-up it should be clear that revenues are correlated to distance traveled, but imperfectly. As a preliminary measure, we therefore plotted fare as a function of trip length.

Figure 1: Distance Traveled v. Fare, Including Outliers



As you can see, these results are heavily skewed by extreme outliers. We therefore filtered the data to remove cases in which distance traveled was greater than 75 miles and cases in which fare was greater than $300. The new graph looked like this:

Figure 2: Distance Traveled v. Fare, Not Including Outliers

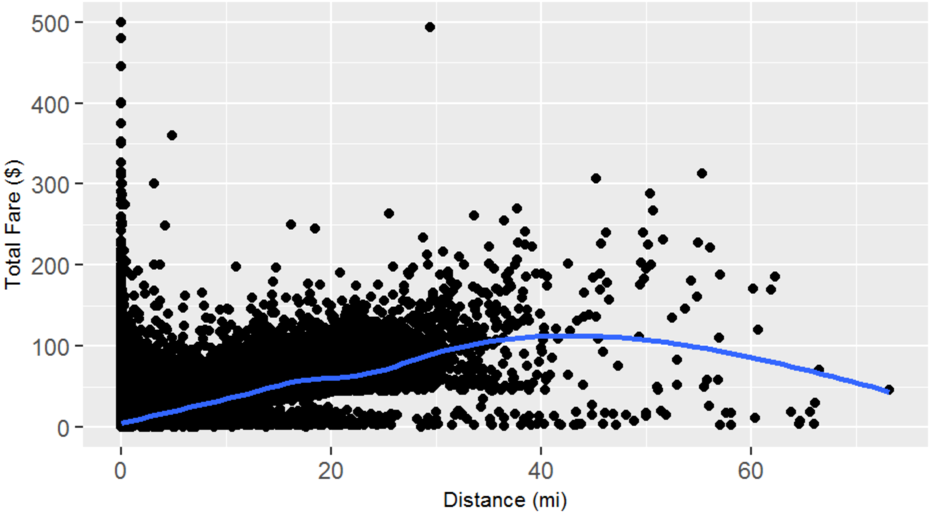


Figure 4: Distance Traveled v. Fare by Month, Not Including Outliers

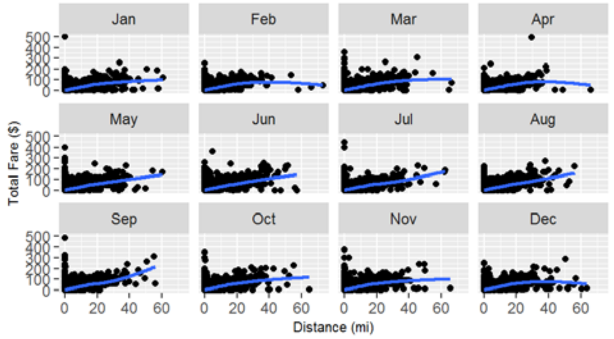
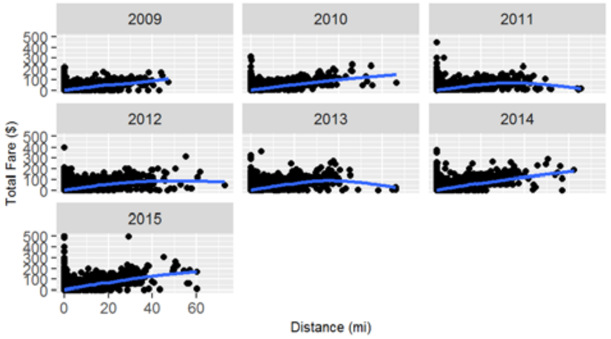


Figure 3: Distance Traveled v. Fare by Year, Not Including Outliers



We then examined average fare by month, average fare by day of week, and average fare by time of day in order to determine which month, day of the week, and hour drivers could expect the highest fares. We further examined average trip length by month, average trip length by day of the week, and average trip length by time of day to determine which month, day of the week, and time of day taxi drivers could expect the longest rides.

## Number of Rides

Because the price structure of a taxi ride includes a flat fare, the revenue a taxi driver receives is also a function of how many rides it produces. We examined the total number of rides per month, the total number of rides by day of week, and the total number of rides per hour to determine which month, day of the week, and time of day taxi drivers could expect the greatest number of rides.

## Location

We examined the most common pick up locations by creating a Google Earth KML file with pushpins to mark pick pick-up longitudes and latitudes. We populated the KML document by writing a for loop to create a new node for each case in our dataset. To make our results easier to interpret, we added a time dimension to the pushpins. By toggling the bar at the top of the document, you can see the pickup locations as a function of time. We coded this time dimension by using regular expressions to rewrite the given timestamps into a form readable by KML. This file is attached to our project submission.

# I. Analysis

## A. Fares

### By Year

Figure 5: Average Fare by Year, Not Including Outliers

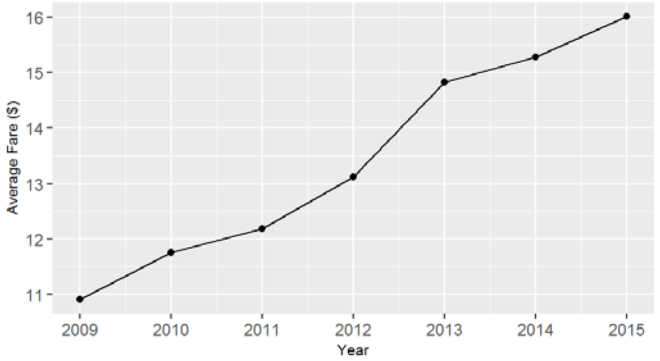


Figure 6: Average Fare by Year, Not Including Outliers

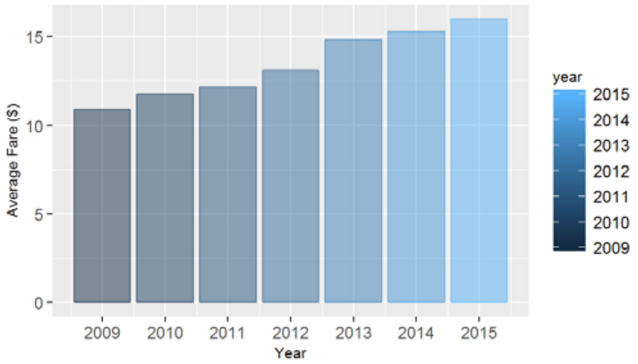
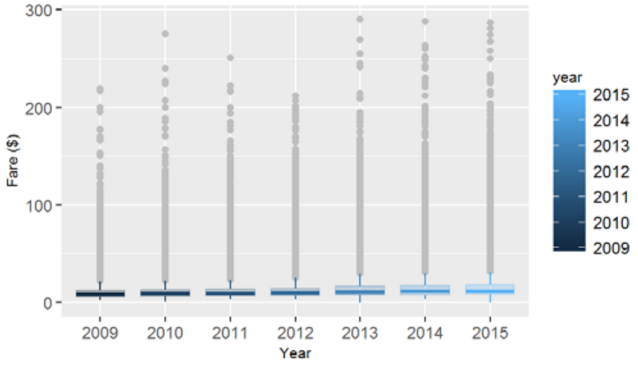


Figure 7: Distribution of Fares by Year, Not Including Outliers



Contrary to what we might expect, we see here that average taxi fares have in fact increased steadily over time. Average fares increased from $11 in 2009 to $16 in 2015 – a hefty 45% change.

