Tools for Understanding Taxicab and E-Hail Service Use in New York City

---

A Thesis

Presented to

The Division of

Smith College

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Arts

Wencong (Priscilla) Li

May 2018

Approved for the Division

(Statistical and Data Sciences)

Benjamin Baumer

Acknowledgements

I would love to thank my thesis advisor Benjamin Baumer, Assistant Professor of Statistical and Data Sciences at Smith College, for encouraging me to challenge myself and guiding me through this project. I want to thank Jordan Crouser for being my second reader and help me to revise my paper. I also want to thank all my friends and my roommates for their love and support.

Table of Contents

List of Tables

List of Figures

Abstract

Yellow Taxi Cab is widely recognized as an important part of New York City. Each taxi trip record is like a little piece of a gigantic puzzle, and all together they tell a story of what happens in New York City everyday. This thesis presents a more efficient and easy-to-use way for users to retrieve trip information of both taxi and other ride-sharing services, such as Uber and Lyft, in New York City. By analyzing trip records of New York City's yellow cab, we answer questions that are commonly asked by taxi drivers, passengers, and TLC officials to help all three parties to improve their services or experiences.

Chapter 1

Introduction

1.1 Motivation

When is the best time during a day to travel to JFK Airport from Brooklyn? How much tip do passengers usually pay to the taxi drivers? Is the $52 flat rate from Manhattan to JFK Airport appropriate? Questions about New York City taxicabs are frequently asked by people travelling in taxis in New York City. New York City Taxi and Limousine Commission (TLC) provides publicly accessible yellow and green taxi trip records on their website for people to investigate and answer these questions. However, it is not easy to work with taxi trip data provided by TLC, because there are more than 250,000 taxi trips happening everyday in New York City (Whitford, 2017), resulting in large size of the datasets. The size of 2017 yellow taxi trip data CSV file is about 10 GB, and this dataset is too big to be processed in an R session. We call data that is too big to be loaded into R environment but not too big to be saved on a hard drive medium data.

Working with medium data, such as the taxi TLC trips records, in **R** is not an easy task. Loading medium-sized data into the **R** environment takes a long time and might

crash an **R** session. Creating a user-friendly platform that allows **R** users to easily work with medium data is our motivation. In our study, we focus on New York City taxicab data because there are a lot of interesting questions about New York City taxicabs that we want to explore.

New York City taxi drivers, passengers, and New York City TLC are the three parties who are closely involved in the New York City taxi industry. Each party has its own needs. Better understanding the needs of the three parties and providing solutions to satisfy their needs is the goal of this thesis.

This work contains two main components. The first component is building the tool to work with the TLC taxi trip data, and the second component is using the tool we build to understand the taxicab and e-hail service use in New York City.

1.2 Background

1.2.1 Yellow Taxi

NYC Taxicabs are operated by private firms and licensed by the New York City Taxi and Limousine Commission (TLC). TLC issues medallions to taxicabs, and every taxicab must have a medallion to operate. There were 13,437 yellow medallion taxicabs licenses in 2014, and taxi patronage has declined since 2011 because of the competition caused by rideshare services (N. T. Staff, 2018).

1.2.2 Green Taxi

The apple green taxicabs in New York City are called Boro taxis and they are only allowed to pick up passengers in the outer boroughs and in Manhattan above East

96th and West 110th Streets. Historically, only the yellow medallion taxicabs were allowed to pick up passengers on the street. However, since 95% of yellow taxi pick-ups occurred in Manhattan to the South of 96th Street and at the two airports, the Five Borough Taxi Plan was started to allow green taxis to fill in the gap in outer boroughs in the summer of 2013 (N. T. Staff, 2009d).

1.2.3 Uber

Uber Technologies Inc. is an American technology company that operates private cars worldwide. Uber drivers use their own cars, instead of corporate-owned vehicles, to pick up passengers (U. Staff, 2009). Uber NYC was launched in May 2011. In NYC, Uber uses 'upfront pricing', meaning that riders are informed about the fares that they will pay before requesting a ride, and gratuity is not required. Riders are given the opportunity to compare different transportations' fares before making their decisions on which one to choose ("Uber Moves New York City," 2015).

1.2.4 Lyft

Similar to Uber, Lyft is also an on-demand transportation company, and it operates the Lyft car transportation mobile app. Lyft is the main competitor of Uber, and it came into market in July 2014 in New York City ("Uber Moves New York City," 2015).

1.3 Literature Review

1.3.1 New York City Traffic and Taxi

New York City is one of the most popular cities in the United States. New York City's traffic is a popular topic in journalism, and different aspects of it has been studied by journalists. New York City's traffic is a "nightmare", and the city officials have long been trying to solve the congestion problem. In 2009, New York City was voted to be the U.S. city with the "angriest and most aggressive drivers", and the bad temper of drivers are exacerbated by New York City's severe cogestion (Reaney, 2009).

How bad is the cogestion? In "NYC is already tired of Christmas and Donald Trump", New York City has been described as "the city that never moves" (Furfaro, Cohen, & Fears, 2016). What has led to the congestion in the city? According to "Uber and Lyft cars now outnumber yellow cabs in NYC 4 to 1", "the city streets are being engineered to create traffic congestion, to slow traffic down, to favor bikers and pedestrians" so that drivers will have the incentive to leave their cars at home and turn to mass transit or bicycles (Sugar, 2017).

No matter how miserable the driving experiences are, taxi drivers have no luxury to choose alternative transportation, and instead thay have to consistently drive their cabs, which are usually surrounded by bad traffic, in order to make a living.

1.3.2 Competition between New York City taxi and e-hail services

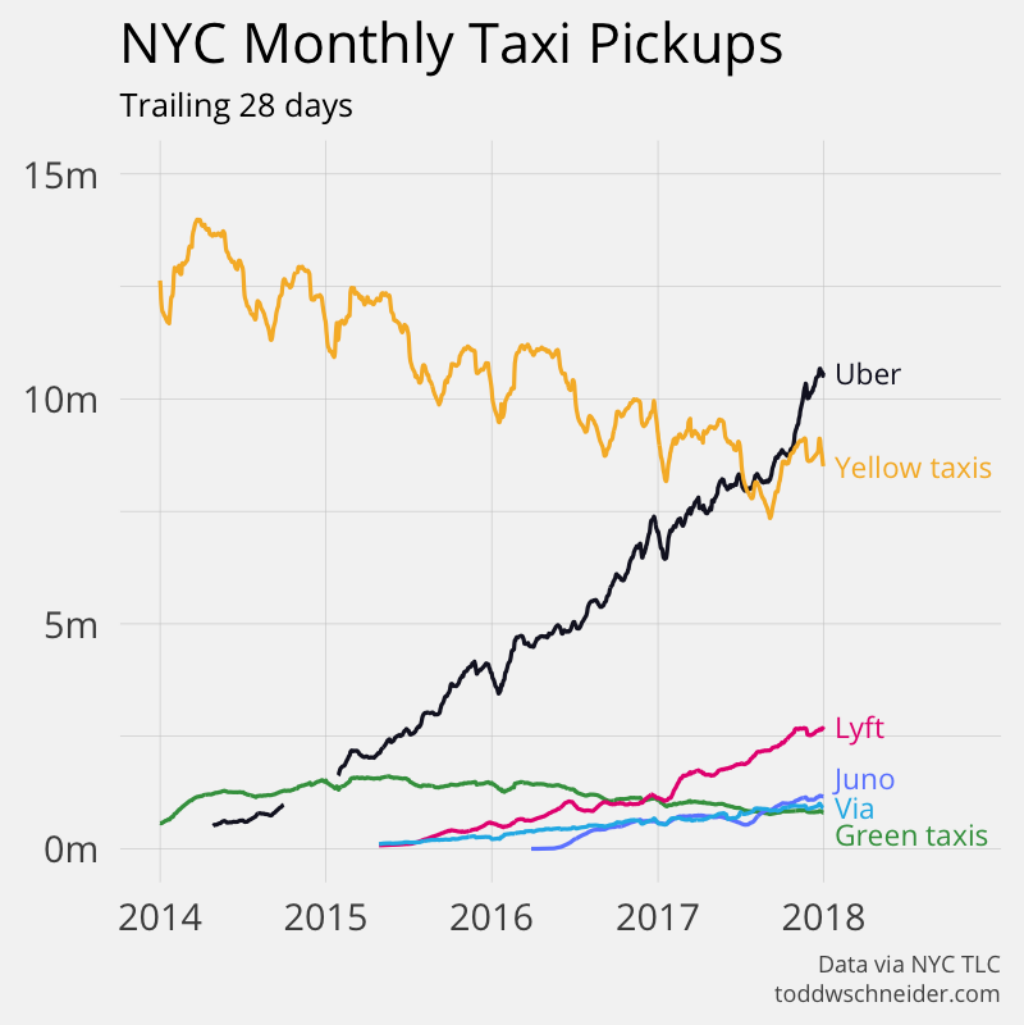


Figure 1.1: NYC Monthly Taxi Pickups

As shown in the visulization above (Schneider, 2015), the number of New York City yellow taxi trips has been consistently declining for about 4 years, and the number of Uber and Lyft trips keep increasing. In 2017, for the first time, the total number of monthly Uber trips exceeded the number of yellow taxi trips.

Some studies have shown how competitive Uber and Lyft are. In 2017, Uber and Lyft registered vehicles outnumbered NYC yellow cabs by 4 to 1 (Sugar, 2017). Even

though yellow cabs used to be the most popular street-hail transportation service in New York City, passengers nowadays tend to choose the more convenient options, ride-hailing apps (Hu, 2017).

Data scientists from the University of Cambridge in the UK and the University of Namur in Belgium found that yellow taxi rides are on average $1.40 cheaper than Uber X, which is one type of economy ride service offered by Uber (H. Staff, 2018). Moreover, Uber appears more expensive for trips that are cheaper than $35, and less expensive than yellow taxi ride for trips that are more expensive than $35. Therefore, for short trips, taking a taxi is more affordable (Vsevolod Salnikov, 2015).

Apps, such as Openstreetcab, that compare the price of Uber and taxi trips are designed to help customers to compare the fares of different transportations (Vsevolod Salnikov, 2015).

1.3.3 `etl` R package

Working with taxi trip data is not an easy task, because of the large size of the taxi trip datasets (Whitford, 2017). Loading these datasets into **R** environment takes a long time and might crush an **R** session. Taxi trip datasets are classified as medium data, because they are too big to be processed in an R session but not too big to be saved on a hard drive.

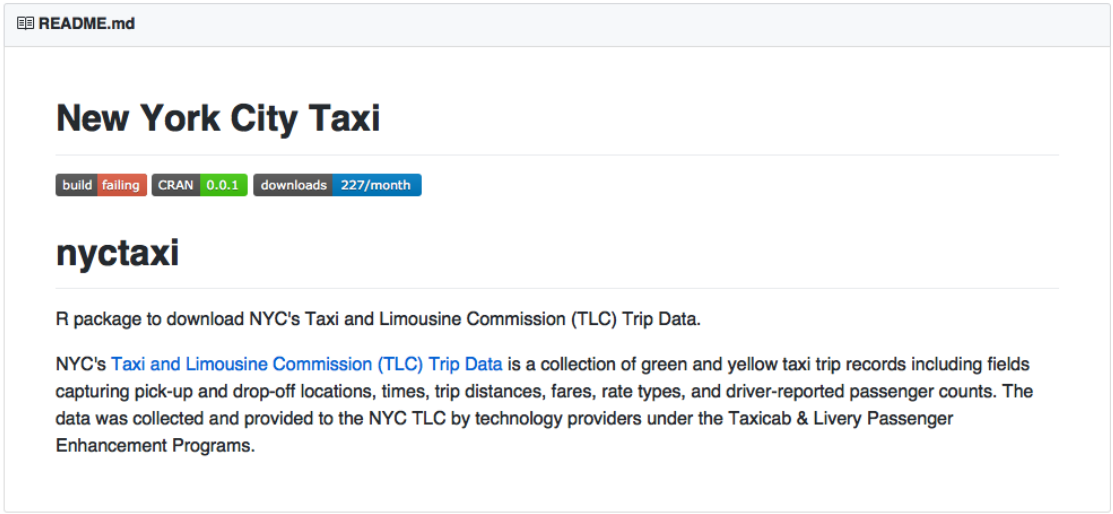
The `etl` **R** package creates a user-friendly platform that allows **R** users to easily work with medium data with the extract, transform, load framework, which is commonly konwn as ETL in computing. The `ETL` process has been set up (B. S. Baumer, 2017) in **R** to facilitate etl operations for medium data, and it is designed to work with any general data set. Packages that are specific to perticular data sets are needed to be written in order to better work with complex medium-sized data sets.

1.4 Contribution

This thesis has two main components: the `nyctaxi` **R** package, which helps users to analyze the New York City street-hail services' data in **R**, and recommendations for taxi drivers, passengers, and TLC officials. In addition to the two main parts, we focus on making all analysis and visulizations in this study reproducible.

1.4.1 'nyctaxi' Package

`nyctaxi` is an **R** package that help users to easily get access to New York City Taxi, Uber and Lyft trip data through Extract, Transform, and Load functions (ETL) (B. S. Baumer, 2017). This package facilitates ETL to deal with medium data that are too big to store in memory on a laptop. Users are given the option to choose specific years and months as the input parameters of the three ETL functions, and a connection to a populated SQL database will be returned as the output. Users do not need to learn SQL queries, since all user interaction is in **R**.


README.md

New York City Taxi

build failing CRAN 0.0.1 downloads 227/month

nyctaxi

R package to download NYC's Taxi and Limousine Commission (TLC) Trip Data.

NYC's Taxi and Limousine Commission (TLC) Trip Data is a collection of green and yellow taxi trip records including fields capturing pick-up and drop-off locations, times, trip distances, fares, rate types, and driver-reported passenger counts. The data was collected and provided to the NYC TLC by technology providers under the Taxicab & Livery Passenger Enhancement Programs.

1.4.2 Reproducible Research

In `R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics`, the authors have presented experimental and statistical evidence that *R Markdown* replaced the antiquated and hard-to-reproduce `copy-and-paste workflow`, and makes creating fully-reprodicible statistical analysis straight-forward (B. Baumer, Cetinkaya-Rundel, Bray, Loi, & Horton, 2014).

Reproducible research and open source are two main points of emphasis in this honors project. As scholars place more emphasis on the reproducibility of research studies, it is essential for us to make our data and code openly available for people to redo the analysis.

`Knitr` (Xie, 2018) and Github are used in this project to make the study reproducible, ranging from the initial data source to the `nyctaxi` package to the statistical data analysis. We used an **R** package called `thesisdown` to typeset this paper, this tool allows authors to create reproducible and dynamic technical report in **R** Markdown. It also allows users to embed **R** code and interactive applicationis, and output into PDF, Word, ePub, or gitbook doocuments. `thesisdown` helps users to efficiently put together any paper with similar format (Solomon, n.d.).

Github is used to store the scripts for `nyctaxi` and this thesis. `nyctaxi` is available on CRAN for people to download and install (W. P. Li, Baumer, & Trang Le, 2017), and the source code for data analysis in this thesis is available under the Github account of the author so that scholars can easily access the information that thery are interested in. In terms of tables, figures, and anything included in the Appendix attached to this thesis, scripts that are used to generate them are included in the Github repository.

1.4.3 Recommendations for taxi drivers, passengers, and TLC officials

In Chapters 3 to 5, we analyze what taxi drivers, passengers, and TLC officials want and we try to find ways for them to achieve their goals.

NYC Taxi drivers want to make the profit. Our analysis has suggested that taxi passengers are sympathetic with the drivers who have to suffer the cogestion in New York City, and pay more tips during rush hours.

Taxi passengers want the cheapest and most convenient way of transportantion. We created a Shiny App for passengers to choose a pick up zone of their interest and then decide when is the most favorable time for them to travel from that zone to any of the three airports in New York.

TLC wants to protect both taxi drivers and passengers, and it creates policies to make NYC taxi more accessible to consumers and taxi drivers enjoy their work. We suggest New York City TLC to modify the fare on rainy or snowy days to incentive taxicab drivers to pick up more trips in order to make taking a street hail vehicle on average more affordable on rainy days for passengers.

Chapter 2

Data and `nyctaxi` Package

2.1 Data and Storage

The `nyctaxi` **R** package allows users to download, clean, and load data into SQL databases. There are four types of data that are available for users to get access to: trip level yellow taxi data from 2009 to the most recent month, trip level green taxi data from August 2013 to the most recent month, Uber pick-up data from April to September 2014 and from Janaury to June 2015, and weekly-aggregated Lyft trip data from 2016 to the most recent week (W. P. Li et al., 2017).

2.1.1 Yellow Taxi

The total size of all yellow taxi trip data `csv` files (from Jan 2010 to Dec 2016) is 191.38 GB, and NYC yellow taxi trip data from Jan 2009 to the most recent month can be found on the NYC Taxi & Limousine Commission (TLC) website (N. T. Staff, 2009b). The data were collected and provided to the NYC TLC by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs

(TPEP/LPEP).

The yellow taxi trip records include the following fields: 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.

2.1.2 Green Taxi

The total size of green taxi trip data `csv` files (from Aug 2013 to Dec 2016) is 7.8 GB, and green taxi trip data from Aug 2013 to the most recent month can be downloaded from NYC Taxi & Limousine Commission (TLC). (N. T. Staff, 2009b) Green taxi trip records include the same variables as yellow taxi trip records.

2.1.3 TLC Summary Report

The New York City TLC publishes summary reports that include aggregate statistics about taxi, Uber, and Lyft usage. These are in addition to the trip-level data; although the summary reports contain much less detail, they're updated more frequently, which provides a more current glimpse into the state of the cutthroat NYC taxi market. (N. T. Staff, 2009a)

In addition, the trip level NYC Uber data only covers two periods, from April to September 2014 and from January to June 2015. However, the summary reports cover weekly-aggreagated data from 2015 to the most recent week (N. T. Staff, 2009c).

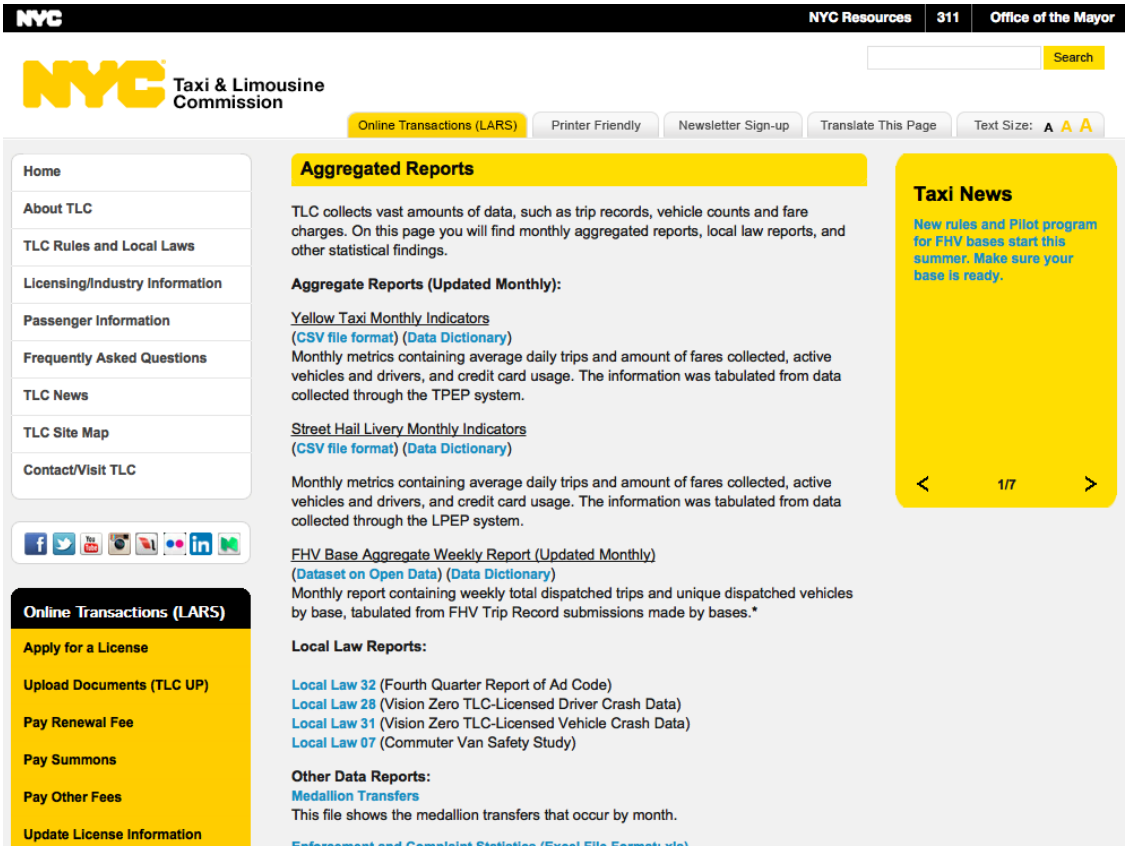


Figure 2.1: NYC Taxi and Limousine Commission Aggregated Reports

The data can be accessed by using the following commands:

- Yellow taxi data

```
download.file("http://www.nyc.gov/html/tlc/downloads/csv/data_reports_monthly_indi
```

- Uber and Lyft data

```
download.file("http://data.cityofnewyork.us/api/views/2v9c-2k7f/rows.csv?accessTyp
```

2.1.4 Uber

The total size of Uber pick-up data (from Apr to Sep 2014 and from Jan to June 2015) is 900 MB, and thanks to FiveThirtyEight (FiveThirtyEight, 2015) who obtained the

data from NYC TLC by submitting a Freedom of Information Law (FOIL) request on July 20, 2015, these data are now open to public.