### By Month

Figure 8: Average Fare by Month, Not Including Outliers

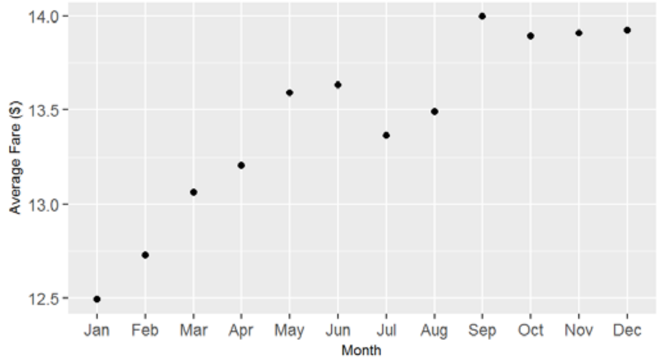


Figure 9: Average Fare by Month, Not Including Outliers

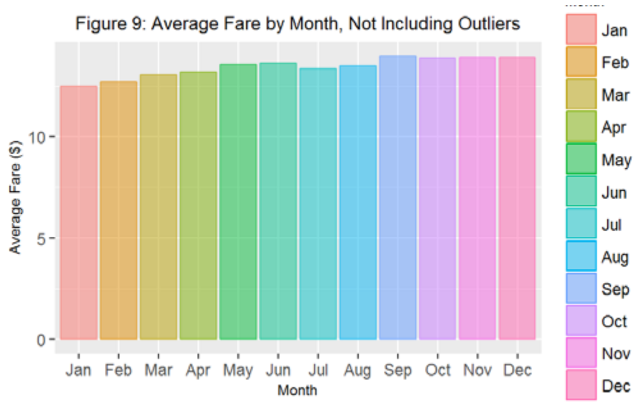
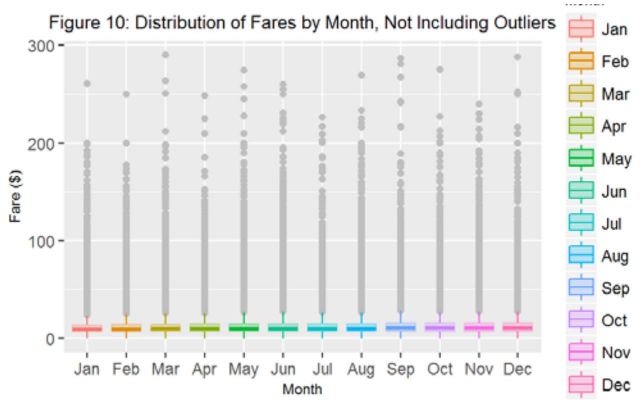


Figure 10: Distribution of Fares by Month, Not Including Outliers



As you can see from the above graphs, the average fare for a taxi ride is lowest in January, at about $12.50, and highest in September, at roughly $14. This represents a 12% percent increase.

## By Day of the Week

Figure 11: Average Fare by Day of the Week, Not Including Outliers

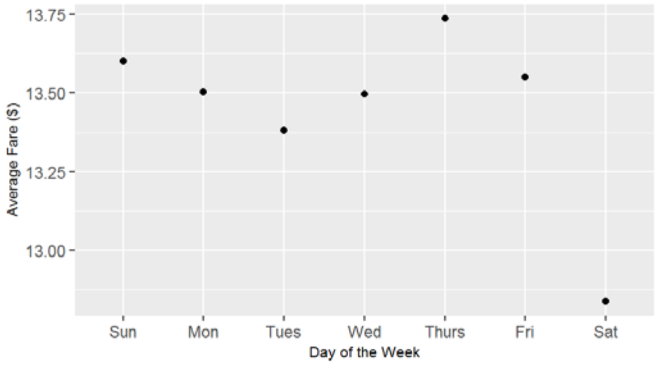


Figure 12: Average Fare by Day of the Week, Not Including Outliers

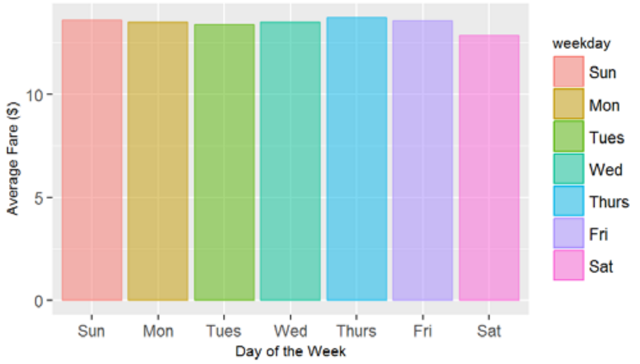
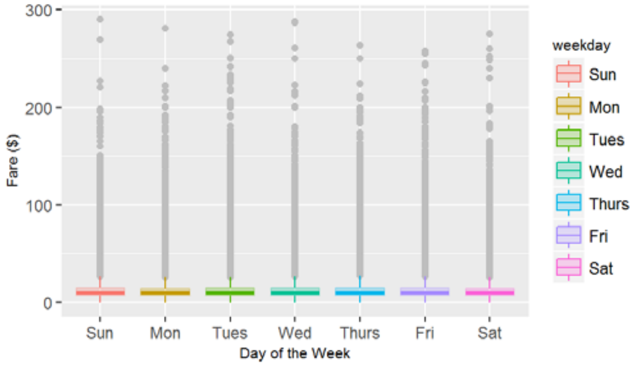


Figure 13: Distribution of Fares by Day of the Week, Not Including Outliers



We also found that the average fare for a taxi ride is lowest on Saturday, at just about $12.80, and highest on Thursday, clocking in at nearly $13.75. In other words, taxi revenues can be up to 7.4% higher on peak days than off days.

## By Time of Day

Figure 14: Average Fare by Hour, Not Including Outliers

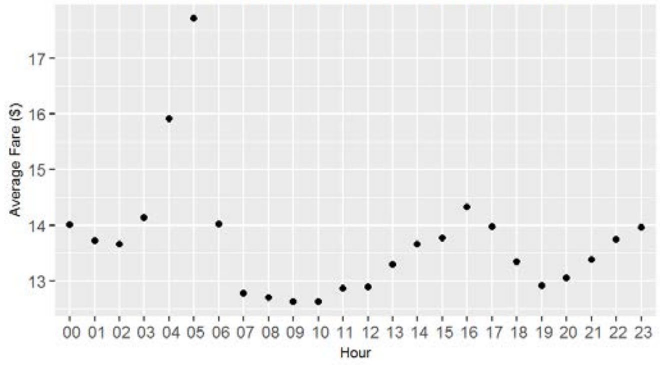


Figure 15: Average Fare by Hour, Not Including Outliers

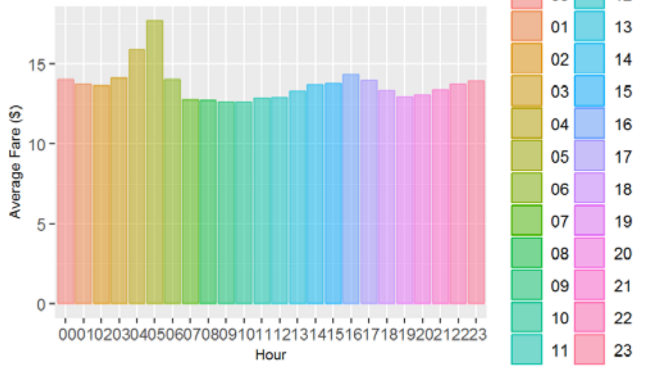
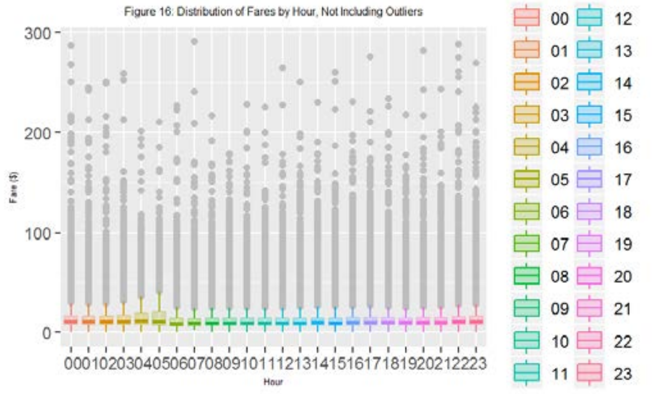


Figure 16: Distribution of Fares by Hour, Not Including Outliers



Unexpectedly, average fares rise dramatically from 4-5 am, spiking at around $17.75. By contrast, the lowest average fare, from 7-8pm, clocks in at roughly $12.50. The data indicates that taxi drivers can expect a whopping 29.5% increase in average fares if they switch from a 7-8pm shift to a 5-6am shift.