The 2014 Uber data contains 4 variables: `Date/Time` (the date and time of the Uber pick-up), `Lat` (the latitude of the Uber pick-up), `Lon` (the longitude of the Uber pick-up), and `Base` (the TLC base company code affiliated with the Uber pickup).

The 2015 Uber data contains 4 variables: `Dispatching_base_num` (the TLC base company code of the base that dispatched the Uber), `Pickup_date` (the date of the Uber pick-up), `Affiliated_base_num` (the TLC base company code affiliated with the Uber pickup), and `locationID` (the pick-up location ID affiliated with the Uber pickup).

NYC Open Data also provides weekly-aggregated Uber pick-up data from 2015 to the most recent month (N. O. Staff, 2015b).

2.1.5 Lyft

The total size of weekly-aggregated Lyft trip data (from Jan 2015 to Dec 2016) is 914.9 MB, and these data are open to public and weekly-aggregated Lyft data from 2015 to the most recent week can be found on NYC OpenData website (N. O. Staff, 2015a).

2.1.6 Data Storage

The total size of all `csv` files of the four services is about 200 GB, and a laptop usually has memory less than or equal to 8 GB. Limited memory constrains the amount of data that can be loaded by a personal computer at one time. When users load data into **R** environment, **R** keeps them in memory; when the amount of data loaded into **R** environment gets close to the limit of a computer's memory, **R** becomes unresponsive

or force quit the current session. Therefore, better ways to work with data that takes more space than 8 GB is needed. Comparing to RAM, hard disk is often used to store medium-sized data, because it is affordable and are designed for storing large items permanently. However, retrieving data from hard drives is about 1,000,000 times slower.

2.2 ETL nyctaxi Package

`etl` is the parent package of `nyctaxi`. `etl` provides a framework that allows **R** users to work with medium data without any knowledge in SQL database. Users can run SQL queries by using `dplyr` commands in **R** and choose to only return the final result, which could be a summary table, from SQL database into **R** Environment in order to aovid **R** from crashing. The user interaction takes place solely within **R**.

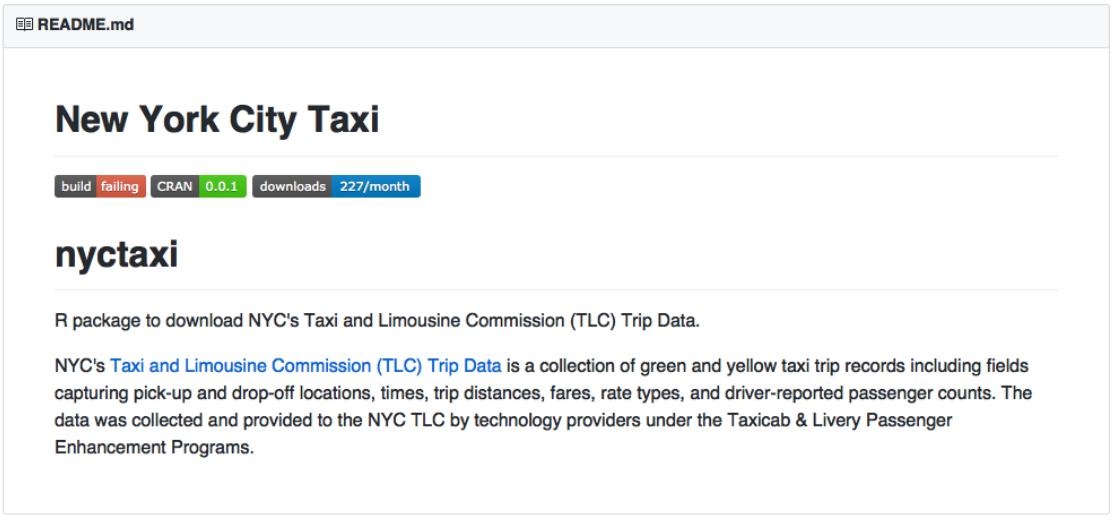
`etl` framework has three operations -Extract, Transfer, and Load- which bring real-time data into local or remote SQL databases. Users can specify which type of SQL database they prefer to connect to. `etl`-dependent packages, such as `nyctaxi`, make medium data more accessible to a wider audience (B. Baumer et al., 2014).

`nyctaxi` was initially designed to work with New York City taxi data, but later on Uber and Lyft data were added and the ETL functions are modified to be specialized in working with these data. This package compiles three major sources of hail service in New York City so that it is convenient for users to compare and contrast the performance of these three services (W. P. Li et al., 2017).

This package inherits functions from many packages: `etl`(B. S. Baumer, 2017), `dplyr` (Wickham, Francois, Henry, & Müller, 2017), `DBI` (R Special Interest Group on Databases (R-SIG-DB), Wickham, & Müller, 2018), `rlang` (Henry & Wickham, 2018), `lubridate` (Grolemund & Wickham, 2011), `leaflet` (Cheng, Karambelkar, & Xie,

2017), and `stringr` (Wickham, 2018).

Since SQL databases are good tools for medium data analysis, ETL functions build connection to a SQL database at the back end and convert **R** code automatically into SQL queries and send them to the SQL database to get data tables containing data of each hail service. Thus, users do not need to have any knowledge of SQL queries and they can draw in any subsets of the data from the SQL database in **R**.


README.md

New York City Taxi

build failing CRAN 0.0.1 downloads 227/month

nyctaxi

R package to download NYC's Taxi and Limousine Commission (TLC) Trip Data.

NYC's Taxi and Limousine Commission (TLC) Trip Data is a collection of green and yellow taxi trip records including fields capturing pick-up and drop-off locations, times, trip distances, fares, rate types, and driver-reported passenger counts. The data was collected and provided to the NYC TLC by technology providers under the Taxicab & Livery Passenger Enhancement Programs.

Figure 2.2: 'nyctaxi' package GitHub Repository

In general, `etl_extract.etl_nyctaxi()` function downloads data of the four types of hail service data (yellow taxi, green taxi, Uber, and Lyft) from the corresponding sources. `etl_transform.etl_nyctaxi()` uses different techniques to clean all four types of data to get then ready for the next step. `etl_load.etl_nyctaxi()` loads the data user selected to a SQL database.

The Comprehensive **R** Archive Network (CRAN) is a collection of sites that carry identical material, consisting of the **R** distributions, the contributed extensions, documentation for **R**, and binaries (C. Staff, n.d.). `nyctaxi` **R** package lives on CRAN, and it can be installed with the `install.packages()` function in **R**.

```
install.packages("nyctaxi")
```

Users need to create an `etl` object in order to apply the etl operations to it, and only the name of the SQL database, working directory, and type of SQL database need to be specified during initialization. If the type of SQL database is not specified, a local RSQLite database will be generated as default.

```
db <- src_mysql("nyctaxi", user = "urname", host = "host",
                password = "pw")
taxi <- etl("nyctaxi", dir = "~/Desktop/nyctaxi", db)
```

In the example above, a folder called `nyctaxi` is created on the desktop and a connection to a MySQL database is generated. In the procession of initialization, two subfolders, `raw` and `load`, are also created under the directory the user specifies. The `raw` folder stores data downloaded from online open sources, and and `load` folder stores cleaned CSV data files that are ready to be loaded into SQL database. The ETL framework keeps data directly scraped from online data sources in their original forms. In this way, the original data is always available to users in case data corruption happens in later stages.

After an `etl` object is created (`taxi` is the `etl` object in this case), four parameters are needed to specify the data that users want: (1) `obj`: an etl object; (2) `years`: a numeric vector giving the years, and the default is the most recent year; (3) `months`: a numeric vector giving the months, and the default is `1:12`; (4) `type`: a character variable giving the type of data the user wants to download. There are four types: `yellow`, `green`, `uber`, and `lyft`. The default is `yellow`.

2.2.1 Taxi zone shapefile attached to nyctaxi R package

Two datasets are attached to `nyctaxi`. The first one is called `taxi_zone_lookup`, and this dataset contains information, such as taxi zone location IDs, location names, and corresponding boroughs for each ID (N. T. Staff, 2009b). A shapefile containing the boundaries for the taxi zones, `taxi_zones`, is also included in the package for users to do spatial analysis. This shapefile is publicly accessible on NYC TLC's website (N. T. Staff, 2009b).

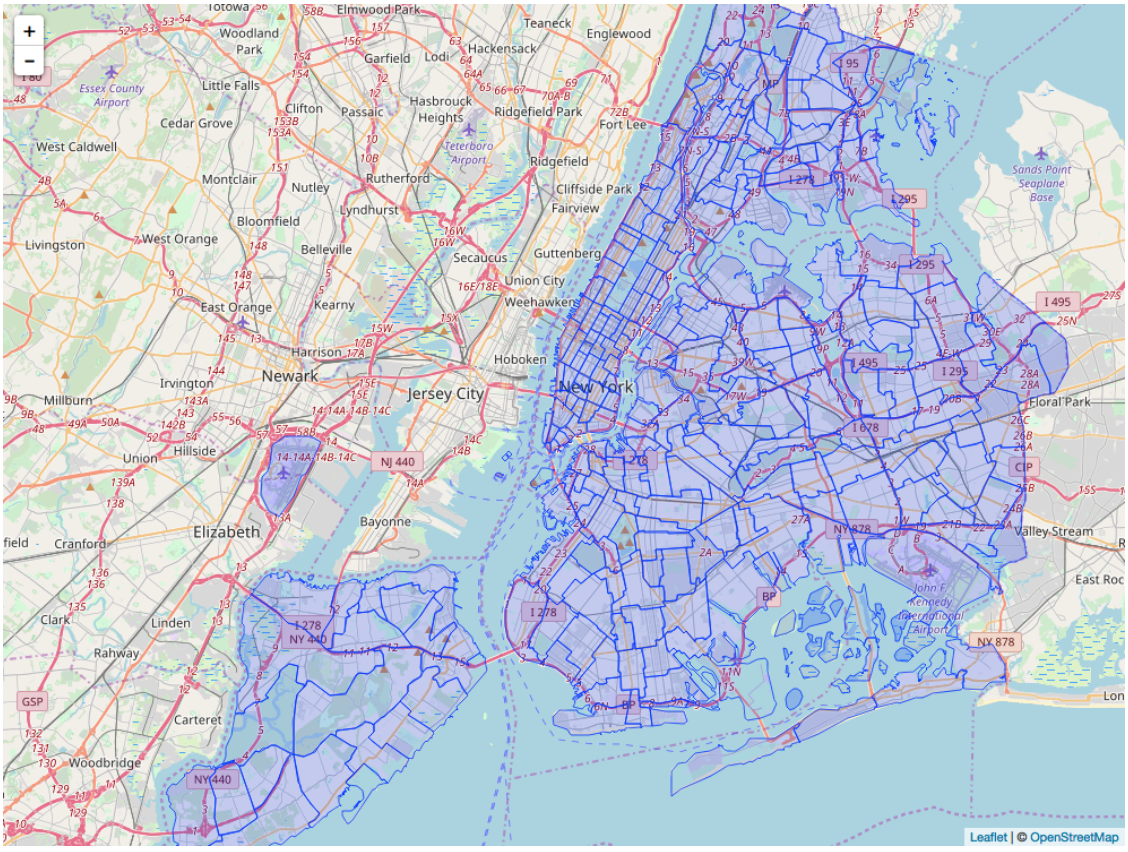


Figure 2.3: NYC Taxi Zone Map

Visulizations similar to Figure 2.3 can be generated with the shapefile.

2.3 Extract-Transform-Load

2.3.1 Extract

`etl_extract.etl_nyctaxi()` allows users to download New York City yellow taxi, green taxi, Uber, and Lyft data from the corresponding data sources. It takes the `years`, `months`, and `type` parameters and download the New York City taxi data specified by users. New York City Yellow and Green Taxi data are updated on NYC Taxi & Limousine Commission (TLC) website on a monthly basis.

```
taxi %>%
   etl_extract(years = 2014:2016, months = 1:12,
               type = c("yellow", "green"))
```

Uber trip record data is static and small, so we decided to only give users the options to either download all data from April to Sepetember, 2014 or download all Uber trip records from Janaury to June, 2015 at once. Users do not have the ability to download Uber data from a specific month.

```
taxi %>%
   etl_extract(years = 2014:2016, months = 1:12,
               type = c("uber"))
```

Lyft data is updated on NYC Open Data website on a weekly basis. Since the weekly-aggregated data is tiny and only data later than 2014 is available, we decided to only allow users to download Lyft data by year.

```
taxi %>%
   etl_extract(years = 2014:2016, months = 1:12,
               type = c("lyft"))
```

The default `years` is the current year, and the default `months` are the all twelve months. The default type of transportation is `yellow`. When an invalid month is entered, warning message will suggest users to reconsider their choice and select a new set of month.

An utility function, `download_nyc_data()`, was written to be used in `etl_extract.etl_nyctaxi()` to make this function more concise (Appendix A).

2.3.2 Transform

`etl_transform.etl_nyctaxi()` allows users to transform New York City yellow taxi, green taxi, Uber, and Lyft data into cleaned formats, and it utilizes different data cleaning techniques when it transforms data for each transportation type. In general, it cleans the data and creates a new `csv` file in the `load` directory to store the cleaned data. It helps us to retain and protect raw data from being modified or destroyed. Users are allowed to specify the month of interest in order to only transform the data that they are interested in. This functionality helps people to be more efficient with their use of time.

By default, it takes the current year yellow taxi trip records data files, and saves copies of them in the `load` diectory. It skips the cleaning step, because the raw yellow taxi data downloaded from TLC is already in a desired format with all variables correctly labelled.

```
taxi %>%
   etl_transform(years = 2014:2016, months = 1:12,
                 type = c("yellow", "green", "uber", "lyft"))
```

There are a few main transformations that are done by this function:

Green Taxi – Extra Blank Row and Column

Green Taxi monthly data from August 2013 to the most recent month besides 2015 all have a blank second row in the `csv` files. Similar to this problem, Green Taxi data from 2013, 2014, and 2015 all have an extra blank columns attanched to the right-most column. These blank rows and columns cause problems in the later stage when users want to load data into SQL database. In order to get Green Taxi data ready for the `load` phase, we used the `system()` function in **R** to invoke the `cut` Terminal command specified to remove the blank rows and columns.

Uber Data – Reconciling Inconsistent Filenames

Uber only released over 4.5 million data records from April to September 2014 and 14.3 million records from Janaury to June 2015. Information of different sets of variables are released for 2014 and 2015, and variables have different naming convention. When users want to download data from both years, variables are renamed so that data from both years can be consolidated into one big dataset with consistent variable names.

Uber Data – Reconciling Inconsistent Data Formats

The data type of `Date/Time` variable in Uber datasets is originally encoded as `character`. In order to enable it to be recognized as `timestamp` by **R**, we use `ymd_hms` in `lubridate` (Grolemund & Wickham, 2011) to transform date time to `POSIXct` objects.

Optimizing I/O Process

Improving file input and output processes is an important part of `etl_transform`. `data.table` (Dowle & Srinivasan, 2017) only takes half of the time to read from and write into datasets comparing to `readr` (Wickham, Hester, & Francois, 2017). Therefore, `etl_transform` uses `fread()` and `fwrite()` from `data.table` instead of `read_csv` or `write_csv` from `readr` to reduce the data processing time (Zhang, 2017).

2.3.3 Load

`etl_load.etl_nyctaxi()` allows users to load New York City yellow taxi, green taxi, Uber, and Lyft data into different data tables in a SQL database. It populates a SQL database with data cleaned by `etl_transform`.

```
taxi %>%
   etl_load(years = 2014:2016, months = 1:12,
            type = c("yellow", "green", "uber", "lyft"))
```

2.3.4 SQL Database Initialization

`init.mysql` is written under `nyctaxi` to help users to set up five basic table structures for MySQL database. `yellow_old` is created for Yellow Taxi data that are prior to August 2016, and `yellow` is created for data later than July 2016. `green`, `uber`, and `lyft` are also initiated for the three transportations.

`etl_init()` can be run after a database connection is built to process to process `init.mysql` to initialize a MySQL database, and default columns with the correct variable names and types defined will be automatically generated.

```
taxi %>%

  etl_init()
```

In order to increase the query speed at the data analysis stage, `KEY`s are created for multiple variables for each transportation. Since there is no variable containing unique value for each observation, no primary variable is needed. Using `KEY`s in data analysis query can speed up the query process.

Due to the large size of Yellow Taxi datasets, `yellow_old` and `yellow` are partitioned into subgroups by `year`. When we need to run a query on data from a specific year, having partitions allows MySQL to directly find the data specified without filtering on every single row. It speeds up the query process.

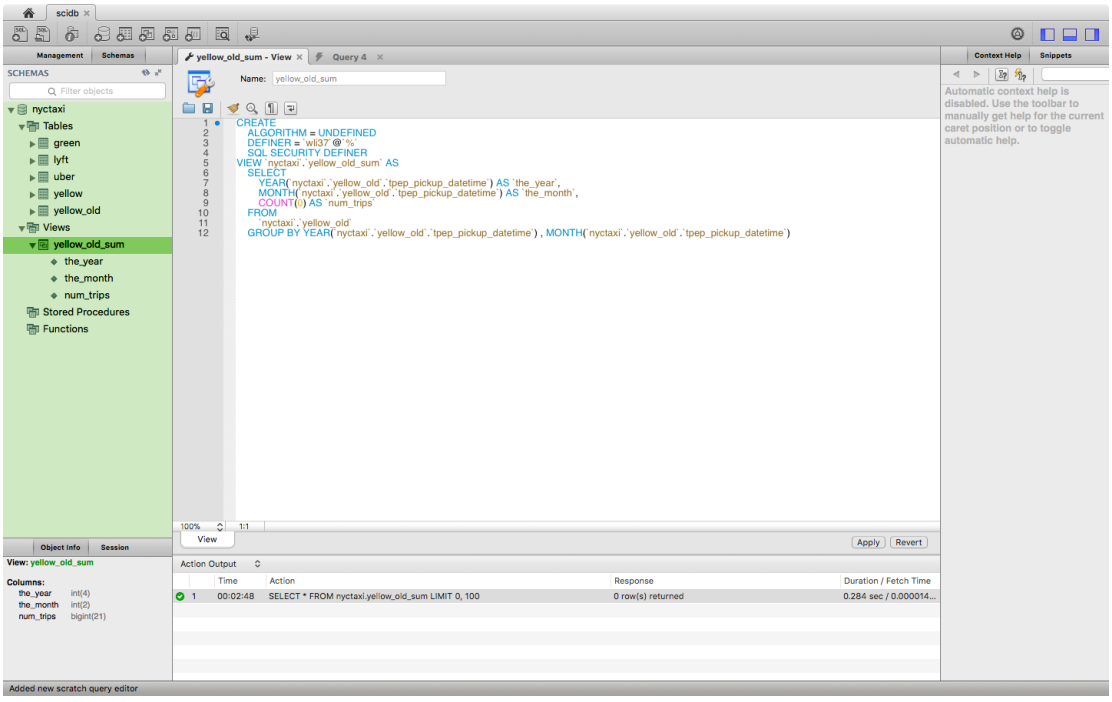


Figure 2.4: MySQL View

A `VIEW` called `yellow_old_sum` is also created to generate a summary table for the number of Yellow Taxi trips in each month.