What explains the sharp increase in average fares in September, on Fridays, or from 5-6am? Following the logic of economic theory, we might be tempted to believe that customers demand more taxi rides at those times, and therefore fare prices rise in order to take advantage of that demand. Yet taxi prices are set by a regulatory agency and enforced vigorously through GPS trackers and automated meters. The fact that this data was published by said regulatory agency would seem to imply that all cases listed contain only legally-collected fares.

One possible confounding factor is the time of day surcharge ($0.50 from 8pm to 6am, and $1 from 4pm to 8pm on weekdays). This might indicate that the peak fares from 5-6am are “artificially inflated” by fifty cents. Yet this explanation is not sufficient; otherwise, we would expect high average fares from 4-8pm because of the $1 surcharge. In fact, although average fares reach a small peak around 4pm, they actually begin *falling* from then on, reaching a trough at 7pm. That means there are two possible explanations for these peak fare periods: customers purchase longer rides at these times, or customers purchase rides that take them through several toll roads and bridges.

## B. Trip Distance

By Year

Figure 17: Average Trip Distance by Year, Not Including Outliers

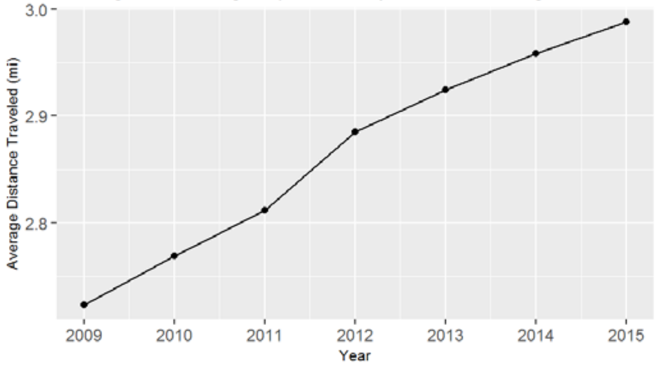


Figure 18: Average Trip Distance by Year, Not Including Outliers

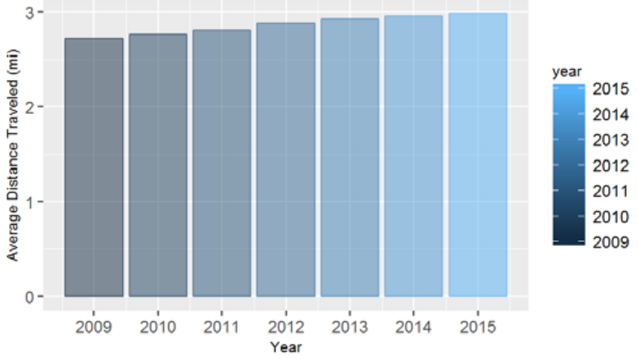
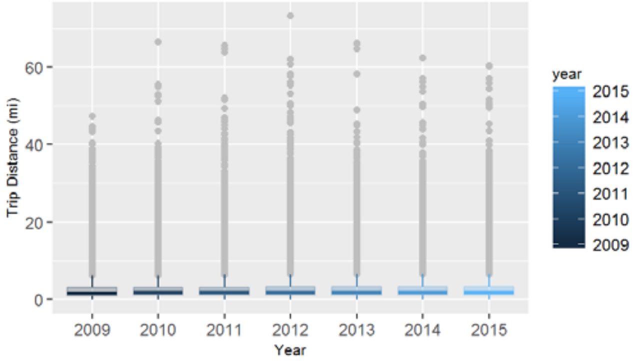


Figure 19: Distribution of Distances by Year, Not Including Outliers



The steady increase in average trip distance by year reflects the steady increase in average fares we found in Figure 5. Average trip distance was lowest in 2009 at about 2.72 miles and greatest in 2015 at nearly 3 miles. This represents a 10.3% change.

By Month

Figure 20: Average Trip Distance by Month, Not Including Outliers

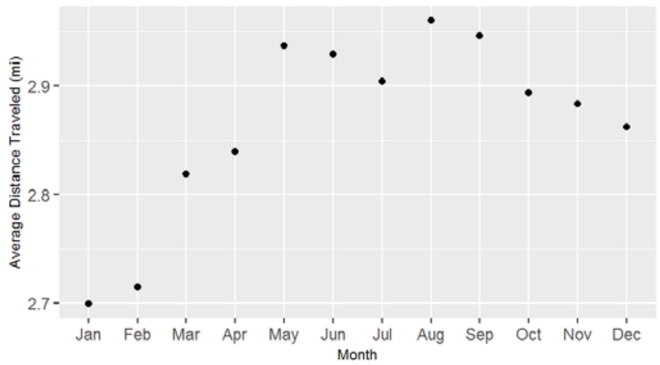


Figure 21: Average Trip Distance by Month, Not Including Outliers

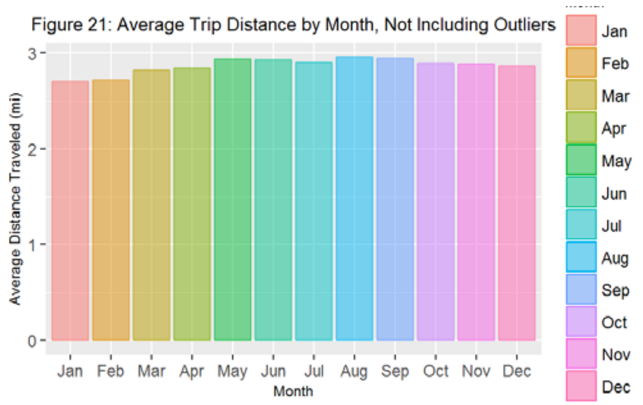


Figure 22: Distribution of Distances by Month, Not Including Outliers

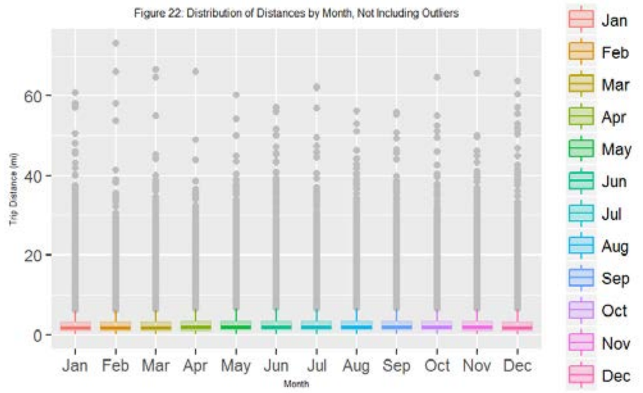


Figure 20 makes it clear that average trip lengths peak in August (2.97 miles on average), with September coming in a close second at roughly 2.95 miles on average, and bottom out in January (2.7 miles on average). This seems to explain the findings in Figure 8 indicating that fares peak in September and reach their lowest levels in January.