2.4 New York City Taxicab and E-hail Services Summary

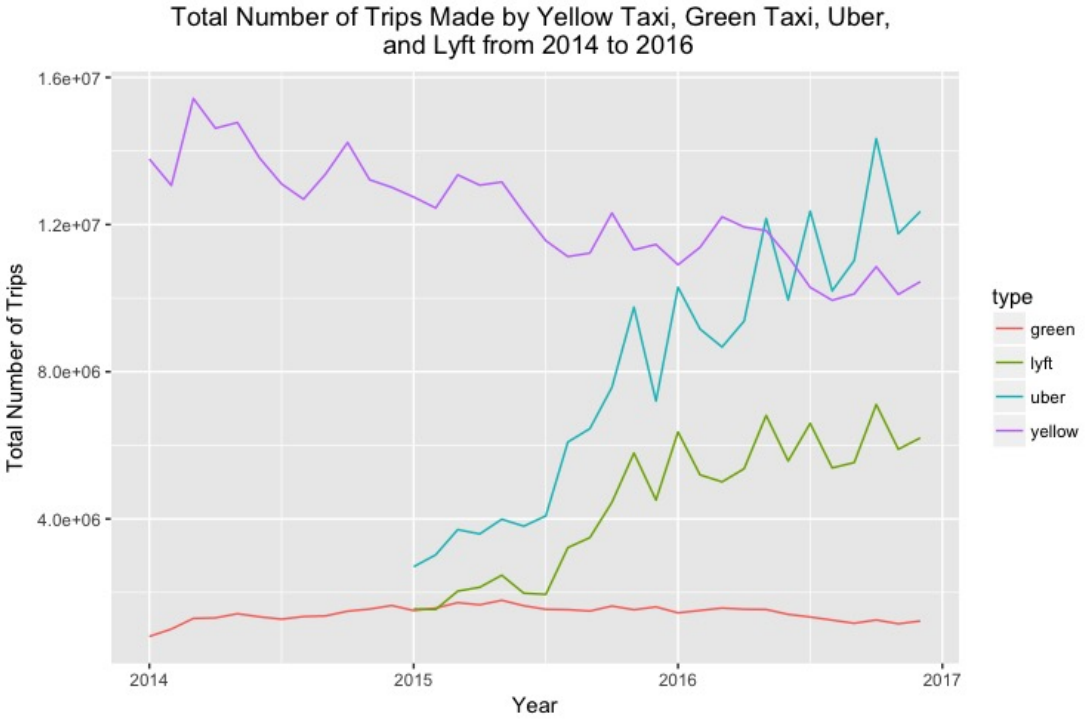


Figure 2.5: Summary of Number of trips Made by 4 Types of Transportations between 2014 and 2016 in NYC

Figure 2.5 is a summary of total number of trips made by all 4 types of transporations that are available to users from 2014 to 2016. In order to generate this summary, I combined trip-level yellow and green taxi data from TLC website (N. T. Staff, 2009b) and weekly Uber (N. O. Staff, 2015b) and Lyft data from NYC OpenData (N. O. Staff, 2015a). Trip-level Uber data that can be accessed by using the `nyctaxi` package is not used in the creation of Figure 2.5 because of the limited ranges of dates where the Uber trip-level data is available.

Data used in Figure 2.5 can be accessed by running the code below:

- Yellow and green taxi trip data

```
taxi %>%
  etl_extract(years = 2014:2016, months = 1:12, type = c("yellow", "green")) %>%
  etl_transform(years = 2014:2016, months = 1:12, type = c("yellow", "green")) %>%
  etl_load(years = 2014:2016, months = 1:12, type = c("yellow", "green"))
```

- Uber weekly data

```
download.file("https://data.cityofnewyork.us/resource/gt3n-7ri6.csv",
              destfile = "~/Desktop/uber_weekly_data.csv")
```

- Lyft weekly data

```
taxi %>%
  etl_extract(years = 2014:2016, months = 1:12, type = "lyft") %>%
  etl_transform(years = 2014:2016, months = 1:12, type = "lyft") %>%
  etl_load(years = 2014:2016, months = 1:12, type = "lyft")
```

2.5 Source Code

2.5.1 ETL Extract

```
etl_extract.etl_nyctaxi <-
  function(obj,
years = as.numeric(format(Sys.Date(),'%Y')),
                                    months = 1:12,
                                    type  = "yellow",...) {
  #TAXI YELLOW-------------------------
  taxi_yellow <- function(obj, years, months,...) {
    message("Extracting raw yellow taxi data...")
```

```
  remote <- etl::valid_year_month(years, months,
  begin = "2009-01-01") %>%
    mutate_(src =
~file.path("https://s3.amazonaws.com/nyc-tlc/trip+data",
                              paste0("yellow", "_tripdata_", year, "-",
                    stringr::str_pad(month, 2, "left", "0"), ".csv")))
  tryCatch(expr = etl::smart_download(obj, remote$src, ...),
           error = function(e){warning(e)},
           finally = warning("Only the following data are availabel on
                              TLC: Yellow taxi data: 2009 Jan -
                              last month"))}
#TAXI GREEN----------------------
taxi_green <- function(obj, years, months,...) {
  message("Extracting raw green taxi data...")
  remote <- etl::valid_year_month(years, months, begin = "2013-08-01") %>%
    mutate_(src =
              ~file.path("https://s3.amazonaws.com/nyc-tlc/trip+data",
                              paste0("green", "_tripdata_", year, "-",
                          stringr::str_pad(month, 2, "left", "0"), ".csv")))
  tryCatch(expr = etl::smart_download(obj, remote$src, ...),
           error = function(e){warning(e)},
           finally = warning("Only the following data are availabel on TLC:
                              Green taxi data: 2013 Aug - last month"))}
#UBER--------------------------------
uber <- function(obj, years, months,...) {
  message("Extracting raw uber data...")
  raw_month_2014 <- etl::valid_year_month(years = 2014, months = 4:9)
```

```
raw_month_2015 <- etl::valid_year_month(years = 2015, months = 1:6)
raw_month <- bind_rows(raw_month_2014, raw_month_2015)
path = "https://raw.githubusercontent.com/
fivethirtyeight/uber-tlc-foil-response/master/uber-trip-data"
remote <- etl::valid_year_month(years, months)
remote_small <- intersect(raw_month, remote)
if (2015 %in% remote_small$year && !(2014 %in% remote_small$year)){
  #download 2015 data
  message("Downloading Uber 2015 data...")
  etl::smart_download(obj, "https://github.com/fivethirtyeight/
                      uber-tlc-foil-response/raw/master/
                  uber-trip-data/uber-raw-data-janjune-15.csv.zip",...)}
else if (2015 %in% remote_small$year && 2014 %in% remote_small$year) {
  #download 2015 data
  message("Downloading Uber 2015 data...")
  etl::smart_download(obj, "https://github.com/fivethirtyeight/
                    uber-tlc-foil-response/raw/master/uber-trip-data
                      /uber-raw-data-janjune-15.csv.zip",...)
  #download 2014 data
  small <- remote_small %>%
    filter_(~year == 2014) %>%
    mutate_(month_abb = ~tolower(month.abb[month]),
            src = ~file.path(path,
            paste0("uber-raw-data-",month_abb,
            substr(year,3,4),".csv")))
  message("Downloading Uber 2014 data...")
  etl::smart_download(obj, small$src,...)
```

```
  } else if (2014 %in% remote_small$year &&
  !(2015 %in% remote_small$year)) {
    message("Downloading Uber 2014 data...")
    #file paths
    small <- remote_small %>%
      mutate_(month_abb =
                ~tolower(month.abb[month]),
              src = ~file.path(path,
              paste0("uber-raw-data-",month_abb,
              substr(year,3,4),".csv")))
    etl::smart_download(obj, small$src,...)}
  else {warning("The Uber data you requested are
                not currently available. Only data
                from 2014/04-2014/09 and 2015/01-
                2015/06 are available...")}
  }
#LYFT---------------------------------
lyft <- function(obj, years, months,...){
  message("Extracting raw lyft data...")
  #check if the week is valid
  valid_months <- etl::valid_year_month(years, months,
  begin = "2015-01-01")
  base_url = "https://data.cityofnewyork.us/
  resource/edp9-qgv4.csv"
  valid_months <- valid_months %>%
    mutate_(new_filenames =
              ~paste0("lyft-", year, ".csv")) %>%
```

```
    mutate_(drop = TRUE)
  #only keep one data set per year
  year <- valid_months[1,1]
  n <- nrow(valid_months)
  for (i in 2:n) {
    if(year == valid_months[i-1,1]) {
      valid_months[i,6] <- FALSE
      year <- valid_months[i+1,1]
    } else {
      valid_months[i,6] <- TRUE
      year <- valid_months[i+1,1]}
    }
  row_to_keep = valid_months$drop
  valid_months <- valid_months[row_to_keep,]

  #download lyft files, try two different methods
  first_try<-tryCatch(
    download_nyc_data(obj, base_url, valid_months$year,
    n = 50000, names = valid_months$new_filenames),
    error = function(e){warning(e)},
    finally = 'method = "libcurl" fails')
}

if (type == "yellow"){taxi_yellow(obj, years, months,...)}
else if (type == "green"){taxi_green(obj, years, months,...)}
else if (type == "uber"){uber(obj, years, months,...)}
else if (type == "lyft"){lyft(obj, years, months,...)}
```

```
  else {message("The type you chose does not exit...")}


  invisible(obj)
}
```

2.5.2 ETL Transform

```
opts_chunk$set(tidy.opts=list(width.cutoff=60))
etl_transform.etl_nyctaxi <- function(obj,
                          years = as.numeric(format(Sys.Date(),'%Y')),
                          months = 1:12,
                          type  = "yellow",...) {
  #TAXI YELLOW-----------------------------
  taxi_yellow <- function(obj, years, months) {
    message("Transforming yellow taxi data from raw to
            load directory...")
    #create a df of file path of the files that the user wants to transform
    remote <- etl::valid_year_month(years, months,
    begin = "2009-01-01") %>%
      mutate_(src = ~file.path(attr(obj, "raw_dir"),
      paste0("yellow", "_tripdata_", year, "-",
      stringr::str_pad(month, 2, "left", "0"), ".csv")))
    #create a df of file path of the files that are in the raw directory
    src <- list.files(attr(obj, "raw_dir"), "yellow", full.names = TRUE)
    src_small <- intersect(src, remote$src)
    #Move the files
    in_raw <- basename(src_small)
```

```
  in_load <- basename(list.files(attr(obj, "load_dir"), "yellow",
  full.names = TRUE))
  file_remian <- setdiff(in_raw,in_load)
  file.copy(file.path(attr(obj, "raw_dir"),file_remian),
            file.path(attr(obj, "load_dir"),file_remian) )}
#TAXI GREEN-----------------------------
taxi_green <- function(obj, years, months) {
  message("Transforming green taxi data from raw
          to load directory...")
  #create a df of file path of the files that the user wants to transform
  remote <- etl::valid_year_month(years, months,
  begin = "2013-08-01") %>%
    mutate_(src = ~file.path(attr(obj, "raw_dir"),
    paste0("green", "_tripdata_", year, "-",
    stringr::str_pad(month, 2, "left", "0"), ".csv")))
  #create a df of file path of the files that are in the raw directory
  src <- list.files(attr(obj, "raw_dir"), "green", full.names = TRUE)
  src_small <- intersect(src, remote$src)
  #Clean the green taxi data files
  #get rid of 2nd blank row
  if (length(src_small) == 0){
    message("The files you requested are not available
            in the raw directory.")
  } else{
    #a list of the ones that have a 2nd blank row
    remote_green_1 <- remote %>% filter_(~year != 2015)
    src_small_green_1 <- intersect(src, remote_green_1$src)
```

```
# check that the sys support command line,
#and then remove the blank 2nd row
if(length(src_small_green_1) != 0) {
  if (.Platform$OS.type == "unix"){
    cmds_1 <- paste("sed -i -e '2d'", src_small_green_1)
    lapply(cmds_1, system)
  } else {
    message("Windows system does not
    currently support removing the 2nd blank row
    in the green taxi datasets. This might affect
    loading data into SQL...")}
  }else {
    "You did not request for any
    green taxi data, or all the green
    taxi data you requested are cleaned."}
#fix column number
remote_green_2 <- remote %>%
  filter_(~year %in% c(2013, 2014, 2015)) %>%
  mutate_(keep =
            ~ifelse(year %in% c(2013,2014), 20,21),
          new_file =
            ~paste0("green_tripdata_", year, "_",
                stringr::str_pad(month, 2, "left", "0"),
                              ".csv"))
src_small_green_2 <- intersect(src, remote_green_2$src)
src_small_green_2_df <- data.frame(src_small_green_2)
names(src_small_green_2_df) <- "src"
```

```
src_small_green_2_df <- inner_join(src_small_green_2_df,
remote_green_2, by = "src")
src_small_green_2_df <- src_small_green_2_df %>%
  mutate(cmds_2 = paste("cut -d, -f1-", keep," ",src, " > ",
  attr(obj, "raw_dir"),"/green_tripdata_",
  year, "_", stringr::str_pad(month, 2, "left", "0"),".csv",
  sep = ""))
#remove the extra column
if(length(src_small_green_2) != 0) {
  if (.Platform$OS.type == "unix"){
    lapply(src_small_green_2_df$cmds_2, system)}
  else {
    message("Windows system does not currently
    support removing the 2nd blank row
    in the green taxi datasets. This might
    affect loading data into SQL...")}
  }else {
    "All the green taxi data you
    requested are in cleaned formats."}
#Find the files paths of the files that need to be transformed
file.rename(file.path(dirname(src_small_green_2_df$src),
                      src_small_green_2_df$new_file),
            file.path(attr(obj, "load_dir"),
            basename(src_small_green_2_df$src)))
#Move the files
in_raw <- basename(src_small)
in_load <- basename(list.files(attr(obj, "load_dir"),
```

```
    "green", full.names = TRUE))
    file_remian <- setdiff(in_raw,in_load)
    file.copy(file.path(attr(obj, "raw_dir"),file_remian),
    file.path(attr(obj, "load_dir"),file_remian) )}}
#UBER--------------------------------
uber <- function(obj) {
  message("Transforming uber data from raw to load directory...")
  #creat a list of 2014 uber data file directory
  uber14_list <- list.files(path = attr(obj, "raw_dir"),
  pattern = "14.csv")
  uber14_list <- data.frame(uber14_list)
  uber14_list <- uber14_list %>% mutate_(file_path =
  ~file.path(attr(obj, "raw_dir"), uber14_list))
  uber14file <- lapply(uber14_list$file_path, readr::read_csv)
  n <- length(uber14file)
  if (n == 1) {
    uber14 <- data.frame(uber14file[1])
  } else if (n == 2) {
    uber14 <- bind_rows(uber14file[1], uber14file[2])
  } else if (n > 2) {
    uber14 <- bind_rows(uber14file[1], uber14file[2])
    for (i in 3:n){uber14 <- bind_rows(uber14, uber14file[i])}
  }
  substrRight <- function(x, n){substr(x, nchar(x)-n+1, nchar(x))}
  uber14_datetime <- uber14 %>%
    mutate(date = gsub( " .*$", "", `Date/Time`),
    len_date = nchar(date),
```

```
          time = sub('.*\\ ', '', `Date/Time`))
uber14_datetime <- uber14_datetime %>%
  mutate(month =
           substr(`Date/Time`, 1, 1),
         day = ifelse(len_date == 8,
         substr(`Date/Time`, 3,3),substr(`Date/Time`, 3,4)),
         pickup_date =
           lubridate::ymd_hms(paste0("2014-", month, "-",
                                     day, " ", time)))
uber14_df <- uber14_datetime[-c(1,5:9)]

#2015
zipped_uberfileURL <- file.path(attr(obj, "raw_dir"),
"uber-raw-data-janjune-15.csv.zip")
raw_month_2015 <- etl::valid_year_month(years = 2015, months = 1:6)
remote_2015 <- etl::valid_year_month(years, months)
remote_small_2015 <- inner_join(raw_month_2015, remote_2015)
if(file.exists(zipped_uberfileURL) &&
   nrow(remote_small_2015) != 0){
  utils::unzip(zipfile = zipped_uberfileURL,unzip = "internal",
  exdir = file.path(tempdir(), "uber-raw-data-janjune-15.csv.zip"))
  uber15 <- readr::read_csv(file.path(tempdir(),
  "uber-raw-data-janjune-15.csv.zip",
  "uber-raw-data-janjune-15.csv"))}

names(uber14_df) <- c("lat", "lon", "affiliated_base_num",
"pickup_date")
```

```
  names(uber15) <- tolower(names(uber15))
  uber <- bind_rows(uber14_df, uber15)
  utils::write.csv(uber, file.path(tempdir() ,"uber.csv"))
  if(nrow(uber) != 0) {
    if (.Platform$OS.type == "unix"){cmds_3 <-
    paste("cut -d, -f2-7",file.path(tempdir(),"uber.csv"), " > ",
    file.path(attr(obj, "load_dir"),"uber.csv"))
      lapply(cmds_3, system)
    } else {
      message("Windows system does not currently
      support removing the 2nd blank row
      in the green taxi datasets. This might
      affect loading data into SQL...")}
    }else {
      "You did not request for any
      green taxi data, or all the green
      taxi data you requested are cleaned."}
  }
#LYFT---------------------------------
lyft <- function(obj, years, months){
  valid_months <- etl::valid_year_month(years, months = 1,
  begin = "2015-01-01")
  message("Transforming lyft data from raw to load directory...")
  src <- list.files(attr(obj, "raw_dir"), "lyft", full.names = TRUE)
  src_year <- valid_months %>% distinct_(~year)
  remote <- data_frame(src)
  remote <- remote %>%
```

```
      mutate_(lcl = ~file.path(attr(obj, "load_dir"),basename(src)),
              basename = ~basename(src), year = ~substr(basename,6,9))
    class(remote$year) <- "numeric"
    remote <- inner_join(remote,src_year, by = "year" )
    for(i in 1:nrow(remote)) {
        datafile <- readr::read_csv(remote$src[i])
        readr::write_delim(datafile, path = remote$lcl[i],
        delim = "|", na = "")}}

  #transform the data from raw to load
  if (type == "yellow"){taxi_yellow(obj, years, months)}
  else if (type == "green"){taxi_green(obj, years, months)}
  else if (type == "uber"){uber(obj)}
  else if (type == "lyft"){lyft(obj, years, months)}
  else {message("The type you chose does not exit...")}

  invisible(obj)
}
```