## By Day of the Week

Figure 23: Average Trip Distance by Day of the Week, Not Including Outliers

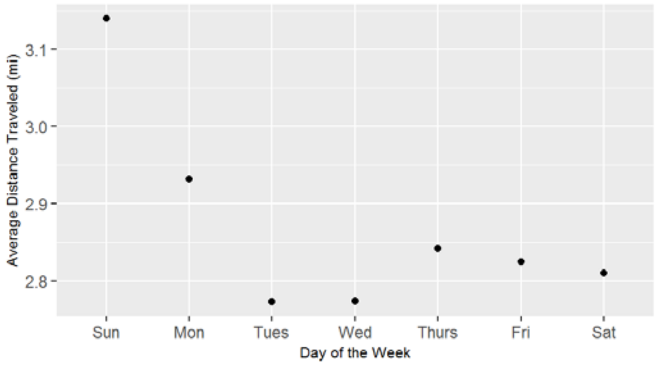


Figure 24: Average Trip Distance by Day of Week, Not Including Outliers

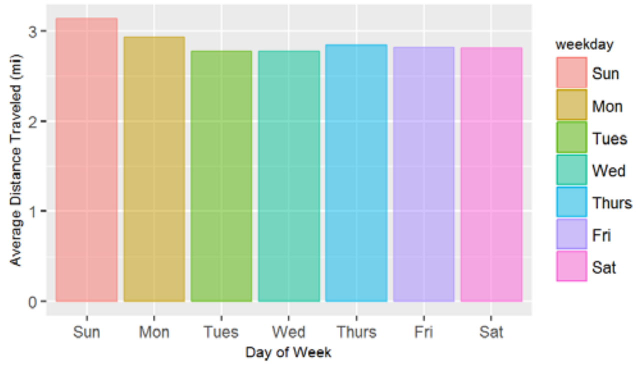


Figure 25: Distribution of Distances by Day of the Week, Not Including Outliers

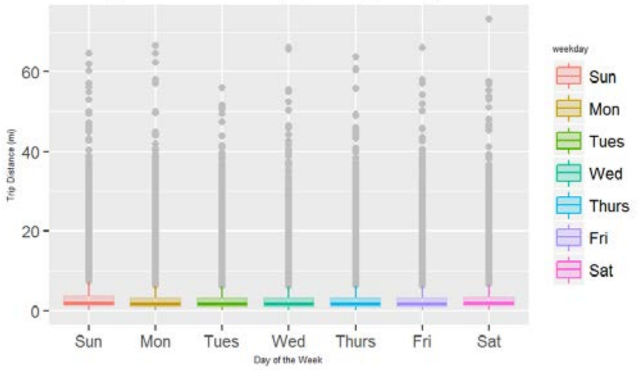


Figure 23, on the other hand, shows that average trip length is not a good explanation for variations in daily average fares. In Figure 11, we found that average fares peak on Thursdays and decrease to their lowest point on Saturdays. But here we see that average trip length is highest on Sundays and lowest on Tuesdays.

## By Time of Day

Figure 26: Average Trip Distance by Hour, Not Including Outliers

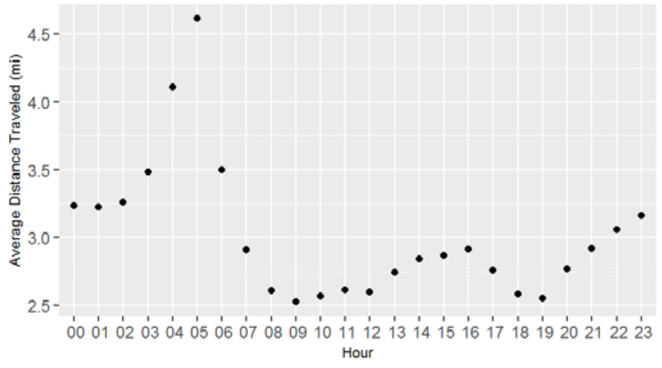


Figure 27: Average Trip Distance by Hour, Not Including Outliers

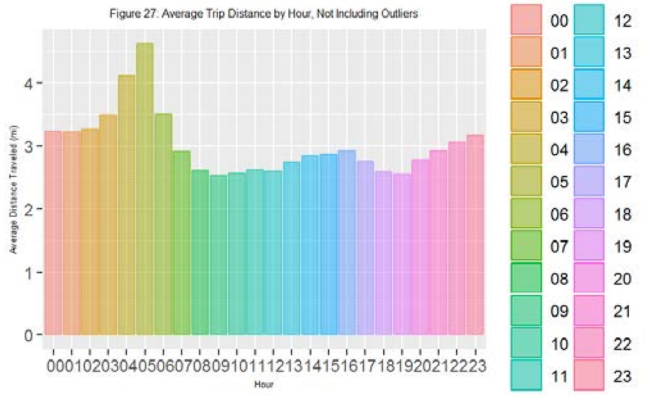
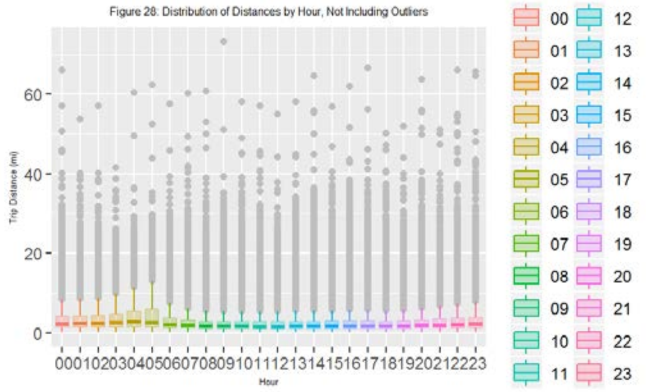


Figure 28: Distribution of Distances by Hour, Not Including Outliers



Finally, we see that average trip length spikes sharply from 4-5am, providing a solid explanation for the average fare spike at the that time as demonstrated in Figure 14. It also explains why average fares are lowest from 7-8pm. In fact, this graph tracks almost perfectly with Figure 14.  This strongly implies that the key to maximizing fares lies in maximizing distance traveled.

## C. Number of Trips

Clearly, maximizing revenue means not only maximizing the average fare of a trip, but also the number of trips a driver makes. For example, a driver who takes several shorter trips may earn more revenue that a driver who makes a few long trips. Our dataset randomly sampled 20,000 cases per month, so the total number of rides by month is always 20,000 and the total number of rides per year is always 240,000. In this section we therefore examine only the day of the week and hour that drivers can expect to deliver the greatest number of rides.

### By Day of the Week

Figure 29: Total Number of Rides by Day of the Week, Not Including Outliers

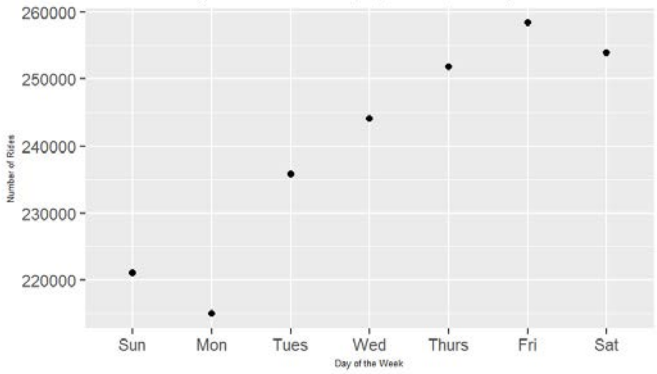
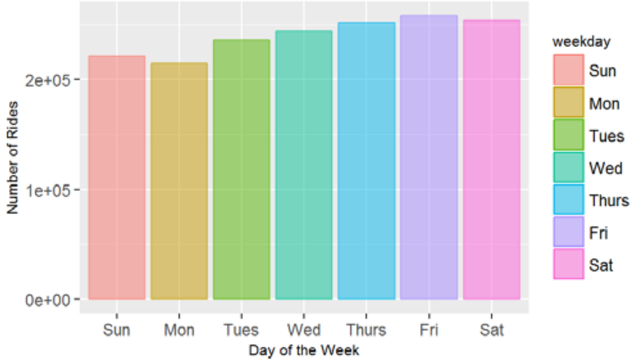


Figure 30: Total Number of Rides by Day of the Week, Not Including Outliers



As these graphs make clear, the total number of rides is greatest on Fridays and lowest on Mondays. This suggests customers are not using taxis to get to work, but instead to travel for leisure.