2.5.3 ETL Load

```
opts_chunk$set(tidy.opts=list(width.cutoff=60))
etl_load.etl_nyctaxi <- function(obj,
 years = as.numeric(format(Sys.Date(),'%Y')),
                                  months = 1:12,
                                  type  = "yellow", ...) {
  #TAXI YELLOW----------------------------
```

```
taxi_yellow <- function(obj, years, months,...) {
  #create a df of file path of the files that are in the load directory
  src <- list.files(attr(obj, "load_dir"), "yellow",
  full.names = TRUE)
  src <- data.frame(src)

  #files before 2016-07
  remote_old <- etl::valid_year_month(years, months,
  begin = "2009-01-01", end = "2016-06-30") %>%
    mutate_(src = ~file.path(attr(obj, "load_dir"),
    paste0("yellow", "_tripdata_", year, "-",
    stringr::str_pad(month, 2, "left", "0"), ".csv")))
  src_small_old <- inner_join(remote_old, src, by = "src")
  #files later then 2017-06
  remote_new <- etl::valid_year_month(years, months,
  begin = "2016-07-01") %>%
    mutate_(src =  ~file.path(attr(obj, "load_dir"),
    paste0("yellow", "_tripdata_", year, "-",
    stringr::str_pad(month, 2, "left", "0"), ".csv")))
  src_small_new <- inner_join(remote_new, src, by = "src")
  #data earlier than 2016-07
  if(nrow(src_small_old) == 0) {
    message("The taxi files (earlier than 2016-07)
            you requested are not available in
            the load directory...")
  } else {
    message("Loading taxi data from
```

```
            load directory to a sql database...")
    mapply(DBI::dbWriteTable,
           name = "yellow_old", value = src_small_old$src,
           MoreArgs =
             list(conn = obj$con, append = TRUE))}


  #data later then 2016-06
  if(nrow(src_small_new) == 0) {
    message("The new taxi files (later than 2016-06)
            you requested are not available in the
            load directory...")
  } else {
    message("Loading taxi data from load
            directory to a sql database...")
    mapply(DBI::dbWriteTable,
           name = "yellow", value = src_small_new$src,
           MoreArgs =
             list(conn = obj$con, append = TRUE))}


  }
#TAXI GREEN----------------------------
taxi_green <- function(obj, years, months,...) {
  #create a list of file that the user wants to load
  remote <- etl::valid_year_month(years, months,
  begin = "2013-08-01") %>%
    mutate_(src = ~file.path(attr(obj, "load_dir"),
    paste0("green", "_tripdata_", year, "-",
```

```
    stringr::str_pad(month, 2, "left", "0"), ".csv")))
  #create a df of file path of the files that are in the load directory
  src <- list.files(attr(obj, "load_dir"), "tripdata",
  full.names = TRUE)
  src <- data.frame(src)
  #only keep the files thst the user wants to transform
  src_small <- inner_join(remote, src, by = "src")
  if(nrow(src_small) == 0) {
    message("The taxi files you requested
            are not available in the
            load directory...")
  } else {
    message("Loading taxi data from
            load directory to a sql database...")
    mapply(DBI::dbWriteTable,
           name = "green", value = src_small$src,
           MoreArgs =
             list(conn = obj$con, append = TRUE, ... = ...))}}
#UBER---------------------------------
uber <- function(obj,...) {
  uberfileURL <- file.path(attr(obj, "load_dir"), "uber.csv")
  if(file.exists(uberfileURL)) {
    message("Loading uber data from
            load directory to a sql database...")
    DBI::dbWriteTable(conn = obj$con, name = "uber",
    value = uberfileURL, append = TRUE, ... = ...)
  } else {
```

```
    message("There is no uber data
            in the load directory...")}}
#LYFT--------------------------------
lyft <- function(obj, years, months,...){
  message("Loading lyft data from
          load directory to a sql database...")
  #create a list of file that the user wants to load
  valid_months <- etl::valid_year_month(years, months,
  begin = "2015-01-01")
  src <- list.files(attr(obj, "load_dir"), "lyft",
  full.names = TRUE)
  src_year <- valid_months %>% distinct_(~year)
  remote <- data_frame(src)
  remote <- remote %>% mutate_(tablename = ~"lyft",
  year =~substr(basename(src),6,9))
  class(remote$year) <- "numeric"
  remote <- inner_join(remote,src_year, by = "year" )
  if(nrow(remote) != 0) {
    write_data <- function(...) {
      lapply(remote$src, FUN = DBI::dbWriteTable,
      conn = obj$con, name = "lyft", append = TRUE,
      sep = "|", ... = ...)}
    write_data(...)
  } else {
    message("The lyft files you requested
            are not available in the
            load directory...")}}
```

```
  if (type == "yellow"){taxi_yellow(obj, years, months,...)
  }else if (type == "green"){taxi_green(obj, years, months,...)
  }else if (type == "uber"){uber(obj,...)
  }else if (type == "lyft"){lyft(obj, years, months,...)
  }else {message("The type you chose does not exit...")
            }

  invisible(obj)
}
```

2.5.4 ETL Init

```
DROP TABLE IF EXISTS `yellow_old`;

CREATE TABLE `yellow_old` (
 `VendorID` tinyint DEFAULT NULL,
 `tpep_pickup_datetime` DATETIME NOT NULL,
 `tpep_dropoff_datetime` DATETIME NOT NULL,
 `passenger_count` tinyint DEFAULT NULL,
 `trip_distance` float(10,2) DEFAULT NULL,
 `pickup_longitude` double(7,5) DEFAULT NULL,
 `pickup_latitude` double(7,5) DEFAULT NULL,
 `RatecodeID` tinyint DEFAULT NULL,
 `store_and_fwd_flag` varchar(10) COLLATE latin1_general_ci DEFAULT NULL,
 `dropoff_longitude` double(7,5) DEFAULT NULL,
 `dropoff_latitude` double(7,5) DEFAULT NULL,
```

```
 `payment_type` tinyint DEFAULT NULL,
 `fare_amount` decimal(5,3) DEFAULT NULL,
 `extra` decimal(5,3) DEFAULT NULL,
 `mta_tax` decimal(5,3) DEFAULT NULL,
 `tip_amount` decimal(5,3) DEFAULT NULL,
 `tolls_amount` decimal(5,3) DEFAULT NULL,
 `improvement_surcharge` decimal(5,3) DEFAULT NULL,
 `total_amount` decimal(5,3) DEFAULT NULL,
 KEY `VendorID` (`VendorID`),
 KEY `pickup_datetime` (`tpep_pickup_datetime`),
 KEY `dropoff_datetime` (`tpep_dropoff_datetime`),
 KEY `pickup_longitude` (`pickup_longitude`),
 KEY `pickup_latitude` (`pickup_latitude`),
 KEY `dropoff_longitude` (`dropoff_longitude`),
 KEY `dropoff_latitude` (`dropoff_latitude`)
)
PARTITION BY RANGE( YEAR(tpep_pickup_datetime) ) (
  PARTITION p09 VALUES LESS THAN (2010),
  PARTITION p10 VALUES LESS THAN (2011),
  PARTITION p11 VALUES LESS THAN (2012),
  PARTITION p12 VALUES LESS THAN (2013),
  PARTITION p13 VALUES LESS THAN (2014),
  PARTITION p14 VALUES LESS THAN (2015),
  PARTITION p15 VALUES LESS THAN (2016),
  PARTITION p16 VALUES LESS THAN (2017)
);
```

```
DROP TABLE IF EXISTS `yellow`;


CREATE TABLE `yellow` (
 `VendorID` tinyint DEFAULT NULL,
 `tpep_pickup_datetime` DATETIME NOT NULL,
 `tpep_dropoff_datetime` DATETIME NOT NULL,
 `passenger_count` tinyint DEFAULT NULL,
 `trip_distance` float(10,2) DEFAULT NULL,
 `RatecodeID` tinyint DEFAULT NULL,
 `store_and_fwd_flag` varchar(10) COLLATE latin1_general_ci DEFAULT NULL,
 `PULocationID` tinyint DEFAULT NULL,
 `DOLocationID` tinyint DEFAULT NULL,
 `payment_type` tinyint DEFAULT NULL,
 `fare_amount` decimal(5,3) DEFAULT NULL,
 `extra` decimal(5,3) DEFAULT NULL,
 `mta_tax` decimal(5,3) DEFAULT NULL,
 `tip_amount` decimal(5,3) DEFAULT NULL,
 `tolls_amount` decimal(5,3) DEFAULT NULL,
 `improvement_surcharge` decimal(5,3) DEFAULT NULL,
 `total_amount` decimal(5,3) DEFAULT NULL,
 KEY `VendorID` (`VendorID`),
 KEY `pickup_datetime` (`tpep_pickup_datetime`),
 KEY `dropoff_datetime` (`tpep_dropoff_datetime`),
 KEY `PULocationID` (`PULocationID`),
 KEY `DOLocationID` (`DOLocationID`)
)
PARTITION BY RANGE( YEAR(tpep_pickup_datetime) ) (
```

```
  PARTITION p16 VALUES LESS THAN (2017),
  PARTITION p17 VALUES LESS THAN (2018)
);



DROP TABLE IF EXISTS `green`;


CREATE TABLE `green` (
 `VendorID` tinyint DEFAULT NULL,
 `lpep_pickup_datetime` DATETIME NOT NULL,
 `Lpep_dropoff_datetime` DATETIME NOT NULL,
 `Store_and_fwd_flag` varchar(10) COLLATE latin1_general_ci DEFAULT NULL,
 `RatecodeID` tinyint DEFAULT NULL,
 `Pickup_longitude` double(7,5) DEFAULT NULL,
 `Pickup_latitude` double(7,5) DEFAULT NULL,
 `Dropoff_longitude` double(7,5) DEFAULT NULL,
 `Dropoff_latitude` double(7,5) DEFAULT NULL,
 `Passenger_count` tinyint DEFAULT NULL,
 `Trip_distance` float(10,2) DEFAULT NULL,
 `Fare_amount` decimal(5,3) DEFAULT NULL,
 `Extra` decimal(5,3) DEFAULT NULL,
 `MTA_tax` decimal(5,3) DEFAULT NULL,
 `Tip_amount` decimal(5,3) DEFAULT NULL,
 `Tolls_amount` decimal(5,3) DEFAULT NULL,
 `improvement_surcharge` decimal(5,3) DEFAULT NULL,
 `Total_amount` decimal(5,3) DEFAULT NULL,
 `Payment_type` tinyint DEFAULT NULL,
```

```
 `Trip_type` tinyint DEFAULT NULL,
 KEY `VendorID` (`VendorID`),
 KEY `pickup_datetime` (`lpep_pickup_datetime`),
 KEY `dropoff_datetime` (`Lpep_dropoff_datetime`)
);


DROP TABLE IF EXISTS `lyft`;

CREATE TABLE `lyft` (
 `base_license_number` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
 `base_name` varchar(40) COLLATE latin1_general_ci DEFAULT NULL,
 `dba` varchar(40) COLLATE latin1_general_ci DEFAULT NULL,
 `pickup_end_date` DATE NOT NULL,
 `pickup_start_date` DATE NOT NULL,
 `total_dispatched_trips` smallint DEFAULT NULL,
 `unique_dispatched_vehicle` smallint DEFAULT NULL,
 `wave_number` tinyint DEFAULT NULL,
 `week_number` tinyint DEFAULT NULL,
 `years` smallint DEFAULT NULL,
 KEY `base_name` (`base_name`),
 KEY `pickup_end_date` (`pickup_end_date`),
 KEY `pickup_start_date` (`pickup_start_date`)
);


DROP TABLE IF EXISTS `uber`;
```

```
CREATE TABLE `uber` (
 `lat` double(7,5) DEFAULT NULL,
 `lon` double(7,5) DEFAULT NULL,
 `dispatching_base_num` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
 `pickup_date` DATETIME NOT NULL,
 `affiliated_base_num` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
 `locationid` tinyint DEFAULT NULL,
 KEY `pickup_date` (`pickup_date`),
 KEY `locationid` (`locationid`)
);


CREATE VIEW yellow_old_sum AS
SELECT YEAR(tpep_pickup_datetime) as the_year, MONTH(tpep_pickup_datetime) AS the_
  FROM yellow_old
  GROUP BY the_year, the_month;
);
```

Chapter 3

New York City Taxi Drivers

The income of Taxi drivers in New York City comes from two sources: taxi fare and tips. Taxi fare is usually calculated by the meters installed in the taxis, and the rate of fare cannot be changed by taxi drivers. Therefore, in order to make more profit, taxi drivers prefer to pick up passengers who offer big amount of tips. What are the regions that provide the most tips to yellow taxicab drivers?

In this analysis, we will focus on trip data collected in 2017. Descriptions of variables mentioned in the following chapters can be found in Appendix B.

In order to answer questions regarding to taxi trips' tips, we filter out trips that are not paid by credit or debit card, because taxi drivers usually do not correctly record the amount of tips paid by cash or check (Appendix C)(W. Li, 2018).

As mentioned in the previous chapter, we can utlize the connection to a MySQL database to run data analysis in MySQL for medium-sized data. Since we are using all 12 month data from 2017 in this analysis, it is impractical to load all data needed into **R** environment. Instead, we want to only load a fraction of the 2017 Yellow Taxi data from MySQL database.

In this section, we only want to load trip records with payment type equals to 1, which represents credit card. Only trip records with payment type credit card have accurate information on tip amount. Let's load the 2017 trip record into **R** environment by using the MySQL connection we just generated, `taxi`.

```
yellow_2017 <- taxi %>%
  tbl("yellow") %>%
  filter(payment_type == 1) %>%
  collect(n = Inf)
```

3.1 Aggregated Zone-level Tip Amount

Instead of the nominal amount of tips, we want to focus on the percentage of tips that passengers pay in addition to the total fare amount. Therefore, we use tip amount over fare amount to calculate the percent tip. We then calculated the mean percent tip, mean distances travelled, mean number of minutes spent travelling, and total number of trips of each pick-up and drop-off pair in 2017 to get the aggregated zone-level information in order to compare the percent tip passengers pay in each zone.

```
yellow_2017_summary <- yellow_2017 %>%
  mutate(year = year(tpep_pickup_datetime),
         month = month(tpep_pickup_datetime),
         tip_perct = tip_amount/fare_amount) %>%
  group_by(year, month, PULocationID, DOLocationID) %>%
  summarise(avg_tip = mean(tip_perct),
            trips = n(),
            avg_dis = mean(trip_distance),
            avg_duration = mean(duration))
```

Each taxi trip has pick-up and drop-off locations associated with it, and there are 263 known taxi zones. Taxi meters sometimes do not function properly, so the information recorded is not always accurate. When taxi meters dysfunction, taxi pick-up and drop-off locations are labelled as "Unknown". We only want to include trips coming from and going to known taxi zones in this analysis.

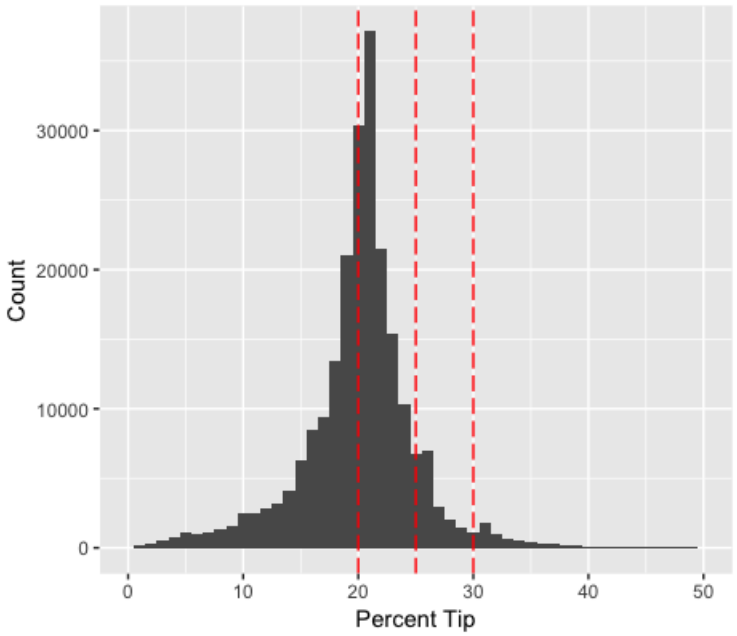


Figure 3.1: Percent Tip Paid by Passengers in Each Pick-up And Drop-off Pair in NYC

Figure 3.1 is a histogram of mean tip percents for all known pick-up and drop-off zone pairs. The red, green, and yellow dash lines are drawn at 20%, 25%, and 30%, which are the default percentage of tips that are shown on the touch panel for credit and debit car payments (see Figure 3.2), and passengers tend to pick the the lowest default percent tip.

Figure 3.2: Tip Payment Page on New York City Touch Panel

3.1.1 Pick-up Zone Percent Tips Amount

Taxi drivers are required to be indifferent to where passengers are going. It is illegal for New York city taxi drivers to refuse service because of passengers' race, ethnicity, cultural background, disbility, gender, or destination (N. T. Staff, 2018). Taxi drivers cannot choose where the passengers want to go, and instead they can only choose which pick-up zone they would prefer to drive around to get hailed. Therefore, it makes sense to investigate the average amount of tips paid by passengers departed from each pick-up zone. What are the taxi pick-up zones that have the highest percent tips paid by passengers?

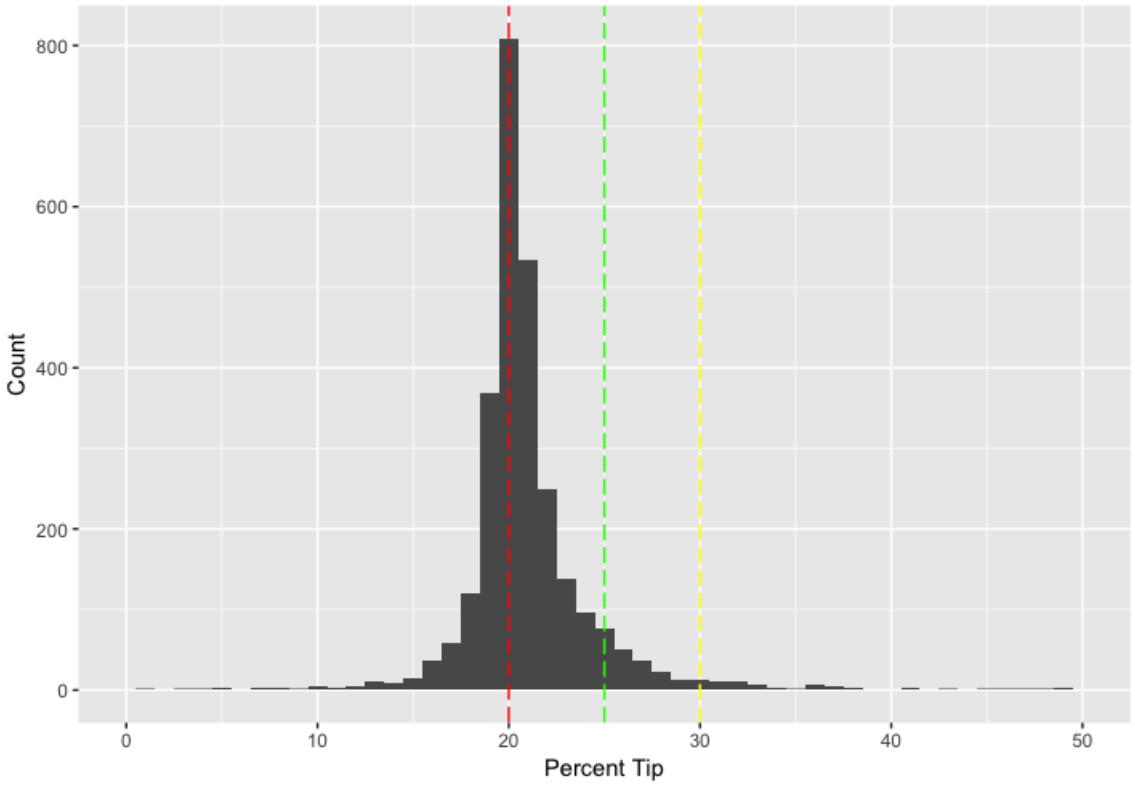


Figure 3.3: Percent Tip Paid by Passengers on Each Pick-up Taxi Zone in NYC

We created a histogram to visualize the distribution of average percent tips paid for all pick-up zones. As show in Figure 3.3, the first peak is around 20%, which is the cheapest default option on the touch panel for passengers to chose. We also calculated the average percent tip paid for each pick-up zone as shown in Table 3.1. According to Table 3.1, 6 out of 10 taxi zones with the highest average percent tips are in Queens. At a first glance, Queens seems to be a good place for taxi drivers to go and pick up passengers to make more money.

Table 3.1: Ten taxi pick-up zones with the highest average tip in January, 2017

Borough	Zone	Average % Tips
Queens	Douglaston	29
Bronx	East Tremont	29
Queens	Oakland Gardens	29
Queens	Glendale	28
Queens	Saint Michaels Cemetery/Woodside	28
Queens	Bayside	27
Brooklyn	Coney Island	27
Queens	Howard Beach	27
Brooklyn	Marine Park/Mill Basin	27
Bronx	Norwood	26

3.1.2 Which taxi zones are the most popular ones for pick-ups?

Which pick-up zones have the highest number of taxi trip pick-ups? We can create a heat map to visualizae the number of trips for each pick-up zones on a map of New York City Taxi Zones.

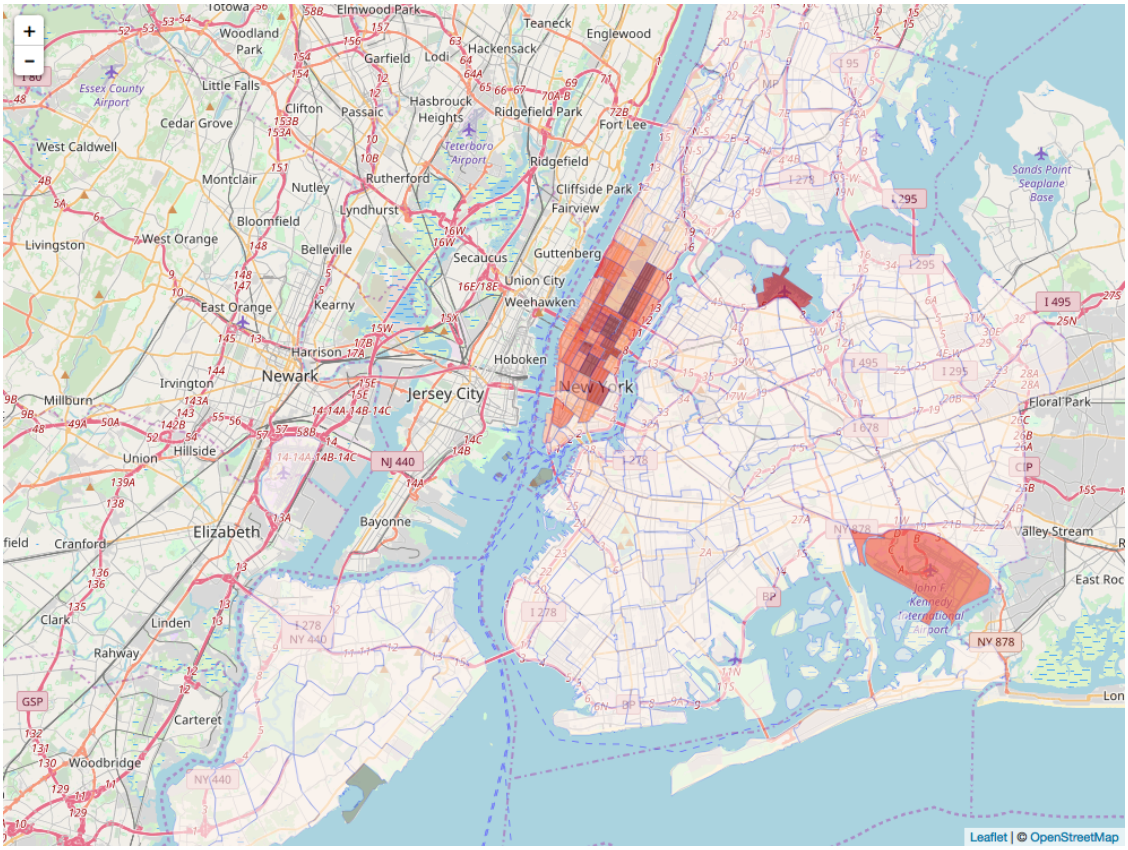


Figure 3.4: Number of Pick-ups in Each Taxi Zone

In Figure 3.4, taxi zones with more number of pick-ups are colored by darker shades of orange, and it is obvious that Manhattan and La Guardia Airport are the most popular locations for taxi pick-ups. Table 3.2 tells which specific taxi zones have the highest number of pick-ups, and 9 out of the top 10 taxi zones that have the most number of pick-ups are located in Manhattan. There are about 6000 yellow taxi pick-ups in the top 10 taxi zones everyday in 2017.

3.1.3 Which pick-up zones have the highest percent tips?

Most yellow cab pick-ups occur in Manhattan. If we focus on the pick-up zones that have at least 1 trip per hour (or 24 trips per day), we will observe that many taxi pick-up zones with the highest percent tips are not necessarily the ones with the

Table 3.2: Ten taxi zones with the highest number of pick-ups

Borough	Zone	Number of Trips
Manhattan	Upper East Side South	2519900
Manhattan	Midtown Center	2461602
Manhattan	Union Sq	2382970
Manhattan	Upper East Side North	2372509
Manhattan	Midtown East	2349386
Manhattan	Murray Hill	2231723
Manhattan	Penn Station/Madison Sq West	2193036
Manhattan	East Village	2097416
Queens	LaGuardia Airport	2059444
Manhattan	Times Sq/Theatre District	1972303

Table 3.3: Ten taxi pick-up zones with the highest percent tip (taxi zones has at least 1 pick-up per hour)

Borough	Zone	Average % Tips
Queens	Baisley Park	21.55
Brooklyn	Gowanus	21.45
Queens	Steinway	21.36
Brooklyn	Carroll Gardens	21.18
Queens	LaGuardia Airport	21.00
Brooklyn	Greenpoint	21.00
Brooklyn	Prospect Heights	21.00
Manhattan	Midtown Center	20.82
Brooklyn	Cobble Hill	20.82
Brooklyn	East Williamsburg	20.82

highest number of pick-ups. People might think it is more reasonable to see a list that is populated with zones in Manhattan, since that's where most of the wealthy people live. However, Table 3.3 shows that passengers who get on taxis from certain zones in Brooklyn and Queens also pay a lot of tips. Taxi drivers who would love to get more tips compensation can drive to the zones listed above to pick-up passengers.

If we focus on the pick-up zones that have more than 1 trip per minute (or 24 trips per hour), then we observe that all pick-up zones that have the highest percent tips are in Manhattan besides La Guardia Airport. There are more than 100 times more yellow

Table 3.4: Ten taxi pick-up zones with the highest percent tip (taxi zones has at least 1 pick-up per minute)

Borough	Zone	Average % Tips
Queens	LaGuardia Airport	21.00
Manhattan	Midtown Center	20.82
Manhattan	Battery Park City	20.73
Manhattan	Midtown East	20.55
Manhattan	Murray Hill	20.55
Manhattan	Penn Station/Madison Sq West	20.55
Manhattan	UN/Turtle Bay South	20.45
Manhattan	Times Sq/Theatre District	20.36
Manhattan	Union Sq	20.18
Manhattan	Midtown North	20.18

cab pick-ups that happen in Manhattan everyday than in Brooklyn. By comparing the average tip percent in Table 3.3 and Table 3.4, we observe that 8 out of 10 average percent tips in taxi zones with high pick-up numbers in Table 3.4 are lower than average percent tips in taxi zones with low pick-up numbers in Table 3.3. Therefore, there could be a correlation between number of trips and average percent tips that passengers pay, and this can be futher studied with more taxi-zone-specific data, such as median household income, provided.

3.2 What features of taxi trips increase the percent tip amount that passengers pay?

So far, we have learned what pick-up zones offer the highest percent tip. Now, we want to dig into the relationships between percent tip and taxi-zone-specific variables.

3.2.1 Does trip distance increase the percent tips paid by passengers?

Do longer trips result in higher tip percent? It takes taxi drivers more time to complete longer trips, so passengers might want to compensate taxi drivers more. I personally pay higher percent of tips for longer rides, so I believe trip distance has an impact on percentage of tips paid.

```
               Estimate   Std. Error   t value Pr(>|t|)
(Intercept)  0.211325482 3.733048e-04 566.09360        0
avg_dis     -0.001052903 2.105014e-05 -50.01881        0
```

Acoording to the simple linear regression result, trip distance does have a negative impact on the percent of tips paid, controlling for both pick-up and drop-off locations. Since the number of obervations in this regression model is big, the p-value quickly goes to zero. Therefore, in this regression, p-value does not matter so much. What is important in this result is the negative correlation between average percent tips and average distance that a taxi travels.

The negative correlation could be caused by a psychological reason. Long trips cost more than short trips. For a constant tip percent, the nominal value of tip amount cost more for longer trips. For example, for a \$100 trip, 20% tip costs \$20; for a \$50 trip, 20% tip costs \$10. Even though consumers are paying the same percent amount of tips, \$20 is more expensive than \$10. Therefore, consumers might decide to pay less percent tip for longer trips.

3.2.2 Do passengers pay more tips during rush hours?

New York City Taxi Fare & Limousine Commission has information on how New York City taxi fare amount is calculated on their official website.

Metered Fare Information

- Onscreen rate is 'Rate #01 – Standard City Rate.'
- The initial charge is $2.50.
- Plus 50 cents per 1/5 mile or 50 cents per 60 seconds in slow traffic or when the vehicle is stopped.
- In moving traffic on Manhattan streets, the meter should "click" approximately every four downtown blocks, or one block going cross-town (East-West).
- There is a 50-cent MTA State Surcharge for all trips that end in New York City or Nassau, Suffolk, Westchester, Rockland, Dutchess, Orange or Putnam Counties.
- There is a 30-cent Improvement Surcharge.
- There is a daily 50-cent surcharge from 8pm to 6am.
- There is a $1 surcharge from 4pm to 8pm on weekdays, excluding holidays.
- Passengers must pay all bridge and tunnel tolls.
- Your receipt will show your total fare including tolls. Please take your receipt.
- The driver is not required to accept bills over $20.
- Please tip your driver for safety and good service.
- There are no charges for extra passengers or bags.

The metered fare rate information is collected from TLC rate of fare webpage (N. T. Staff, n.d.).

In taxi fare calculation, the only unknown variable is slow-traffic time, and all other variables were collected by the meters installed on each medallion taxi for each trip. It is reasonable to assume that for trips with the same pick-up and drop-off locations, the longer the total slow traffic time is, the longer the trip would take. Taxi drivers are compensated for both the normal-speed trip distance and the time spent in slow-traffic. According to the fare calculation algorithm, in moving traffic on Manhattan streets,

the meter should "click" approximately every four downtown blocks, or one block going cross-town (East-West); in slow traffic, the meter should "click" every 60 seconds. Therefore, slow traffic increases the minute per mile ratio.

New York City has the worst traffic jams, and it has overtaken Miami to be voted the U.S. city with the angriest and most aggressive drivers in 2009, according to a survey on road rage. Bad traffic also causes slow-traffic, and taxi drivers tend to get stuck in traffic during rush hours (Reaney, 2009). Does minute per mile ratio have an impact on the percent tip that passengers pay? Do passengers compensate taxi drivers more during rush hours? Are passengers sympathetic to taxi drivers for the time they spend in slow traffic?

```
               Estimate   Std. Error   t value Pr(>|t|)
(Intercept)  0.193453599 1.870711e-04 1034.1180        0
min_per_mile 0.002841351 3.134322e-05   90.6528        0
```

As shown in the regression result, `min_per_mile` ratio does have a positive impact on percent tips. Since trips with slow traffic can be depicted by high minute per mile ratio, passengers do pay more tips during rush hours.

3.3 Recommendations to Taxi Drivers

Our analysis has suggested that taxi passengers are sympathetic with the drivers who have to suffer the cogestion in New York City, and taxi drivers do get compensated more during rush hours. Therefore, we hope that taxi drivers would feel better during rush hours by knowing that passengers do pay for tips to compensate the negative feelings that drivers carry in cogestion.

Chapter 4

New York City Taxi Passengers

4.1 How long does it take passengers to get to JFK, La Guardia, and Newark Airports from anywhere in New York City?

We want to calculate the average number of minutes it takes to go to all three airport from a specific taxi zone at every hour. First, we want to focus on trips going to any of the three airports, JFK, LaGuardia, or Newark Airport. We need to load trip records with destination as one of the three airports from the MySQL connection we built.

```
to_jfk_trip <- taxi %>%
  tbl("yellow") %>%
  filter(DOLocationID == 132) %>%
  collect(n = Inf)


to_lg_trip <- taxi %>%
```

Table 4.1: Average number of minutes it takes from Alphabet City, Manhattan to JFK Airport during different hours

	Borough	Zone	Hour of Departure	Average Number of Minutes
10	Manhattan	Alphabet City	0	45.37000
11	Manhattan	Alphabet City	1	36.77500
12	Manhattan	Alphabet City	2	28.66000
13	Manhattan	Alphabet City	3	27.83350
14	Manhattan	Alphabet City	4	27.19490
15	Manhattan	Alphabet City	5	28.68889
16	Manhattan	Alphabet City	6	34.25271
17	Manhattan	Alphabet City	7	38.13817
18	Manhattan	Alphabet City	8	41.59687
19	Manhattan	Alphabet City	9	35.39226
20	Manhattan	Alphabet City	10	36.22867
21	Manhattan	Alphabet City	11	41.12000
22	Manhattan	Alphabet City	12	41.02800

```
  tbl("yellow") %>%
  filter(DOLocationID == 138) %>%
  collect(n = Inf)


to_newark_trip <- taxi %>%
  tbl("yellow") %>%
  filter(DOLocationID == 1) %>%
  collect(n = Inf)
```

Now we want to calculate the average amount of time it take from each zone to one of the three airports during each hour.

So far, we have created three tables summarizing the average number of minutes it takes to go to all three airports for every hour from different taxi zones. It would be easier if we combine all three tables and put information related to trip duration to all three airports in the same table. Table 4.1 displays the average number of minutes

it takes from Alphabet City, Manhattan to JFK Airport during different hours.

4.1.1 Case Study: From Central Park, Manhattan to all three airports

Central Park, Manhattan has pick-up zone ID number 43. Let's take a look at how much time is needed to travel to all three airports from taxi zone No.4.

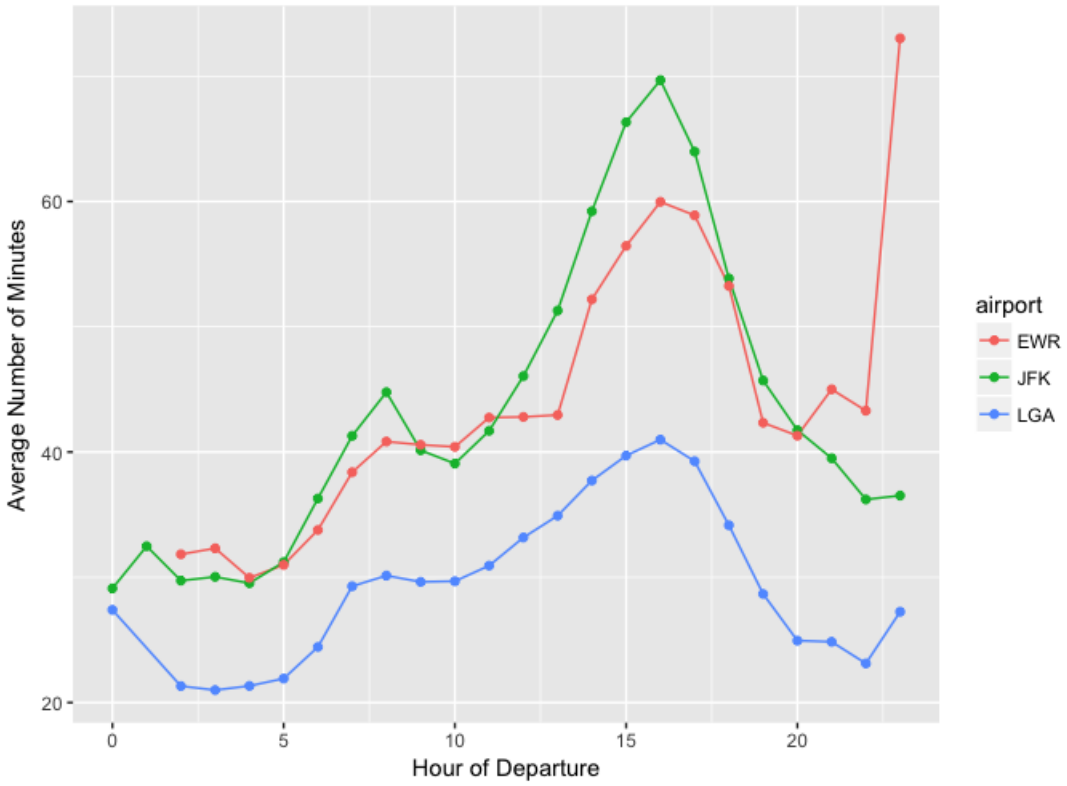


Figure 4.1: Average number of minutes it takes from Central Park, Manhattan to all three airports during different hours

According to the red line Figure 4.1, it takes the least time, less than 30 minutes, to travel from Central Park, Manhattan to Newark Airport around 4 AM in the morning and it takes more than 70 minutes around 11 PM at night.

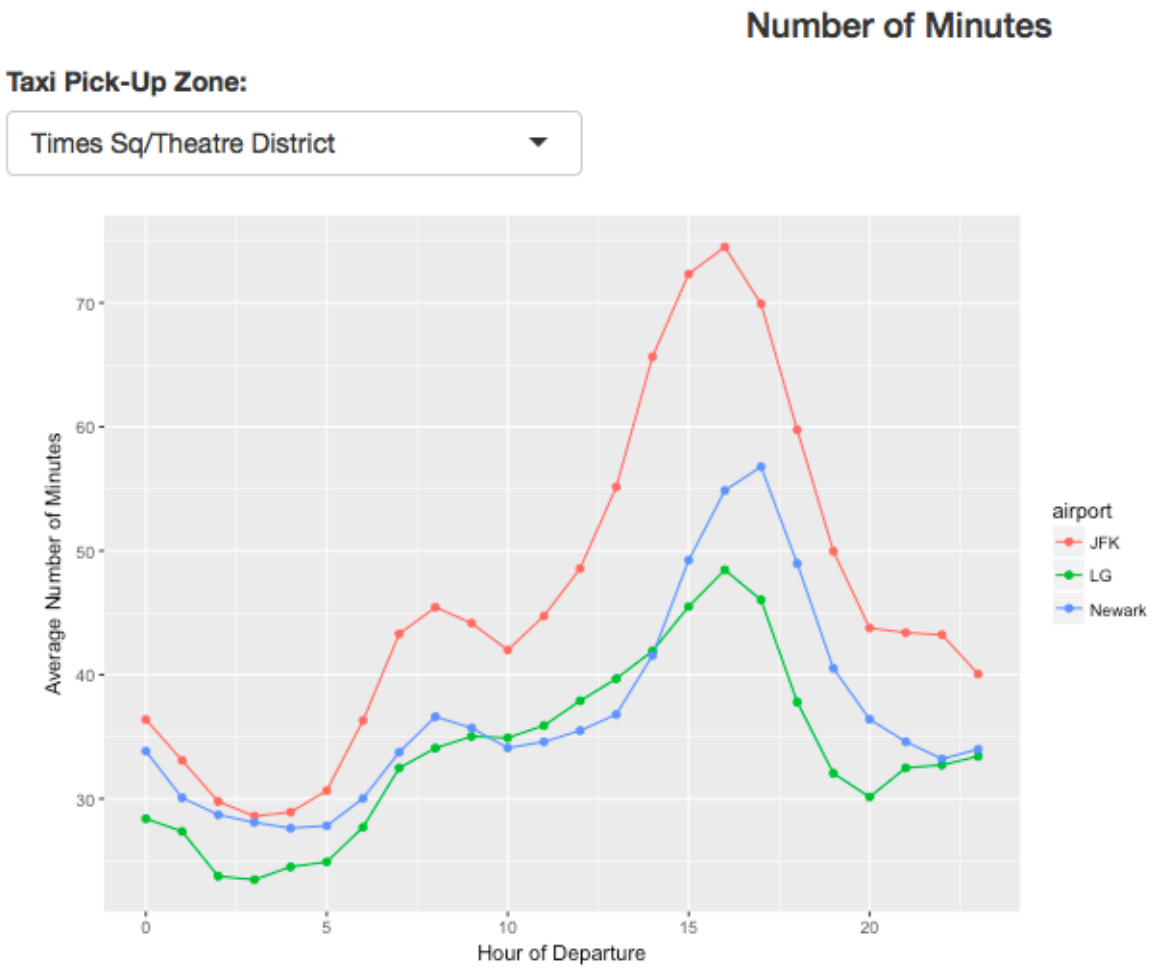
According to the green line, it only takes about 30 minutes to travel to JFK Airport

around 4 AM in the morning, and it takes the most time, about 70 minutes, around 4 PM in the afternoon.