### By Time of Day

Figure 31: Total Number of Rides by Hour, Not Including Outliers

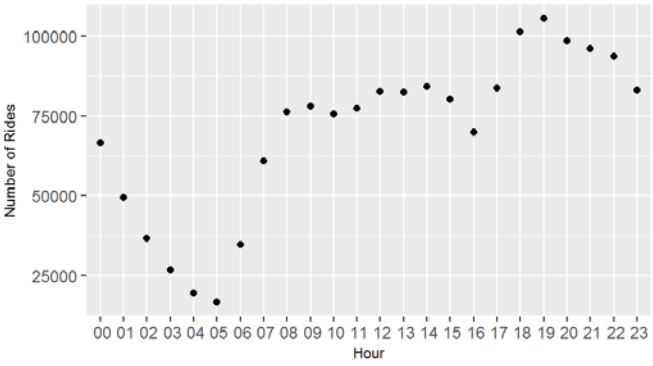


Figure 30: Total Number of Rides by Day of the Week, Not Including Outliers

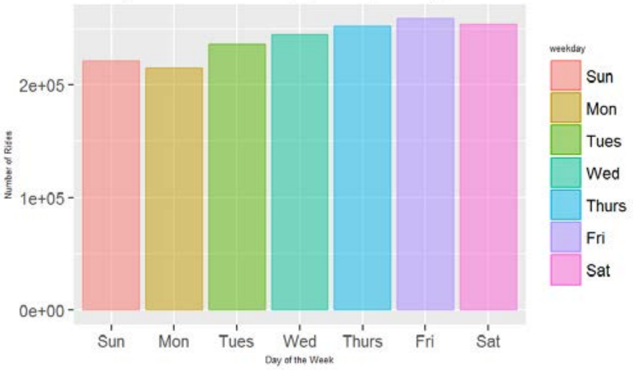


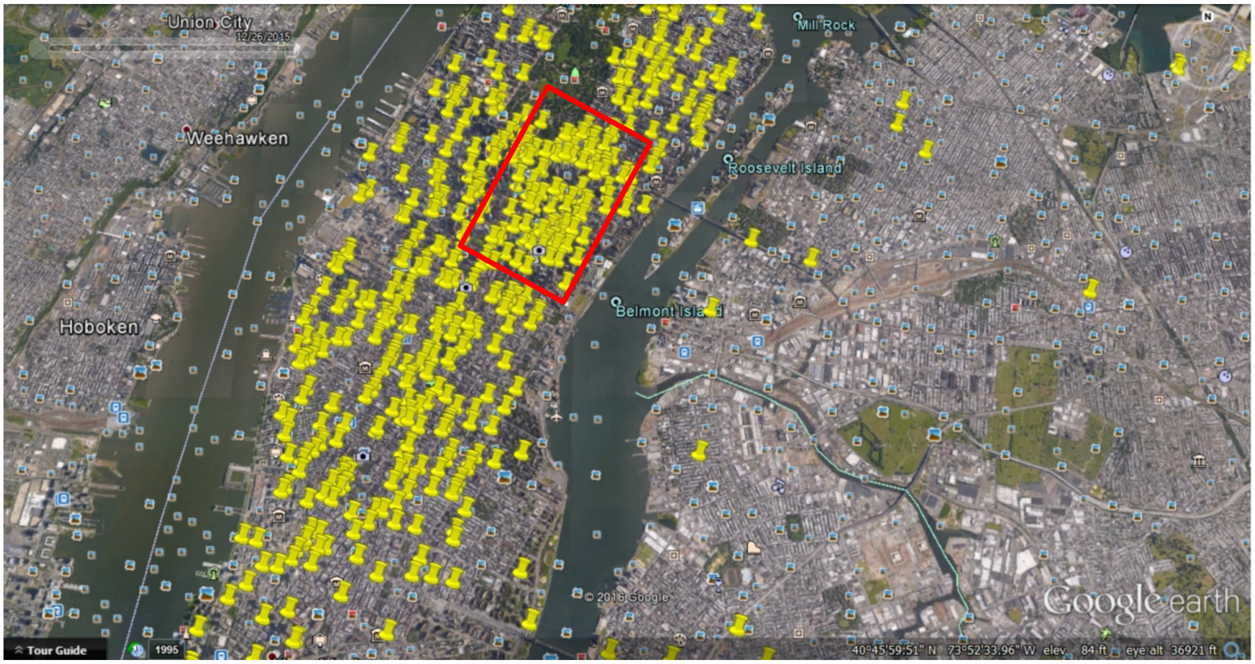
Figure 31 shows a very interesting variation in the data: namely, that the total number of rides is lowest by far from 4-5am and highest from 7-8pm. This is remarkable because in Figure 14 we found that average fares peak from 4-5pm and trough from 7-8pm, and in Figure 26 we found that average trip length similarly fares peak from 4-5pm and trough from 7-8pm.

This leads us to conclude that although there are far fewer trips taken from 4-5am, customers who request rides at that time take significantly longer trips. Furthermore, it seems although that many riders need taxi services from 7-8pm, these trips tend to be quite short.

## D. Location

A driver can earn more fares per hour if he or she is able to cut down on the amount of time it takes to find his or her next customer. This map denotes all New York City taxi pickups from 2009-2015. It shows that the heaviest concentration of pickups is a rectangle south of Central Park in lower Manhattan.

**Figure 11**



The boundaries of this area are West 58th Street (Central Park) to the north, West 42nd Street (Times Square) to the south, Lexington Avenue to the west, and 7th Avenue to the east. This also happens to be one of New York City's prime tourism areas. Interestingly, well-known areas such as Wall Street, SoHo, Chinatown, and Little Italy down south and Harlem up north have significantly fewer data points than mid-Manhattan and the Upper East Side.

# IV. Conclusion

We suggest that New York City taxis can better compete with ride-sharing services like Uber and Lyft by following these guidelines:

- If a driver prefers to make fewer but longer trips, he or she should consider switching to a 4-5am shift.
- If a driver prefers to make short but frequent trips, he or she should plan to work from 7-8pm.
- Drivers should plan to work on Fridays, when there are a greater number of total rides given than any other day of the week.
- The best day for the average taxi driver to take off work is on Monday, the day with the least number of total rides given of any day of the week.
- Drivers should expect the lowest revenue in January. For the months of January and February in particular, taxi leasing agencies may wish to idle a greater portion of their fleets. This would be a good time for taxi drivers to temporarily switch to driving for ride-sharing services.
- On the flip side, drivers should expect the highest revenue in September. Leasing agencies should plan to deploy their entire fleets in August and September, and taxi drivers will likely find a better deal driving a taxi than driving an Uber or Lyft.
- Drivers are almost guaranteed to find passengers in the rectangular area encompassed by West 58th Street to the north, West 42nd Street to the south, Lexington Avenue to the west, and 7th Avenue to the east. This area is in mid-Manhattan, bounded by Central Park and Times Square.

## A. Possible Explanations

“Rush hour,” a time of day when most people are traveling between work, school, and home, is typically from Monday to Friday between 7am to 9am and again from 4pm to 6pm. In that light, it’s difficult to extrapolate what might be going on from 4-5pm and from 7-8pm in New York. On the other hand, it seems fairly self-evident that customers require more rides on Friday than any other day of the week because they have the free time for leisure on that day.

As for the expected drop in revenue in January (and expected peak in August/September), the reason is no secret. A statement given by Uber in January 2016 reads, "Seasonality affects every business, and Uber is no exception because when people hunker down at home, demand for rides drops. We've learned that the single most effective way to boost demand during the winter slump is to cut prices for riders." In other words, customers request fewer rides in cold winter months, preferring to stay home or indoors. The corollary is also true: in warmer months, customers are more likely to use taxis because they are more likely to venture outside the house. NewYork.com further notes that “the closest thing the city has to an “off period” is in January and February, when the weather is at its coldest and hotel occupancy dips briefly in between the craze that is New Year's Eve and the energy that comes with the longer days and warmer weather of spring." In other words, there are fewer tourists in the city in January and February.

The peak pickup locations in the area indicated by our map include some of New York City's most popular tourist attractions. These findings indicate that a majority of the New York taxi customer base might be tourists, rather than those commuting to and from work. Taking hourly, daily, and location-based statistics together, this seems to prove that New York taxi services are largely used for leisure, not as daily transportation.

## B. Areas for Expansion

A future project may want to consider both yellow and green taxi data in order to get a more holistic view of the situation faced by New York taxis. If researchers are able to control for the fact that green taxis are legally limited by where they can pick up passengers, such a project would provide more valuable insight into the strengths and weaknesses of the New York taxi service as a whole.

An even more ambitious proposal would be to find detailed Uber and Lyft data to run a thorough comparison determining how taxis can specifically compete with ride-sharing services. For example, ride-sharing data that included fare amounts would open up so many possibilities for research. With such a dataset, researchers could determine the price differential between an Uber ride, a Lyft ride, and a taxi ride of the same length.