As shown by the blue line, it takes the least time, about 20 minutes, to travel to La Guardia Airport at 1 AM at midnight, and it takes a little more than 40 minutes around 4 PM in the evening.

Being able to know the average time it takes to go to one of the airports ahead, passengers can buy their flight tickets accordingly. For example, a mum who wants to visit Disney World with her kids can use this visulization to estimate the amount of time needed for her and her families to catch their flight. If this mum wants to catch a flight that departs at 10 AM from JFK Airport, then when she should depart from Central Park in order to get to the airport on time? According to Figure 4.1, she can depart at 7 AM, and it will take a little bit more than 40 minutes for her to get to JFK Airport to catch her 10 AM flight.

4.1.2 A Shiny App: When is the best time to travel to JFK Airport?



This Shiny App helps passengers to estimate the amount of time that is needed for them to travel to any one of the these three airports from any New York City taxi zones.

4.2 How does weather affect the number of taxi and Uber trips?

On a snowy or rainy day, it is hard for passengers to find a yellow cab on the street. Taxi drivers get paid at the same rate no matter how bad the weather gets, so they tend to stay at home instead of going out to work when the weather is bad. Uber drivers, however, get paid more on a snowy or rainy day, since Uber uses a pricing model that takes the number of Uber vehicles available on the street into account. When weather is bad, fewer Uber vehicles are available on the street, so Uber fare rate increases. Uber's pricing model gives Uber drivers an incentive to keep working on ugly days.

In this section, we study the number of pickups of yellow cab and Uber. We compare number of pick-ups in each taxi zone in the weeks of bad weather with previous weeks' total number of pick-ups to see whether Uber drivers have an incentive to drive around the city more when weather gets bad.

Uber Weekly Data We first calculated the number of total dispatched trips of Uber by using weekly-aggregated Uber pick-up data available on NYC OpenData (N. O. Staff, 2015b), and summary is shown in Table 4.2.

Yellow Cab Weekly Data

We also calculated the number of total dispatched trips of New York City yellow cabs by using `nyctaxi` package to retrieve yellow taxi data from 2017, and the summary is shown in Table 4.3.

Table 4.2: Uber 2017 Weekly Total Dispatched Trips

Pickup Start Date	Pickup End Date	Uber Total Dispatched Trips
2017-01-01	2017-01-07	2866569
2017-01-08	2017-01-14	3114792
2017-01-15	2017-01-21	3089595
2017-01-22	2017-01-28	3299763
2017-01-29	2017-02-04	3224451
2017-02-05	2017-02-11	3310481
2017-02-12	2017-02-18	3456042
2017-02-19	2017-02-25	3194805
2017-02-26	2017-03-04	3533347
2017-03-05	2017-03-11	3614559

Table 4.3: Yellow Taxi 2017 Weekly Total Dispatched Trips

Pickup Start Date	Pickup End Date	Yellow Total Dispatched Trips
2017-01-01	2017-01-07	2044643
2017-01-08	2017-01-14	2230950
2017-01-15	2017-01-21	2219214
2017-01-22	2017-01-28	2307122
2017-01-29	2017-02-04	2331749
2017-02-05	2017-02-11	2181622
2017-02-12	2017-02-18	2387399
2017-02-19	2017-02-25	2225850
2017-02-26	2017-03-04	2464800
2017-03-05	2017-03-11	2456285

Table 4.4: Yellow Taxi Total Dispatched Trips

Pickup Start Date	Pickup End Date	Yellow Total Dispatched Trips
2017-03-05	2017-03-11	2456285
2017-03-12	2017-03-18	2066285

Table 4.5: Uber Total Dispatched Trips

Pickup Start Date	Pickup End Date	Uber Total Dispatched Trips
2017-03-05	2017-03-11	3614559
2017-03-12	2017-03-18	3430189

4.2.1 Case Study: March 14th, 2017 Snow Storm

There are two commonly known bad weather conditions, rainy and snowy days. Let's first focus on snowstorm. On March 14th, 2017, a snow storm brought seven inches of snow to New York City.

Yellow Taxi

```
[1] -0.1587764
```

Yellow taxi's number of total dispatched trips declined by 15% (see Table 4.4).

Uber

```
[1] -0.05100761
```

Uber's number of total dispatched trips declined by 5% (see Table 4.5).

In this case, we observe that the percent decline in Uber's total number of pick-ups is 10% less than the percent decline in Yellow Taxi's total number of drop-off. Even though the total number of Uber pick-ups did not increase, Uber's pricing model may keep more drivers in the market on a snowy day.

Table 4.6: 10 weeks that have the most rainfall in 2017

Pickup Start Date	Pickup End Date	Weekly Rainfall
2017-06-18	2017-06-25	7.00
2017-04-30	2017-05-07	6.71
2017-10-29	2017-11-05	6.16
2017-07-02	2017-07-09	5.56
2017-03-26	2017-04-02	5.11
2017-01-22	2017-01-29	3.99
2017-08-13	2017-08-20	3.95
2017-06-11	2017-06-18	3.91
2017-04-02	2017-04-09	3.17
2017-04-23	2017-04-30	2.88

4.2.2 Case Study: Impact of Precipitation on Taxi Rides

People living in New York might have noticed that it is hard to find a taxi on the street when it rains. Economists have studied this phenomena for a long time, and an analysis that studied the correlation between taxi movement and hourly rainfall data in Central Park from 2009 to 2013 found that there is no significant correlation between a driver's hourly wage and rain in the city, which implies that drivers don't earn more when it's raining (Jaffe, 2014).

We got access to the 2017 daily Central Park weather data from the National Climatic Data Center by submitting a Climate Data Online request (Appendix D) to National Centers for Environmental Information (Environmental Information Staff, n.d.), and joined it to the 2017 taxi data to study relationship between rainfall and taxi rides.

First, we generate a list of total amount of daily rainfall in New York City and we pick the 10 weeks that have the most rainfall in 2017 (see Table 4.6). We then find the weekly total number of dispatched yellow taxi trips of the 10 weeks with the most rainfall (see Table 4.7). We also need to add the weekly total number of dispatched Uber trips of the 10 weeks with the most rainfall (see Table 4.8). We combine the

Table 4.7: 10 weeks that have the most rainfall in 2017 and the total number of dispatched yellow taxi trips in those weeks

Pickup Date	Dispatched Trips	Last Week Date	Last Week Trips	% Change Trips
2017-06-18	2231205	2017-06-11	2285958	-2.40
2017-04-30	2386559	2017-04-23	2394329	-0.32
2017-10-29	2266196	2017-10-22	2267693	-0.07
2017-07-02	1664159	2017-06-25	2038406	-18.36
2017-03-26	2341096	2017-03-19	2272369	3.02
2017-01-22	2307122	2017-01-15	2219214	3.96
2017-08-13	1871668	2017-08-06	1929860	-3.02
2017-06-11	2285958	2017-06-04	2313236	-1.18
2017-04-02	2414700	2017-03-26	2341096	3.14
2017-04-23	2394329	2017-04-16	2337161	2.45

Table 4.8: 10 weeks that have the most rainfall in 2017 and the total number of dispatched Uber trips in those weeks

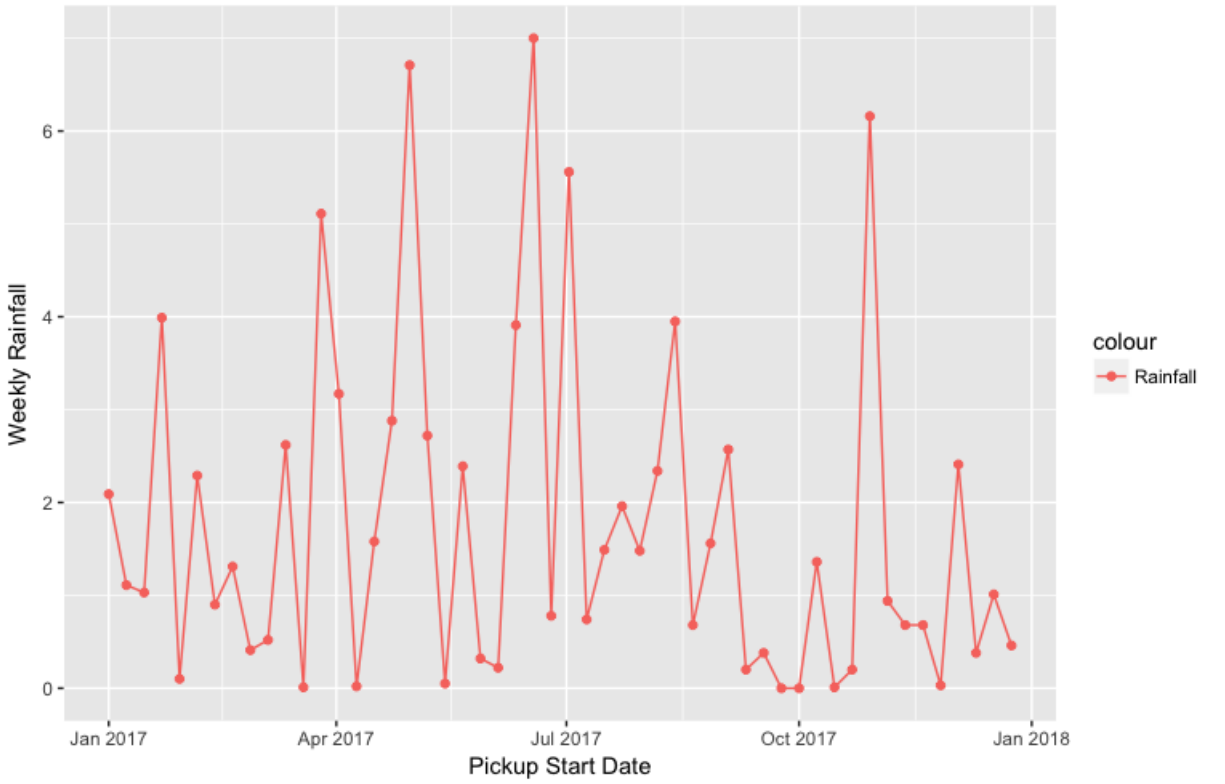
Pickup Date	Dispatched Trips	Last Week Date	Last Week Trips	% Change Trips
2017-06-18	3654932	2017-06-11	3658220	-0.09
2017-04-30	3546893	2017-04-23	3606408	-1.65
2017-10-29	4317572	2017-10-22	4193611	2.96
2017-07-02	3212582	2017-06-25	3406814	-5.70
2017-03-26	3541624	2017-03-19	3425475	3.39
2017-01-22	3299763	2017-01-15	3089595	6.80
2017-08-13	3599772	2017-08-06	3584023	0.44
2017-06-11	3658220	2017-06-04	3622252	0.99
2017-04-02	3443444	2017-03-26	3541624	-2.77
2017-04-23	3606408	2017-04-16	3427564	5.22

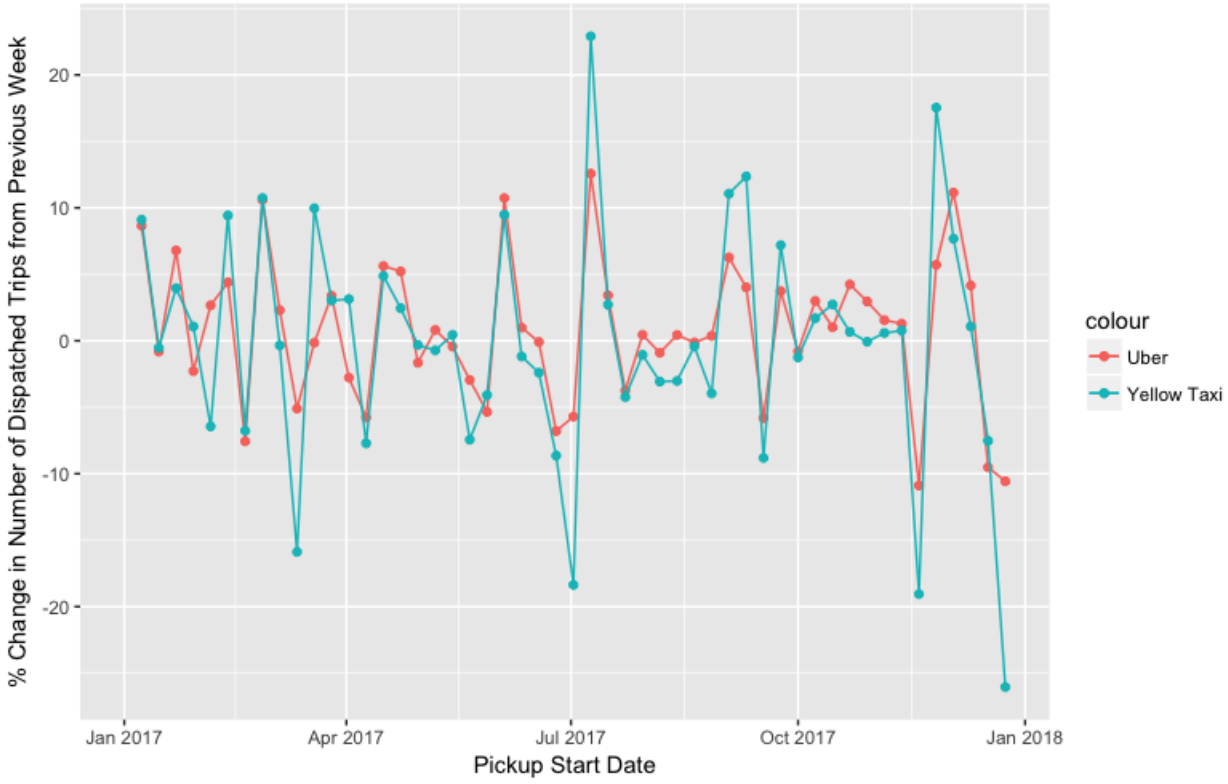
Table 4.9: The percentage change in total number of dispatched trips comparing to the previous weeks of yellow taxi and Uber

Pickup Start Date	Weekly Rainfall	Last Week Date	Uber	Yellow
2017-06-18	7.00	2017-06-11	-0.09	-2.40
2017-04-30	6.71	2017-04-23	-1.65	-0.32
2017-10-29	6.16	2017-10-22	2.96	-0.07
2017-07-02	5.56	2017-06-25	-5.70	-18.36
2017-03-26	5.11	2017-03-19	3.39	3.02
2017-01-22	3.99	2017-01-15	6.80	3.96
2017-08-13	3.95	2017-08-06	0.44	-3.02
2017-06-11	3.91	2017-06-04	0.99	-1.18
2017-04-02	3.17	2017-03-26	-2.77	3.14
2017-04-23	2.88	2017-04-16	5.22	2.45

percentage change in total number of dispatched trips of yellow taxi and Uber, and we compare the result (see Table 4.9). Besides the week of April 30th, 2017, all other weeks have higher increases in the number of total dispatched trips of Uber or lower declines in the number of weekly Uber trips. Therefore, on rainy days, Uber drivers tend to increase the number of trips they drive at a higher rate.

We then plot the weekly rainfall and Yellow Taxi and Uber's percent Change in Number of Dispatched Trips from Previous Week.


Weekly Rainfall
6
4
2
0
Jan 2017
Apr 2017
Jul 2017
Oct 2017
Jan 2018
Pickup Start Date
colour
Rainfall


% Change in Number of Dispatched Trips from Previous Week
20
10
0
-10
-20
Jan 2017
Apr 2017
Jul 2017
Oct 2017
Jan 2018
Pickup Start Date
colour
Uber
Yellow Taxi

According to Figure @ref(fig:rainfall_vis) and Figure @ref(fig:weather_vis), when weekly rainfall is high, Uber usually have less percent decline in total number of dispatched trips comparing to the total number of trips from previous week than yellow cab does. Uber passengers pay higher fare on rainy days becasue of Uber's pricing model. Since taxi drivers do not get paid more on rainy days, they tend to work less than Uber drivers, which limits the options for passengers. Passengers sometimes have to choose the more costly Uber instead.

4.3 Recommendations to Taxi Passengers and NYC TLC

We suggest passengers to use our Shiny App to choose a pick up zone of their interest and then decide when is the most favorable time for them to travel from that zone to any of the three airports in New York.

In this chapter, we have shown that Uber's pricing model might keep more drivers working on rainy days. Therefore, we suggest New York City TLC to modify the fare on rainy or snowy days to incentive taxicab drivers to pick up more trips in order to make taking a street hail vehicle on average more affordable on rainy days for passengers.

Chapter 5

New York City Taxi & Limousine Commission

5.1 Should there be a flat rate between Manhattan and John F. Kennedy International Airport?

Why is there a flat rate to and from JFK airport and any location in Manhattan? Why is the flat rate $52? Does TLC make profit from the $52 flat rate? Does $52 reduce the cogestion on the road to JFK airport and make taking a train a more preferable choice? The New York City taxi trip records can reveal the answers to these questions.

Imagine it's your first time travelling to New York City, and you decided to stay in a hotel in Manhattan. Since you might not know much about the city, the fixed $52 flat rate provides cost certainty for you, and it incentivizes you to take taxi to JFK Airport. If there is no flat rate, there is uncertainty in how much someone needs to

pay to take a taxi to JFK, and tourists might instead choose to take the train, even though taking a train would cost them more time, anxiety, and inconvenience.

Additionally, people living in most parts of Manhattan would have paid more than $52 to take a taxi to go to the JFK Airport. The higher the taxi fare is, the less the demand for taxi will be. Therefore, having a flat rate might help taxi drivers to get more trips from Manhattan to JFK Airport.

5.2 Passengers departing from Manhattan benefit from the $52 flat rate

If there is no flat rate between JFK and Manhattan, how much would passengers pay for the distance they travelled between JFK Airport AND Manhattan? And how much more or less should they have paid comparing to the $52 flat rate?

In this study, we are only interested in yellow taxi trip between Manhattan and JFK Airport. Since JFK Airport's Location ID is 132, we only retrieve trip records with either pick-up or drop-off location ID as 132 from the database.

```
to_jfk <- taxi %>%
  tbl("yellow") %>%
  filter(DOLocationID == 132) %>%
  collect(n = Inf)


from_jfk <- taxi %>%
  tbl("yellow") %>%
  filter(PULocationID == 132) %>%
  collect(n = Inf)
```

5.2.1 Trips from Manhattan to JFK Airport

We first focus on all the trips that departed from Manhattan and went to JFK Airport, and then we calculate the estimated fare amount that the passengers should have paid based on the distance travelled from each pick-up point to JFK Airport based on the fare rate suggested by TLC for each pick-up zone.

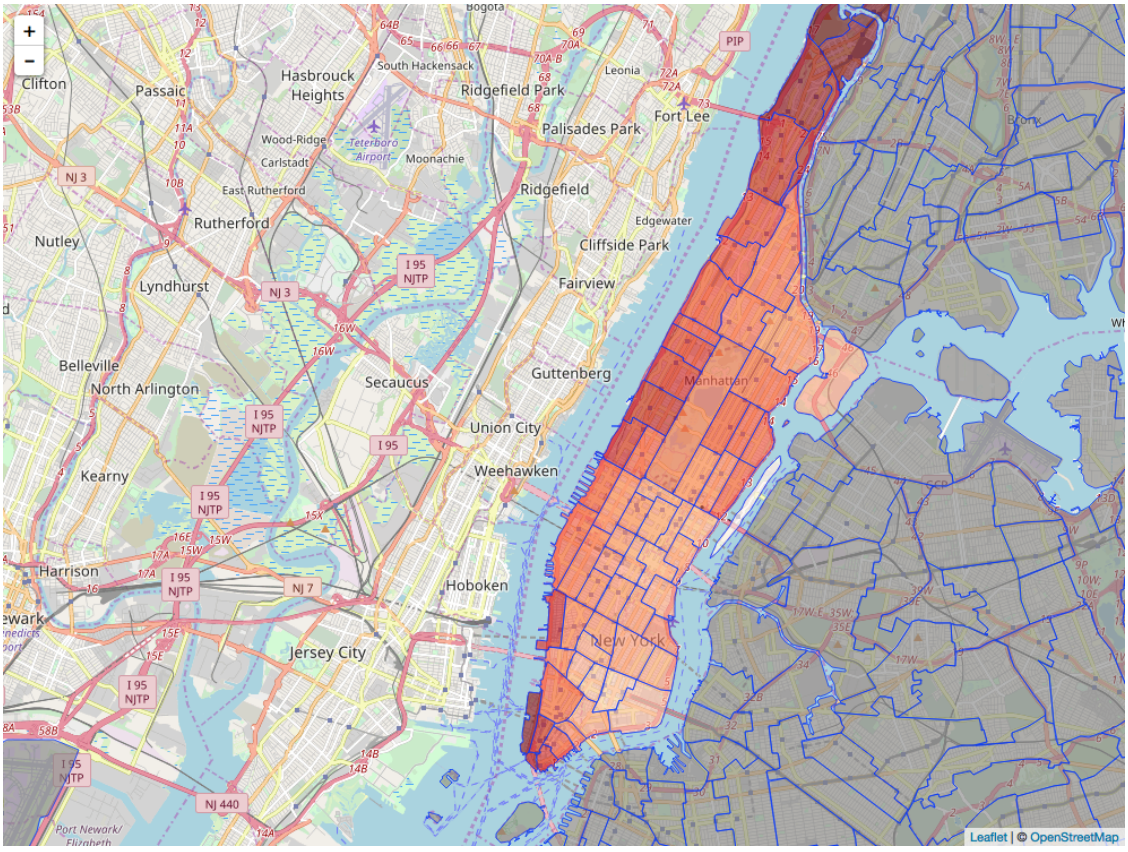


Figure 5.1: Estmated fare amount from the each pick-up zone to JFK Airport

Figure 5.1 is a map of estmated fare amount calculated by taking the average of all estimated fare amounts from the same pick-up zone to JFK Airport based on the fare rate suggested by TLC for each pick-up zone. According to the map, trips from Midtown on average cost less than trips from other taxi zones in Manhattan.

Table 5.1: Ten pick-up zones with the highest avergae fare from Manhattan to JFK Airport

avg_est_fare	avg_est_diff	Borough	Zone
64.03150	11.844558	Manhattan	Battery Park City
63.98256	9.970366	Manhattan	Inwood
62.97567	10.892992	Manhattan	Washington Heights North
61.99327	9.889636	Manhattan	Battery Park
60.49388	8.278941	Manhattan	Washington Heights South
60.18006	8.107309	Manhattan	Upper West Side South
59.74384	7.511991	Manhattan	World Trade Center
59.31411	7.058534	Manhattan	Meatpacking/West Village West
59.24692	7.200516	Manhattan	Lincoln Square West
59.13439	7.083517	Manhattan	Upper West Side North

5.2.2 Which taxi zones would pay more than $52 without the flat rate?

We computed the average fare paid by passengers for trips going from each taxi zone in Manhattan to JFK Airport in Table 5.1.

Let's visualize the taxi zones that would have cost more than the $52 flat rate.

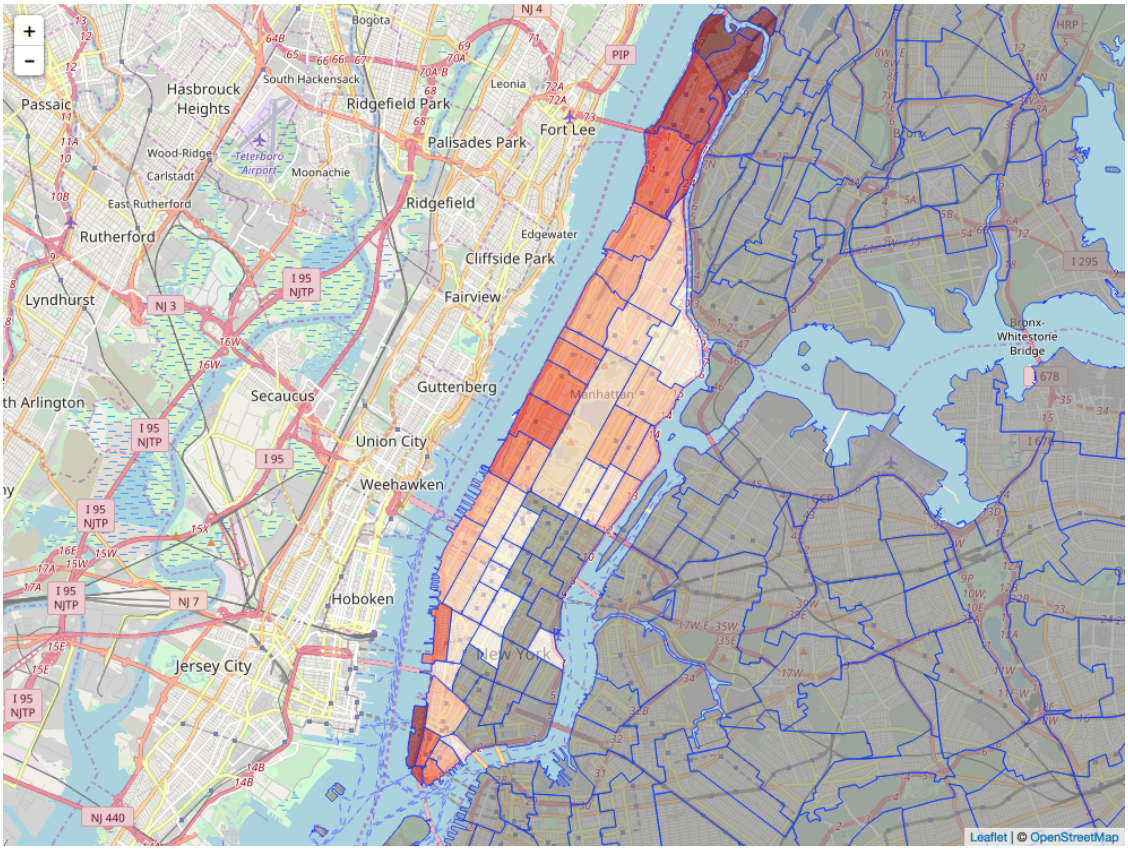


Figure 5.2: Pick-up Zones that cost more than the 52 US Dollar flat rate

Therefore, passengers from places in Manhattan besides Midtown, East Village, and some parts of Lower Manhattan benefit from the $52 flat rate. However, people living in Midtown, East Village, and some parts of Lower Manhattan might be relatively more indifferent to the price of taxi. Instead, they probably put more emphasis on convenience and time.

```
[1] 2.83
```

On average people travel from Manhattan pay 2.83 less with the $52 flat rate policy. Therefore, passengers overall benefit from the $52 flat rate policy.

5.3 Are taxi drivers happy when a passenger wants to go to JFK Airport from Manhattan?

Everytime I travel to New York City, I always take Yellow cabs to go around the city. It seemed to me that the cab drivers were always happy when they heard me telling them that I needed to go to the JFK Airport from Manhattan. Are taxi drivers happy when their passengers want to go to JFK Airport from Manhattan? In this section, we study the hourly wage of taxi drivers for different trips they completed, and we investigate whether taxi driver hourly wage from Manhattan to JFK Airport is higher than other trips.

```
[1] 63.29
```

The average hourly wage of taxi drivers calculated by using all trips excluding the ones going from Manhattan to JFK Airport is 63.29.

```
[1] 69.05
```

The average hourly wage of taxi drivers calculated by using trips going from Manhattan to JFK Airport is 69.05. 69.05 dollar per hour is higher than 63.29 dollar per hour, which means that on average taxi drivers driving from Manhattan to JFK Airport have an hourly wage that is about $6 higher than the hourly wage of taxi drivers doing other trips.

5.3.1 How much on average would taxi driver make on their way back from JFK Airport?

A taxi driver waiting in line to pickup passengers at JFK Airport could be directed back to anywhere in the city. Therefore, the estimated fare that a taxi driver would make on the way back from JFK is unknown. We calculate the average taxi fare

Table 5.2: 5 most popular destinations in Manhattan

Borough	Zone	num_trips	avg_fare	avg_duration
Manhattan	Times Sq/Theatre District	59419	69.80599	55.92389
Manhattan	Midtown East	40513	69.40195	47.42096
Manhattan	Murray Hill	40071	69.91174	43.66998
Manhattan	Midtown South	38890	70.11065	48.34342
Manhattan	Midtown Center	36405	69.64272	52.62410

amount that a taxi driver would get paid for a trip from JFK Airport to any part of the city.

What are the most popular drop-off locations for passengers departing from JFK Airport?

Table 5.2 shows that Times Square is the most popular destination for passengers coming from the JFK Airport in 2017!

Table 5.3: Number of Trips going to Manhattan or other boroughs from JFK Airport

Going to Manhattan	Number of Trips
0	521476
1	970366

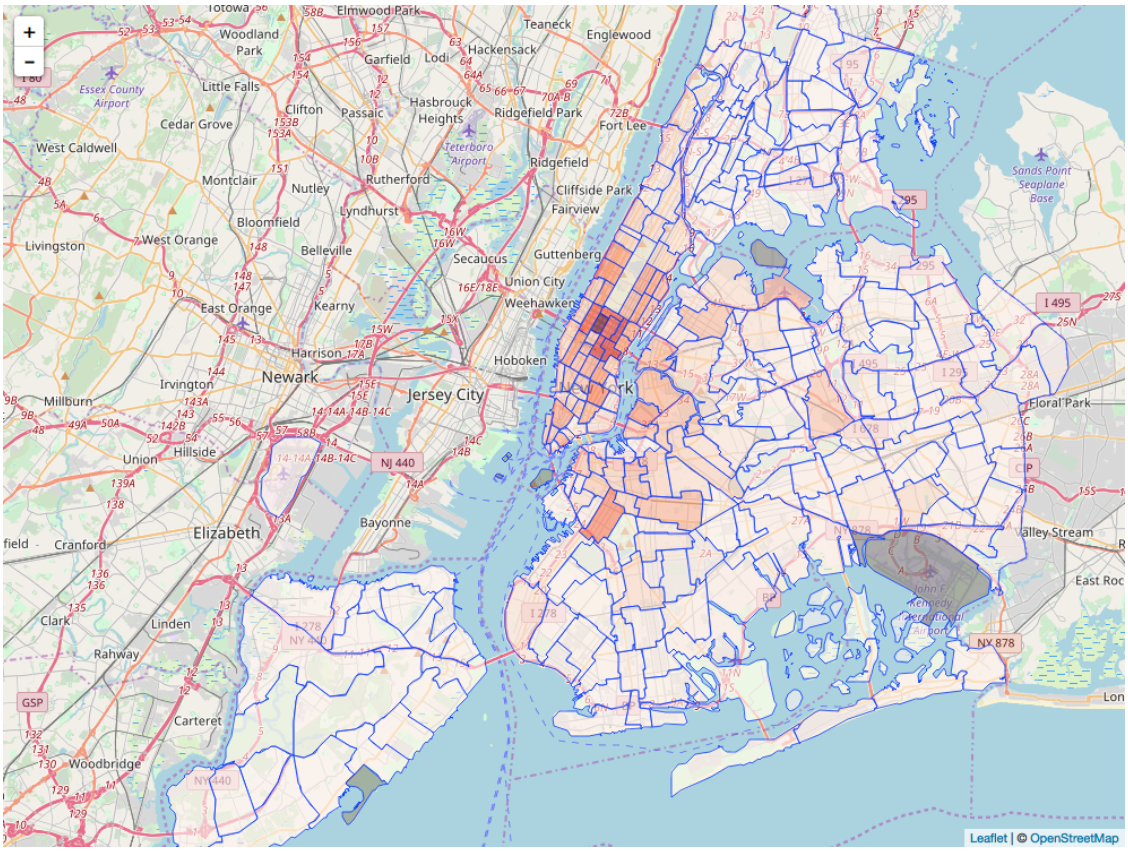


Figure 5.3: Number of trips from JFK Airport to any Taxi Zones

According to the Figure @??fig:from-jfk-num-trips), Manhattan is the most popular destination for passengers departing from JFK Airport. According to the summary, the total amount of trips from JFK Airport to Manhattan is about 0.6504482% of the total number of trips travelling from JFK Airport to all other Borough. Therefore, it is very likely for taxi drivers to get passengers who want to go to Manhattan with a flat rate of $52.

Table 5.4: 10 most popular taxi drop-off zones from JFK Airport with the corresponding average fare amount

Borough	Zone	# of Trips	Average Fare	Average Duration
Manhattan	Times Sq/Theatre District	59419	69.81	55.92389
Manhattan	Midtown East	40513	69.40	47.42096
Manhattan	Murray Hill	40071	69.91	43.66998
Manhattan	Midtown South	38890	70.11	48.34342
Manhattan	Midtown Center	36405	69.64	52.62410
Manhattan	Clinton East	35297	69.20	55.78806
Manhattan	Midtown North	34538	68.30	55.80455
Brooklyn	Park Slope	27219	60.96	45.75234
Manhattan	East Village	26595	66.98	45.45684
Manhattan	Upper West Side South	24723	69.95	50.87777

What's the average fare to each drop-off zone from JFK Airport?

We can use a map to visualize the distribution of average fare amount needed to travel from JFK Airport to any taxi zone in New York City.

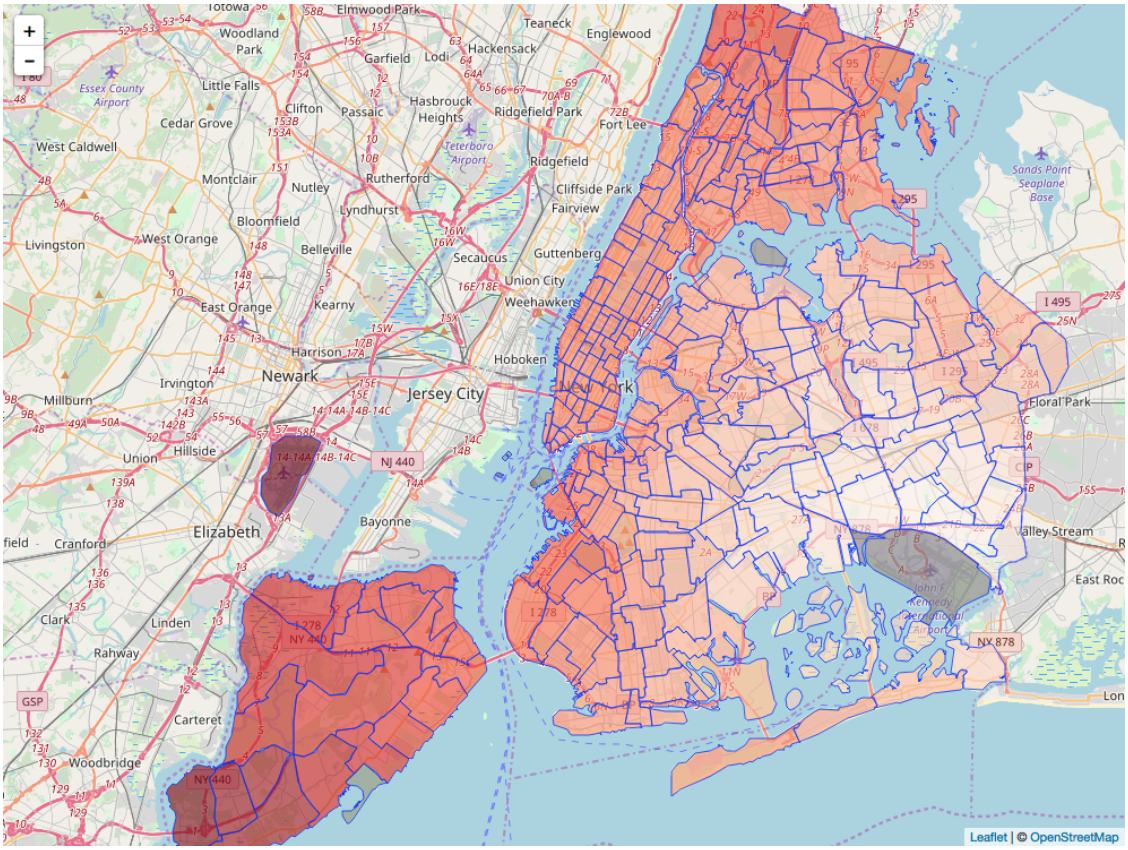


Figure 5.4: Zones that cost more than the 52 US Dollar flat rate

As we expected, the red shades are smoothly distributed, since taxi zones that are futher away should cost more to get there.

How much on average would taxi driver make on their way back from JFK Airport?

```
[1] 62.7
```

On average, taxi drivers would get paid for on average 62.7 dollar for a trip from the JFK Airport to any taxi zone in New York City, and on average trips from JFK Airport last 44 minutes.

```
[1] 63.15
```

The average hourly wage of taxi drivers calculated by using all trips excluding the ones departing from JFK Airport is 63.15 dollar per hour.

```
[1] 94.65
```

The average hourly wage of taxi drivers calculated by using trips departing from JFK Airport is 94.65 dollar per hour. 94.65 dollar per hour is higher than 63.15 dollar per hour, which means that on average taxi drivers going from JFK Airport to any taxi zones in New York City have an hourly wage that is more than $30 higher than the hourly wage of taxi drivers doing other trips.

How much more do taxi drivers make on average for a round trip to and from JFK Airport, comparing to any other trips?

On average taxi drivers driving from Manhattan to JFK Airport make $6 more every hour than taxi drivers doing other trips. On average taxi drivers going from JFK Airport to any taxi zones in New York City make $30 more every hour than taxi drivers doing other trips. Overall, a taxi driver doing a round trip to and from JFK Airport make $36 more every hour, comparing to a taxi driver doing any other trips. In this case, a round trip to and from JFK Airport is worthwhile, and that's why taxi drivers should feel happy when pick up a passenger in Manhattan and is told that he or she wants to go to JFK Airport.

5.4 Recommendations to New York City Taxi Fare & Limousine Commission

In this chapter, we have found the reason why taxicab drivers feel happy when they pick up passengers who want to go to JFK Airport in Manhattan. Even though $52 is

lower than the average amount of fare that taxi drivers would have made without the flat rate, it still induces an higher than average hourly wage, so it does not disincentives drivers to go to JFK Airport from Manhattan. Therefore, we suggest the TLC to keep this flat rate so that passengers do not have any uncertainty in cost and they are more willing to take a taxi to travel to JFK Airport from Manhattan.

Chapter 6

Conclusion

In this Honors thesis, we present a more efficient and user-friendly way for **R** users to retrieve trip record of both taxi and other ride-sharing services, such as Uber and Lyft, in New York City.

By analyzing trip records of New York City's yellow taxi, we found answers to questions that are of obvious interest to taxi drivers, passengers, and TLC officials.

We found which taxi zones have passengers who offer the highest percent of tips, and we showed that taxi drivers do get compensated more during rush hours. We helped passengers to know the average time it takes to go to one of the three airports in New York City so that passengers can plan their trips accordingly. We also found that the $52 flat rate between Manhattan and JFK Airport is beneficial for the passengers, because it is cheaper than the average amount of fare that passenger would need to pay without the flat rate. We have also shown that the flat rate does not discourage drivers, even though taxi drivers would have be paid more without the flat rate.

We suggest passengers to use our Shiny App to choose a pick up zone of their interest and then decide when is the most favorable time for them to travel to any airport in New York. We recommend New York City TLC to modify the fare on rainy or snowy

days to incentive taxicab drivers to pick up more trips in order to make street-hail service more affordable on rainy days for passengers. We also suggest the TLC to keep the $52 flat rate between Manhattan and JFK Airport so that passengers do not have any uncertainty in cost and they are more willing to take a taxi to travel to JFK Airport from Manhattan.

6.1 Future Research

We would love to investigate the sharp decline in the consumption of NYC yellow cab after e-hail services were introduced into the NYC ride-hail market. By looking into the patterns in market shares, it might be possible for one to predict the future market share distribution and find out what features of ride-hail transportation are the ones that affect market share the most.

We also want to study what the impact of introducing new GPS and entertainment system is on the number of rides. The global product and marketing director at Verifone, Jason Gross, said that, "We like to say that we provide what Uber says it provides." With the raised expectation among rides caused by Uber and Lyft, yellow taxi industry need to respond quickly (Hawkins, 2016). How does the market react to the newly installed entertainment system? Has the market share of yellow cab rebounded since the installation of the entertainment system in 2016?