On the other hand, taxis face pre-set fares determined by the Taxi and Limousine Commission. Even if they want to, taxi drivers are therefore unable to price-compete with Uber and Lyft. Economic theory suggests taxis must compete with on-demand services through other ways, such as location, quality of service, branding, safety and reliability. A future paper might consider the best ways for traditional taxis to compete with Uber and Lyft through strategic location placements. We found here that taxi pick-up locations are most concentrated in mid-Manhattan. With detailed Uber and Lyft data, a future project could find Uber and Lyft pickup "hotspots" and lay them over taxi hotspots. This would benefit consumers, who would get better-targeted, more convenient service. This would also benefit drivers, who could maximize revenue by maximizing the number of rides and minimizing the time between passengers.

## V. Works Cited

"Best Time to Visit New York City." *NewYork.com.* 13 Aug. 2014. Web. 2 May 2016. <http://www.newyork.com/articles/travel/best-time-to-visit-nyc-71484/>.

FiveThirtyEight. "Uber TLC FOIA Response." *GitHub.* 14 Oct. 2015. Web. 13 Apr. 2016. <https://github.com/fivethirtyeight/uber-tlc-foil-response/tree/master/uber-trip-data>.

Johnston, Caitlin. "Uber's low fares spark backlash: drivers protest pay cuts, customers may face surge pricing." *Tampa Bay Times.* 14 Jan. 2016. Web. 2 May 2016. <http://www.tampabay.com/news/transportation/ubers-low-fares-spark-backlash-drivers-protest-pay-cuts-customers-may-face/2261405>.

Mullin, Joe. "Cab medallion owners sue NYC, blame Uber for ruining business." *ArsTechnica.* 17 Nov. 2015. Web. 23 Apr. 2016. <http://arstechnica.com/tech-policy/2015/11/cab-medallion-owners-sue-nyc-blame-uber-for-ruining-business/>.

New York City Taxi and Limousine Commission. "TLC Trip Record Data." *New York City Taxi and Limousine Commission.* Web. 13 Apr. 2016. <http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml>.

New York City Taxi and Limousine Commission. "Taxicab Rate of Fare." *New York City Taxi and Limousine Commission.* Web. 23 Apr. 2016. <http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml>.

New York City Taxi and Limousine Commission. "Your Guide to Boro Taxis." *New York City Taxi and Limousine Commission.* Web. 23 Apr. 2016. <http://www.nyc.gov/html/tlc/html/passenger/shl_passenger.shtml>.

Schneider, Todd W. "Analyzing 1.1 Billion NYC Taxi and Uber Trips, with a Vengeance." *ToddWSchneider.com.* 17 Nov. 2015. Web. 13 Apr. 2016. <http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/>.

# VI. Appendix

## A. Bash Script to Clean the Data

```
path="https://storage.googleapis.com/tlc-trip-data/"
yellow="yellow_tripdata_"
green="green_tripdata_"
dir="taxi_data/"
extension=".csv"

for year in `seq 2009 2014`; do
   for month in `seq 1 12`; do
               echo "*******Downloading $year-$month*******"
               if [ $month -lt 10 ]; then
                  url="$path$year/$yellow$year-0$month$extension"
               else
                  url="$path$year/$yellow$year-$month$extension"
               fi
               wget $url
               if [ $month -lt 10 ]; then
                  file="$yellow$year-0$month"
               else
                  file="$yellow$year-$month"
               fi
               lines=($(wc -l "$file$extension"))
               lines=$((lines-1))
               new=$dir$file"_clean"$extension
               echo "*******Cleaning $file$extension*******"
               tail -$lines $file$extension | shuf -n 20000 | cut -f 2,3,4,5,6,7,10,11,12,13,18 -d ','
> $new
               rm $file$extension
       done
done

year="2014"
for month in `seq 1 12`; do
       if [ $month -lt 10 ]; then
          url="$path$year/$green$year-0$month$extension"
       else
          url="$path$year/$green$year-$month$extension"
       fi
```

```
        wget $url
        if [ $month -lt 10 ]; then
           file="$green$year-0$month"
        else
           file="$green$year-$month"
        fi
        lines=($(wc -l "$file$extension"))
        lines=$((lines-1))
        new=$dir$file"_clean"$extension
        echo "*******Cleaning $file$extension*******"
        tail -$lines $file$extension | shuf -n 20000 | cut -f 2,3,4,5,6,7,10,11,12,13,18 -d ',' > $new
        rm $file$extension
done

year="2013"
for month in `seq 8 12`; do
        if [ $month -lt 10 ]; then
           url="$path$year/$green$year-0$month$extension"
        else
           url="$path$year/$green$year-$month$extension"
        fi
        wget $url
        if [ $month -lt 10 ]; then
           file="$green$year-0$month"
        else
           file="$green$year-$month"
        fi
        lines=($(wc -l "$file$extension"))
        lines=$((lines-1))
        new=$dir$file"_clean"$extension
        echo "*******Cleaning $file$extension*******"
        tail -$lines $file$extension | shuf -n 20000 | cut -f 2,3,4,5,6,7,10,11,12,13,18 -d ',' > $new
        rm $file$extension
done

yellow="yellow_tripdata_"
green="green_tripdata_"
dir="taxi_data/"
extension=".csv"
col_names="p_time,d_time,num_passengers,distance,p_lat,p_long,d_lat,d_long,payment_type,fare,total"
```

```
for year in `seq 2009 2015`; do
   for month in `seq 1 12`; do
                  if [ $month -lt 10 ]; then
                     file="$dir$yellow$year-0$month"
                  else
                     file="$dir$yellow$year-$month"
                  fi
                  file=$file"_clean"$extension

                  echo $col_names | cat - $file > "temp"
                  mv "temp" $file
         done
done
```