As mentioned in Chapter 3, we also want to study the correlation between number of trips and average percent tips in each taxi zone. If more taxi-zone-specific data is provided, we could find out the true correlation between the two variables.

Appendix A

Utility Function

This utility function was written to shorten the source code in ETL `etl_extract.etl_nyctaxi()` function. It takes in '`url`, `year`, `n` (number of observations), and `names` (which are the names CSV data files), and create a list of raw data directories.

```
download_nyc_data <- function(obj, url, years, n, names, ...) {
  url <- paste0(url, "?years=", years, "&$limit=", n)
  lcl <- file.path(attr(obj, "raw"), names)
  downloader::download(url, destfile = lcl, ...)
  lcl
}
```

Appendix B

Data Dictionary – Yellow Taxi

All variables used in data analysis of yellow taxi data are listed in this data dictionary.

Data Dictionary – Yellow Taxi Trip Records September 28, 2015 Page 1 of 1

This data dictionary describes yellow taxi trip data. For dictionaries describing green taxi and FHV data, please visit http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. **1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.**
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
Pickup_longitude	Longitude where the meter was engaged.
Pickup_latitude	Latitude where the meter was engaged.
RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. **Y= store and forward trip** **N= not a store and forward trip**
Dropoff_longitude	Longitude where the meter was disengaged.
Dropoff_ latitude	Latitude where the meter was disengaged.
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
MTA_tax	$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.

Figure B.1: Data Dictionary – Yellow Taxi Trips Records

Appendix C

Freedom of Information Law Request

We submitted this FOIL Request to seek answers of the questions listed below to better analyze the yellow taxi data.

Taxi & Limousine Commission

Meera Joshi
Commissioner

Christopher Wilson
Deputy Commissioner/General Counsel
Office of Legal Affairs
33 Beaver Street, 22nd Floor
New York, NY 10004

+1 212 676 1102 fax

FOIL REQUEST FORM

Taxi and Limousine Commission
Office of Legal Affairs
33 Beaver Street, 22nd Floor
New York, New York 10004
Attn: Records Access Officer

FROM: ***(Please print your name, address, telephone # and email address)***

Our email address: FOIL@tlc.nyc.gov **Your email address:** wli37@smith.edu

I request the following record(s) under the Freedom of Information Law:
Please reasonably describe the record(s) you are requesting to allow us to identify any responsive document(s).

I have a few questions about some of the variables, and could you please help mw to answer them:
1. negative fare amount: I saw some negative fare amounts. Could you please tell me what situations lead to negative fare amount data entries? Broken meters?
2. zero fare amount: Some trips have zero fare amounts. Does it mean that the taxi driver cancelled the trip? Or does it mean that the passengers refused to pay the taxi fare?
3. negative tip amount: I saw some negative tip amounts. Could you please tell me what situations lead to negative tip amount data entries? Broken meters?
4. huge tolls amount: I saw some huge tolls amount. Do you know what might have caused that? One of the trips has a toll amount that is more than $450.
5. passenger count: Did taxi drivers manually entered this data?
6. Do smaller taxi (5 seats) and bigger taxis (6 or 7 seats) have the same rate of fare?
7. Do green taxi and yellow taxis have the same rate of fare?

Please state the reason for your request: (optional)

I am Wencong Li (Priscilla), a senior at Smith College, MA. I am currently working on my thesis and I am using the TLC taxi data to do data analysis.

WENCONG LI
Signature

02/06/18
Date

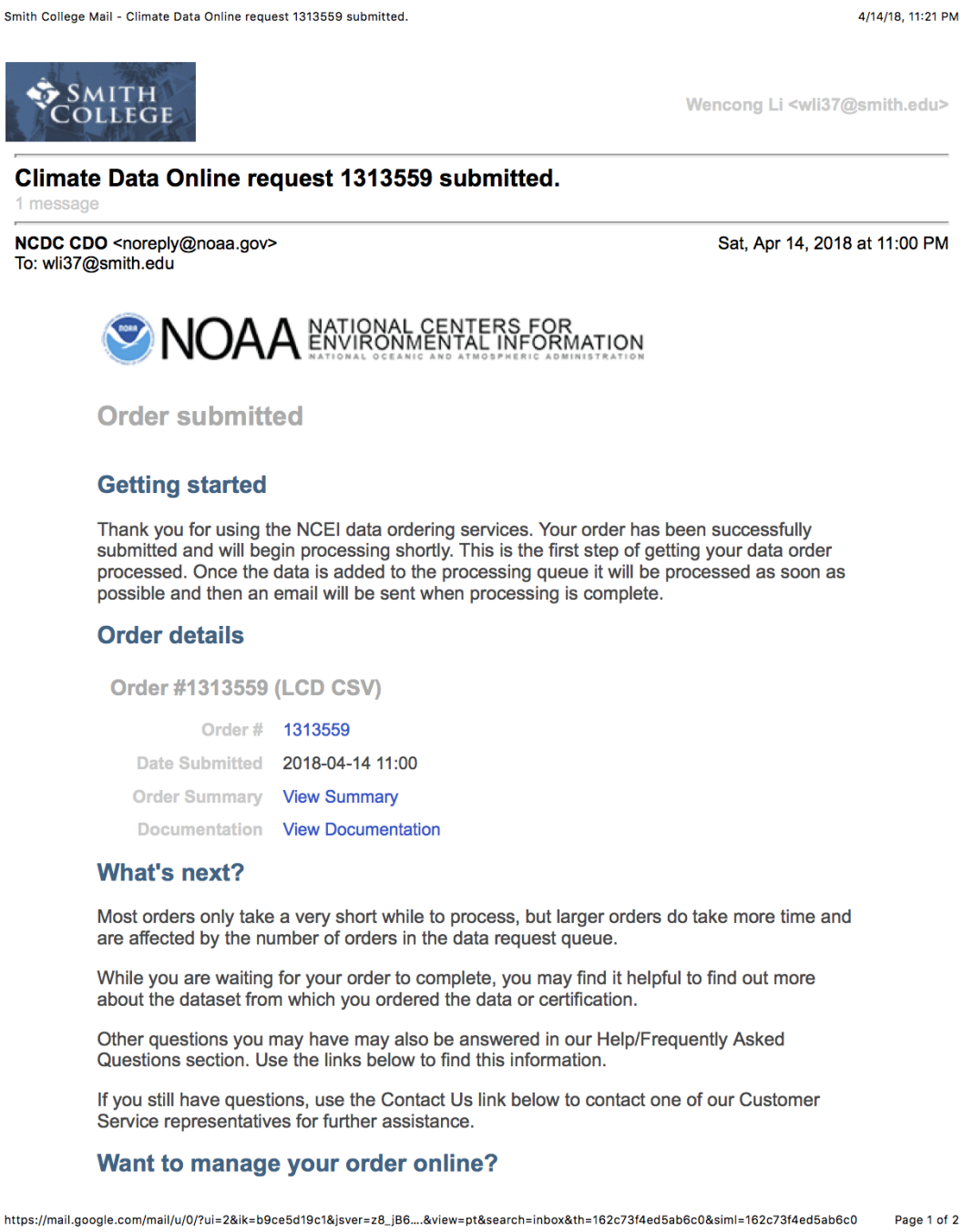
This is a Freedom of Information Law ("FOIL") request. As such, your request will be considered under the Public Officer's Law, Article 6, Section 84 et seq. Subject to the provisions of this article, the Taxi and Limousine Commission, within five business days of the receipt of a written request for a record reasonably described, shall furnish a written acknowledgement of the receipt of such request.

Figure C.1: Freedom of Information Law Request

Appendix D

NOAA Climate Data Request

We submitted the NOAA Climate Data Request to get access to 2017 New York City weather data.



Smith College Mail - Climate Data Online request 1313559 submitted. 4/14/18, 11:21 PM

SMITH COLLEGE

Wencong Li <wli37@smith.edu>

Climate Data Online request 1313559 submitted.

1 message

NCDC CDO <noreply@noaa.gov> Sat, Apr 14, 2018 at 11:00 PM
To: wli37@smith.edu

NOAA NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Order submitted

Getting started

Thank you for using the NCEI data ordering services. Your order has been successfully submitted and will begin processing shortly. This is the first step of getting your data order processed. Once the data is added to the processing queue it will be processed as soon as possible and then an email will be sent when processing is complete.

Order details

Order #1313559 (LCD CSV)

Order #	1313559
Date Submitted	2018-04-14 11:00
Order Summary	View Summary
Documentation	View Documentation

What's next?

Most orders only take a very short while to process, but larger orders do take more time and are affected by the number of orders in the data request queue.

While you are waiting for your order to complete, you may find it helpful to find out more about the dataset from which you ordered the data or certification.

Other questions you may have may also be answered in our Help/Frequently Asked Questions section. Use the links below to find this information.

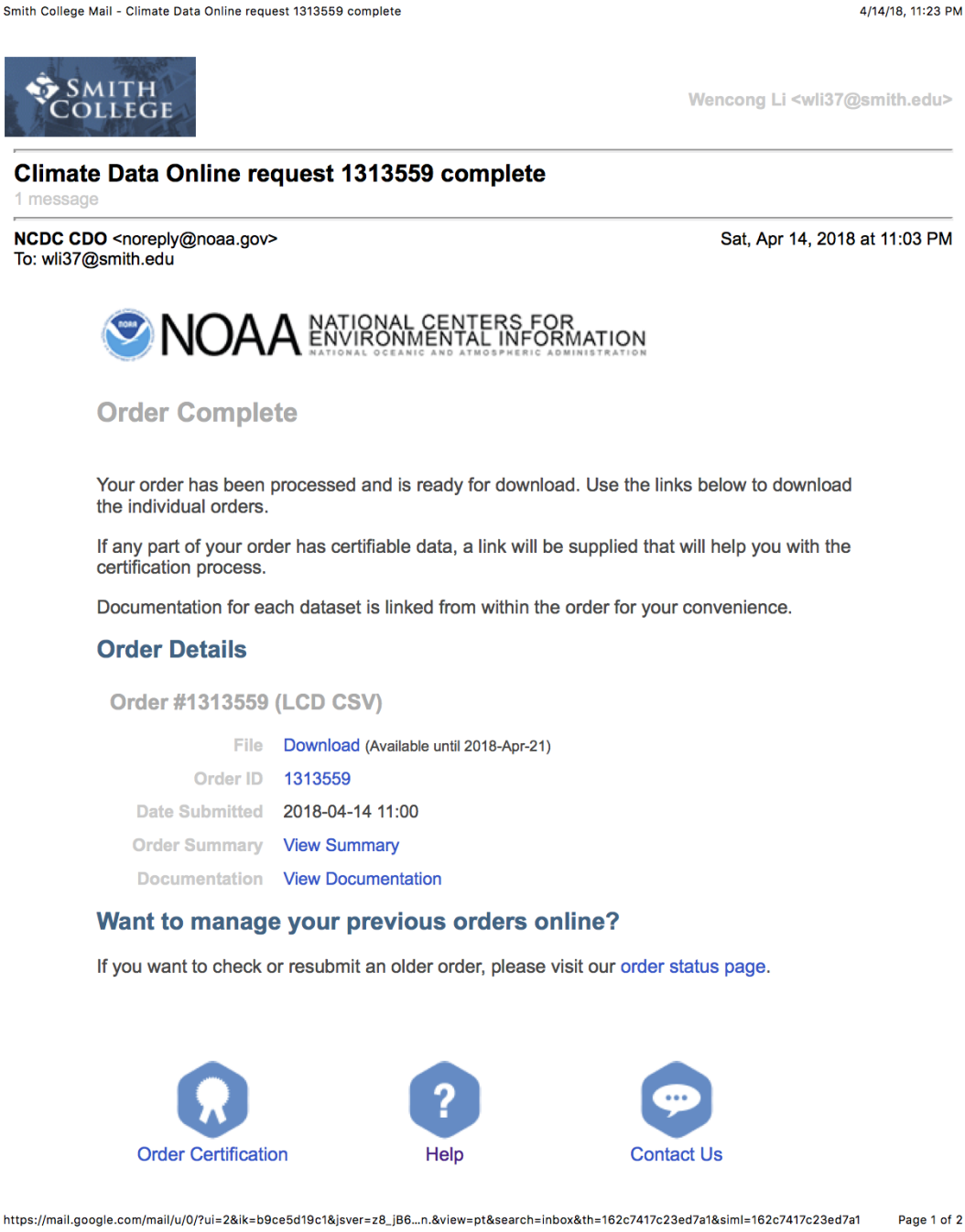
If you still have questions, use the Contact Us link below to contact one of our Customer Service representatives for further assistance.

Want to manage your order online?

https://mail.google.com/mail/u/0/?ui=2&ik=b9ce5d19c1&jsver=z8_jB6....&view=pt&search=inbox&th=162c73f4ed5ab6c0&siml=162c73f4ed5ab6c0 Page 1 of 2

Figure D.1: NOAA Climate Data Request

The NOAA Climate Data Order Completion Notification grants me rights the access to 2017 New York City weather data.



Smith College Mail - Climate Data Online request 1313559 complete 4/14/18, 11:23 PM

SMITH COLLEGE

Wencong Li <wli37@smith.edu>

Climate Data Online request 1313559 complete

1 message

NCDC CDO <noreply@noaa.gov> Sat, Apr 14, 2018 at 11:03 PM
To: wli37@smith.edu

NOAA NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Order Complete

Your order has been processed and is ready for download. Use the links below to download the individual orders.

If any part of your order has certifiable data, a link will be supplied that will help you with the certification process.

Documentation for each dataset is linked from within the order for your convenience.

Order Details

Order #1313559 (LCD CSV)

File	Download (Available until 2018-Apr-21)
Order ID	1313559
Date Submitted	2018-04-14 11:00
Order Summary	View Summary
Documentation	View Documentation

Want to manage your previous orders online?

If you want to check or resubmit an older order, please visit our order status page.

Order Certification Help Contact Us

https://mail.google.com/mail/u/0/?ui=2&ik=b9ce5d19c1&jsver=z8_jB6...n.&view=pt&search=inbox&th=162c7417c23ed7a1&siml=162c7417c23ed7a1 Page 1 of 2

Figure D.2: NOAA Climate Data Order Compeletion

References

Baumer, B. S. (2017). A grammar for reproducible and painless extract-transform-load operations on medium data. Retrieved from https://arxiv.org/abs/1708.07073

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., & Horton, N. J. (2014). R markdown: Integrating a reproducible analysis tool into introductory statistics. *TISE*.

Cheng, J., Karambelkar, B., & Xie, Y. (2017). *Leaflet: Create interactive web maps with the javascript 'leaflet' library*. Retrieved from https://CRAN.R-project.org/package=leaflet

Dowle, M., & Srinivasan, A. (2017). *Data.table: Extension of 'data.frame'*. Retrieved from https://CRAN.R-project.org/package=data.table

Environmental Information Staff, N. C. for. (n.d.). Climate Data Online. National Centers for Environmental Information. Retrieved from https://www.ncdc.noaa.gov/cdo-web/

FiveThirtyEight. (2015, September). Uber TLC FOIL Response. Retrieved from https://github.com/fivethirtyeight/uber-tlc-foil-response

Furfaro, D., Cohen, S., & Fears, D. (2016, December). NYC is already tired of Christ-

mas and Donald Trump. New York Post. Retrieved from https://nypost.com/2016/12/01/nyc-is-already-tired-of-christmas-and-donald-trump/

Grolemund, G., & Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, *40*(3), 1–25. Retrieved from http://www.jstatsoft.org/v40/i03/

Hawkins, A. J. (2016, September). Yellow taxis have a new weapon in their war against uber: Gadgets. The Verge. Retrieved from https://www.theverge.com/2016/9/26/13035642/nyc-taxi-cab-android-touchscreen-tablet-verifone

Henry, L., & Wickham, H. (2018). *Rlang: Functions for base types and core r and 'tidyverse' features*. Retrieved from https://CRAN.R-project.org/package=rlang

Hu, W. (2017, January). Yellow Cab, Long a Fixture of City Life, Is for Many a Thing of the Past. The New York Times. Retrieved from https://www.nytimes.com/2017/01/15/nyregion/yellow-cab-long-a-fixture-of-city-life-is-for-many-a-thing-of-the-past.html

Jaffe, E. (2014, October). Why New Yorkers Can't Find a Taxi When It Rains. CITYLAB. Retrieved from https://www.citylab.com/environment/2014/10/why-new-yorkers-cant-find-a-taxi-when-it-rains/381652/

Li, W. (2018, February). FOIL request. NYC TLC.

Li, W. P., Baumer, B., & Trang Le. (2017). *Nyctaxi: Accessing new york city taxi data*. Retrieved from http://github.com/beanumber/nyctaxi

R Special Interest Group on Databases (R-SIG-DB), Wickham, H., & Müller, K. (2018). *DBI: R database interface*. Retrieved from https://CRAN.R-project.

org/package=DBI

Reaney, P. (2009, June). New York Drivers Named Most Aggressive, Angry in U.S. Reuters. Retrieved from https://www.reuters.com/article/us-driving-roadrage-life/new-york-drivers-named-most-aggressive-angry-in-u-s-idUSTRE55F1J720090616

Schneider, T. W. (2015, November). Analyzing 1.1 Billion NYC Taxi and Uber Trips, with a Vengeance. Todd W. Schneider. Retrieved from http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

Solomon, N. (n.d.). *Thesisdown: An updated r markdown thesis template using the bookdown package*.

Staff, C. (n.d.). The Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/index.html

Staff, H. (2018, January). Uber's 4 Basic Level of Service. HyreCar. Retrieved from https://hyrecar.com/blog/difference-between-uber-cars/

Staff, N. O. (2015a). LYFT Data. NYC OpenData. Retrieved from https://data.cityofnewyork.us/Transportation/LYFT-Data/juxc-sutg/data

Staff, N. O. (2015b). Uber Trips NYC 2016. NYC OpenData. Retrieved from https://data.cityofnewyork.us/Transportation/Uber-Trips-NYC-2016/gt3n-7ri6/data

Staff, N. T. (2009a). TLC Aggregated Reports. NYC Taxi & Limousine Commission. Retrieved from http://www.nyc.gov/html/tlc/html/technology/aggregated_data.shtml

Staff, N. T. (2009b). TLC Trip Record Data. NYC Taxi & Limousine Commission.

Retrieved from http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

Staff, N. T. (2009c). TLC Trip Record Data. NYC Taxi & Limousine Commission. Retrieved from http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

Staff, N. T. (2009d). Your guide to Boro Taxis. NYC Taxi & Limousine Commission. Retrieved from http://www.nyc.gov/html/tlc/html/passenger/shl_passenger.shtml

Staff, N. T. (2018). NYC Taxi & Limousine Commission. NYC Taxi & Limousine Commission. Retrieved from http://www.nyc.gov/html/tlc/html/home/home.shtml

Staff, N. T. (n.d.). NYC tlc Taxicab Rate of Fare. NYC TLC. Retrieved from http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml

Staff, U. (2009, March). Uber Technologies Inc. Retrieved from https://www.uber.com

Sugar, R. (2017, January). Uber and Lyft cars now outnumber yellow cabs in NYC 4 to 1. Curbed New York. Retrieved from https://ny.curbed.com/2017/1/17/14296892/yellow-taxi-nyc-uber-lyft-via-numbers

Uber Moves New York City. (2015, November). Uber. Retrieved from https://www.uber.com/cities/new-york/

Vsevolod Salnikov, A. N., Renaud Lambiotte. (2015). OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs.

Whitford, E. (2017, October). Daily Uber Trips Have Officially Outstripped Taxi Trips. Gothamist. Retrieved from http://gothamist.com/2017/10/13/uber_

taxis_nyc.php

Wickham, H. (2018). *Stringr: Simple, consistent wrappers for common string operations*. Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H., Francois, R., Henry, L., & Müller, K. (2017). *Dplyr: A grammar of data manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., Hester, J., & Francois, R. (2017). *Readr: Read rectangular text data*. Retrieved from https://CRAN.R-project.org/package=readr

Xie, Y. (2018). *Knitr: A general-purpose package for dynamic report generation in r*. Retrieved from https://CRAN.R-project.org/package=knitr

Zhang, W. (2017, May). Improving access to open-source data about the nyc bike sharing system (Citi Bike). Smith College.