## B. Importing the Data into R

```
yellow2009_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-01_clean.csv")

yellow2009_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-02_clean.csv")

yellow2009_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-03_clean.csv")

yellow2009_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-04_clean.csv")

yellow2009_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-05_clean.csv")

yellow2009_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-06_clean.csv")

yellow2009_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-07_clean.csv")

yellow2009_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-08_clean.csv")

yellow2009_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2009-09_clean.csv")
```

```
yellow2009_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2009-10_clean.csv")

yellow2009_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2009-11_clean.csv")

yellow2009_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2009-12_clean.csv")

yellow2010_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-01_clean.csv")

yellow2010_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-02_clean.csv")

yellow2010_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-03_clean.csv")

yellow2010_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-04_clean.csv")

yellow2010_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-05_clean.csv")

yellow2010_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-06_clean.csv")

yellow2010_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-07_clean.csv")

yellow2010_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-08_clean.csv")

yellow2010_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-09_clean.csv")
yellow2010_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-10_clean.csv")

yellow2010_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-11_clean.csv")
```

```
yellow2010_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2010-12_clean.csv")

yellow2011_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-01_clean.csv")

yellow2011_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-02_clean.csv")

yellow2011_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-03_clean.csv")

yellow2011_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-04_clean.csv")

yellow2011_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-05_clean.csv")

yellow2011_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-06_clean.csv")

yellow2011_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-07_clean.csv")

yellow2011_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-08_clean.csv")

yellow2011_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-09_clean.csv")

yellow2011_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-10_clean.csv")

yellow2011_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-11_clean.csv")

yellow2011_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2011-12_clean.csv")

yellow2012_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-01_clean.csv")
```

```
yellow2012_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-02_clean.csv")

yellow2012_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-03_clean.csv")

yellow2012_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-04_clean.csv")

yellow2012_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-05_clean.csv")

yellow2012_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-06_clean.csv")

yellow2012_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-07_clean.csv")

yellow2012_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-08_clean.csv")

yellow2012_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-09_clean.csv")

yellow2012_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-10_clean.csv")

yellow2012_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-11_clean.csv")

yellow2012_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2012-12_clean.csv")

yellow2013_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-01_clean.csv")

yellow2013_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-02_clean.csv")

yellow2013_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-03_clean.csv")
```

```
yellow2013_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-04_clean.csv")

yellow2013_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-05_clean.csv")

yellow2013_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-06_clean.csv")

yellow2013_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-07_clean.csv")

yellow2013_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-08_clean.csv")

yellow2013_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-09_clean.csv")

yellow2013_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-10_clean.csv")

yellow2013_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-11_clean.csv")

yellow2013_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2013-12_clean.csv")

yellow2014_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-01_clean.csv")

yellow2014_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-02_clean.csv")

yellow2014_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-03_clean.csv")

yellow2014_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-04_clean.csv")

yellow2014_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-05_clean.csv")
```

```
yellow2014_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-06_clean.csv")

yellow2014_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-07_clean.csv")

yellow2014_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-08_clean.csv")

yellow2014_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-09_clean.csv")

yellow2014_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-10_clean.csv")

yellow2014_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-11_clean.csv")

yellow2014_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2014-12_clean.csv")

yellow2015_1 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-01_clean.csv")

yellow2015_2 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-02_clean.csv")

yellow2015_3 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-03_clean.csv")

yellow2015_4 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-04_clean.csv")

yellow2015_5 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-05_clean.csv")

yellow2015_6 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-06_clean.csv")

yellow2015_7 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-Final/master/taxi_data/yellow_tripdata_2015-07_clean.csv")
```

```
yellow2015_8 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2015-08_clean.csv")

yellow2015_9 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2015-09_clean.csv")

yellow2015_10 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2015-10_clean.csv")

yellow2015_11 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2015-11_clean.csv")

yellow2015_12 <- read.file("https://raw.githubusercontent.com/sudeshna-b/Stat133-
Final/master/taxi_data/yellow_tripdata_2015-12_clean.csv")

ytot <- rbind(yellow2009_1, yellow2009_2, yellow2009_3, yellow2009_4, yellow2009_5,
yellow2009_6, yellow2009_7, yellow2009_8, yellow2009_9, yellow2009_10, yellow2009_11,
yellow2009_12, yellow2010_1, yellow2010_2, yellow2010_3, yellow2010_4, yellow2010_5,
yellow2010_6, yellow2010_7, yellow2010_8, yellow2010_9, yellow2010_10, yellow2010_11,
yellow2010_12, yellow2011_1, yellow2011_2, yellow2011_3, yellow2011_4, yellow2011_5,
yellow2011_6, yellow2011_7, yellow2011_8, yellow2011_9, yellow2011_10, yellow2011_11,
yellow2011_12, yellow2012_1, yellow2012_2, yellow2012_3, yellow2012_4, yellow2012_5,
yellow2012_6, yellow2012_7, yellow2012_8, yellow2012_9, yellow2012_10, yellow2012_11,
yellow2012_12, yellow2013_1, yellow2013_2, yellow2013_3, yellow2013_4, yellow2013_5,
yellow2013_6, yellow2013_7, yellow2013_8, yellow2013_9, yellow2013_10, yellow2013_11,
yellow2013_12, yellow2014_1, yellow2014_2, yellow2014_3, yellow2014_4, yellow2014_5,
yellow2014_6, yellow2014_7, yellow2014_8, yellow2014_9, yellow2014_10, yellow2014_11,
yellow2014_12, yellow2015_1, yellow2015_2, yellow2015_3, yellow2015_4, yellow2015_5,
yellow2015_6, yellow2015_7, yellow2015_8, yellow2015_9, yellow2015_10, yellow2015_11,
yellow2015_12)

ytot <- ytot %>%
  mutate(year = lubridate::year(p_time)) %>%
  mutate(month = lubridate::month(p_time, label=TRUE)) %>%
  mutate(weekday = lubridate::wday(p_time, label=TRUE)) %>%
  mutate(hour = str_sub(ytot$p_time, -9, -7))
```

## C. GGPLOT

### Fares

```
#total amount paid (including surchage, tolls, etc) v. distance traveled
ytot %>%
  ggplot(aes(y = total, x = distance)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  ggtitle("Figure 1: Distance Traveled v. Fare, Including Outliers") +
  ylab("Total Fare ($)") +
  xlab("Distance (mi)") +
  theme(title = element_text(size=8))

#entire dataset, without outliers
ytot %>%
  filter(total > 0, distance < 75) %>%
  ggplot(aes(y = total, x = distance)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  ggtitle("Figure 2: Distance Traveled v. Fare, Not Including Outliers") +
  ylab("Total Fare ($)") +
  xlab("Distance (mi)") +
  theme(title = element_text(size=8))

#faceted by year, without outliers
ytot %>%
  filter(total > 0, distance < 75) %>%
  ggplot(aes(y = total, x = distance)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~ year) +
  ggtitle("Figure 3: Distance Traveled v. Fare by Year, Not Including Outliers") +
  ylab("Total Fare ($)") +
  xlab("Distance (mi)") +
  theme(title = element_text(size=8))

#faceted by month, without outliers
ytot %>%
  filter(total > 0, distance < 75) %>%
  ggplot(aes(y = total, x = distance)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~ month) +
  ggtitle("Figure 4: Distance Traveled v. Fare by Month, Not Including Outliers") +
  ylab("Total Fare ($)") +
  xlab("Distance (mi)") +
  theme(title = element_text(size=8))

#average fare by year, without outliers
ytot_year_ave_fare <- ytot %>%
  filter(total > 0, total < 300) %>%
  group_by(year)%>%
  summarize(ave=mean(total))

#line graph
ytot_year_ave_fare %>%
  ggplot(aes(x=year, y=ave)) +
  geom_point() +
  geom_line()+
  scale_x_continuous(breaks=seq(2009,2015,1)) +
  ggtitle("Figure 5: Average Fare by Year, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#bar graph
ytot_year_ave_fare %>%
  ggplot(aes(x=year, y=ave, col=year, fill=year)) +
  geom_bar(stat ="identity", alpha=.5) +
  scale_x_continuous(breaks=seq(2009,2015,1)) +
  ggtitle("Figure 6: Average Fare by Year, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(total > 0, total < 300) %>%
  ggplot(aes(factor(year), total)) +
  geom_boxplot(aes(color=year, fill=year), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 7: Distribution of Fares by Year, Not Including Outliers") +
  ylab("Fare ($)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#average fare by month, without outliers
ytot_month_ave_fare <- ytot %>%
  filter(total > 0, total < 300) %>%
  group_by(month)%>%
  summarize(ave=mean(total))

#line graph
ytot_month_ave_fare %>%
  ggplot(aes(x=month, y=ave)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 8: Average Fare by Month, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#bar graph
ytot_month_ave_fare %>%
  ggplot(aes(x=month, y=ave, col=month, fill=month)) +
  geom_bar(stat ="identity", alpha=0.5) +
  ggtitle("Figure 9: Average Fare by Month, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(total > 0, total < 300) %>%
  ggplot(aes(factor(month), total)) +
  geom_boxplot(aes(color=month, fill=month), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 10: Distribution of Fares by Month, Not Including Outliers") +
  ylab("Fare ($)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#average fare by day of the week, without outliers
ytot_day_ave_fare <- ytot %>%
  filter(total > 0, total < 300) %>%
  group_by(weekday)%>%
  summarize(ave=mean(total))

#line graph
ytot_day_ave_fare %>%
  ggplot(aes(x=weekday, y=ave)) +
  geom_point() +
  geom_line()+
  ggtitle("Figure 11: Average Fare by Day of the Week, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#bar graph
ytot_day_ave_fare %>%
  ggplot(aes(x=weekday, y=ave, col=weekday, fill=weekday)) +
  geom_bar(stat ="identity", alpha=.5) +
  ggtitle("Figure 12: Average Fare by Day of the Week, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(total > 0, total < 300) %>%
  ggplot(aes(factor(weekday), total)) +
  geom_boxplot(aes(color=weekday, fill=weekday), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 13: Distribution of Fares by Day of the Week, Not Including Outliers") +
  ylab("Fare ($)") +
  xlab( "Day of the Week") +
  theme(title = element_text(size=8))

#average fare by time of day, without outliers
ytot_hour_ave_fare <- ytot %>%
  filter(total > 0, total < 300) %>%
  group_by(hour) %>%
  summarize(ave=mean(total))

#line graph
ytot_hour_ave_fare %>%
  ggplot(aes(x=hour, y=ave)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 14: Average Fare by Hour, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Hour") +
  theme(title = element_text(size=8))

#bar graph
ytot_hour_ave_fare %>%
  ggplot(aes(x=hour, y=ave, col=hour, fill=hour)) +
  geom_bar(stat ="identity", alpha=.5) +
  ggtitle("Figure 15: Average Fare by Hour, Not Including Outliers") +
  ylab("Average Fare ($)") +
  xlab("Hour") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(total > 0, total < 300) %>%
  ggplot(aes(factor(hour), total)) +
  geom_boxplot(aes(color=hour, fill=hour), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 16: Distribution of Fares by Hour, Not Including Outliers") +
  ylab("Fare ($)") +
  xlab("Hour") +
  theme(title = element_text(size=8))
```

## Trip Length/Distance

```
#average trip length by year, without outliers
ytot_year_ave_dist <- ytot %>%
  filter(distance > 0, distance < 75)%>%
  group_by(year)%>%
  summarize(ave=mean(distance))

#line graph
ytot_year_ave_dist %>%
  ggplot(aes(x=year, y=ave)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks=seq(2009, 2015, 1)) +
  ggtitle("Figure 17: Average Trip Distance by Year, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#bar graph
ytot_year_ave_dist %>%
  ggplot(aes(x=year, y=ave, col=year, fill=year)) +
  geom_bar(stat ="identity", alpha=.5) +
  scale_x_continuous(breaks=seq(2009, 2015, 1)) +
  ggtitle("Figure 18: Average Trip Distance by Year, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(distance > 0, distance < 75) %>%
  ggplot(aes(factor(year), distance)) +
  geom_boxplot(aes(color=year, fill=year), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 19: Distribution of Distances by Year, Not Including Outliers") +
  ylab("Trip Distance (mi)") +
  xlab("Year") +
  theme(title = element_text(size=8))

#average trip length by month, without outliers
ytot_month_ave_dist <- ytot %>%
  filter(distance > 0, distance < 75)%>%
  group_by(month)%>%
  summarize(ave=mean(distance))

#line graph
ytot_month_ave_dist %>%
  ggplot(aes(x=month, y=ave)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 20: Average Trip Distance by Month, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#bar graph
```

```
ytot_month_ave_dist %>%
  ggplot(aes(x=month, y=ave, col=month, fill=month)) +
  geom_bar(stat ="identity", alpha=0.5) +
  ggtitle("Figure 21: Average Trip Distance by Month, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(distance > 0, distance < 75) %>%
  ggplot(aes(factor(month), distance)) +
  geom_boxplot(aes(color=month, fill=month), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 22: Distribution of Distances by Month, Not Including Outliers") +
  ylab("Trip Distance (mi)") +
  xlab("Month") +
  theme(title = element_text(size=8))

#average trip length by day of the week, without outliers
ytot_day_ave_dist <- ytot %>%
  filter(distance > 0, distance < 75)%>%
  group_by(weekday)%>%
  summarize(ave=mean(distance))

#line graph
ytot_day_ave_dist %>%
  ggplot(aes(x=weekday, y=ave)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 23: Average Trip Distance by Day of the Week, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#bar graph
ytot_day_ave_dist %>%
  ggplot(aes(x=weekday, y=ave, col=weekday, fill=weekday)) +
  geom_bar(stat ="identity", alpha=0.5) +
  ggtitle("Figure 24: Average Trip Distance by Day of Week, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Day of Week") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(distance > 0, distance < 75) %>%
  ggplot(aes(factor(weekday), distance)) +
  geom_boxplot(aes(color=weekday, fill=weekday), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 25: Distribution of Distances by Day of the Week, Not Including Outliers") +
  ylab("Trip Distance (mi)") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#average trip length by time of day, without outliers
ytot_hour_ave_dist <- ytot %>%
  filter(distance > 0, distance < 75) %>%
  group_by(hour) %>%
  summarize(ave=mean(distance))

#line graph
ytot_hour_ave_dist %>%
  ggplot(aes(x=hour, y=ave)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 26: Average Trip Distance by Hour, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Hour") +
  theme(title = element_text(size=8))

#bar graph
ytot_hour_ave_dist %>%
  ggplot(aes(x=hour, y=ave, col=hour, fill=hour)) +
  geom_bar(stat ="identity", alpha=.5) +
  ggtitle("Figure 27: Average Trip Distance by Hour, Not Including Outliers") +
  ylab("Average Distance Traveled (mi)") +
  xlab("Hour") +
  theme(title = element_text(size=8))

#boxplot
ytot %>%
  filter(distance > 0, distance < 75) %>%
  ggplot(aes(factor(hour), distance)) +
  geom_boxplot(aes(color=hour, fill=hour), alpha=.25, outlier.colour="gray") +
  ggtitle("Figure 28: Distribution of Distances by Hour, Not Including Outliers") +
  ylab("Trip Distance (mi)") +
  xlab("Hour") +
  theme(title = element_text(size=8))
```

### Number of Rides

```
#total number of taxi rides for each day of week
ytot_day_rides <- ytot %>%
  group_by(weekday) %>%
  summarise(tot = n())

#line graph
ytot_day_rides %>%
  ggplot(aes(y = tot, x = weekday)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 29: Total Number of Rides by Day of the Week, Not Including Outliers") +
  ylab("Number of Rides") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#bar graph
ytot_day_rides %>%
  ggplot(aes(y=tot, x=weekday, col=weekday, fill=weekday)) +
  geom_bar(stat ="identity", alpha=0.5) +
  ggtitle("Figure 30: Total Number of Rides by Day of the Week, Not Including Outliers") +
  ylab("Number of Rides") +
  xlab("Day of the Week") +
  theme(title = element_text(size=8))

#total number of taxi rides per hour
ytot_hour_rides <- ytot %>%
  group_by(hour) %>%
  summarize(tot = n())

#line graph
ytot_hour_rides %>%
  ggplot(aes(x = hour, y = tot)) +
  geom_point() +
  geom_line() +
  ggtitle("Figure 31: Total Number of Rides by Hour, Not Including Outliers") +
  ylab("Number of Rides") +
  xlab("Hour") +
  theme(title = element_text(size=8))

#bar graph
ytot_hour_rides %>%
  ggplot(aes(x=hour, y=tot, col=hour, fill=hour)) +
  geom_bar(stat ="identity", alpha=0.5) +
  ggtitle("Figure 32: Total Number of Rides by Hour, Not Including Outliers") +
  ylab("Number of Rides") +
  xlab("Hour") +
  theme(title = element_text(size=8))
```

## D. KML

```
rand <- sample(1:240000, 100, replace=F)

library(XML)
doc <- newXMLDoc()
root <- newXMLNode(name = "kml", namespaceDefinitions =
"http://www.opengis.net/kml/2.2", doc = doc)
d <- newXMLNode(name = "Document", parent = root)
nm <- newXMLNode(name = "name", "Taxi", parent = d)
description <- newXMLNode(name = "description", "Taxi Rides in  NYC, 2009-2015", parent =
d)

datetime <- as.character(ytot$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat15[i],",", long15[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}
saveXML(doc, "~/Desktop/taxi_2015.kml")
```

```
library(XML)
doc <- newXMLDoc()
root <- newXMLNode(name = "kml", namespaceDefinitions =
"http://www.opengis.net/kml/2.2", doc = doc)
d <- newXMLNode(name = "Document", parent = root)
nm <- newXMLNode(name = "name", "Taxi", parent = d)
description <- newXMLNode(name = "description", "Taxi Rides in  NYC, 2009-2015", parent =
d)

datetime <- as.character(ytot$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat15[i],",", long15[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y14$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat14[i],",", long14[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y13$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat13[i],",", long13[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y12$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat12[i],",", long12[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y11$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat11[i],",", long11[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y10$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat10[i],",", long10[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}

datetime <- as.character(y09$p_time)
datetime <- gsub("/", "-", datetime)
datetime <- gsub(" ", "T", datetime)
datetime <- gsub("$", "Z", datetime)

for (i in rand){
  pm <- newXMLNode(name = "Placemark", parent = d)
  p <- newXMLNode(name = "Point", parent = pm)
  coor <- newXMLNode(name = "coordinates", c(lat09[i],",", long09[i]), parent = p)
  ts <- newXMLNode(name = "TimeStamp", parent = pm)
  when <- newXMLNode("when", datetime[i], parent = ts)
}
saveXML(doc, "~/Desktop/taxi_all.kml")
```