Estimate trip dutation by MLE

Hukai Luo

19 November 2018

1 Introduction

The goal of this project is to predict the *duration of taxi rides in NYC* based on features like trip coordinates or pickup date and time.

Since there are lots of parameters when we are estimating the trip duration, we will first study and visualise the original data, view the relations between the trip_duration and some parameters, decide which parameters should be considered in our Finally, we will choose some important parameters to curve fit the data using the Maximum likelihood estimation.

First of all, let's load the given data and find the parameters of the data

```r
library('tibble')
library('data.table')
library('dplyr')
train <- as.tibble(fread("/Users/luohukai/Documents/GitHub/final-project-hul17011/train.csv"))
test <- as.tibble(fread("/Users/luohukai/Documents/GitHub/final-project-hul17011/test.csv"))
summary(train)                  #get data structure
```

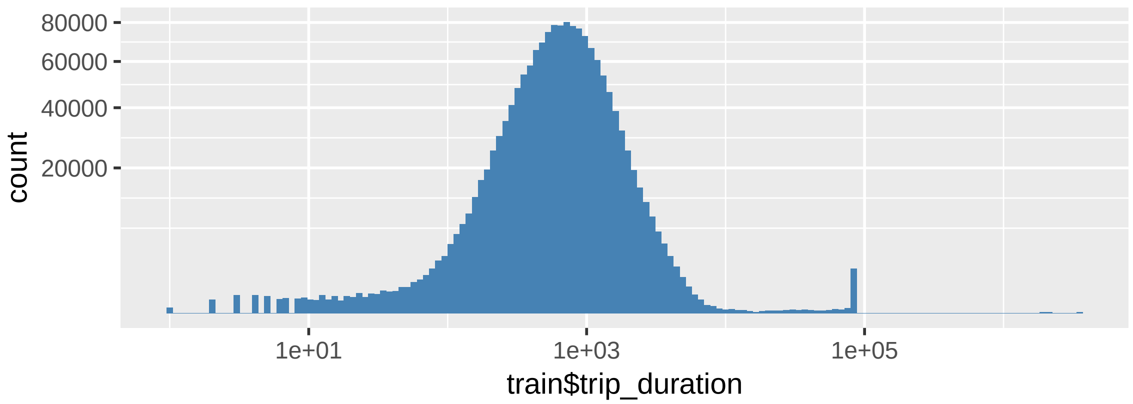
We find the data contains several factors: **vender_id** takes only 1 or 2 which represents two taxi companies; **pickup_datetime**; **dropoff_datetime**; **passenger_count**; **pickup_longitude**; **pickup_latitude**; **dropoff_longitude**; **dropoff_latitude**; **store_and_fwd_flag**; **trip_duration** which is measured in seconds.
In order to make the data easy to use, we will make some change to the data,just make it easy to use.

```r
library('lubridate')
train <- train %>%
  mutate(pickup_datetime = ymd_hms(pickup_datetime),
         dropoff_datetime = ymd_hms(dropoff_datetime),
         vendor_id = factor(vendor_id),
         passenger_count = factor(passenger_count))
```

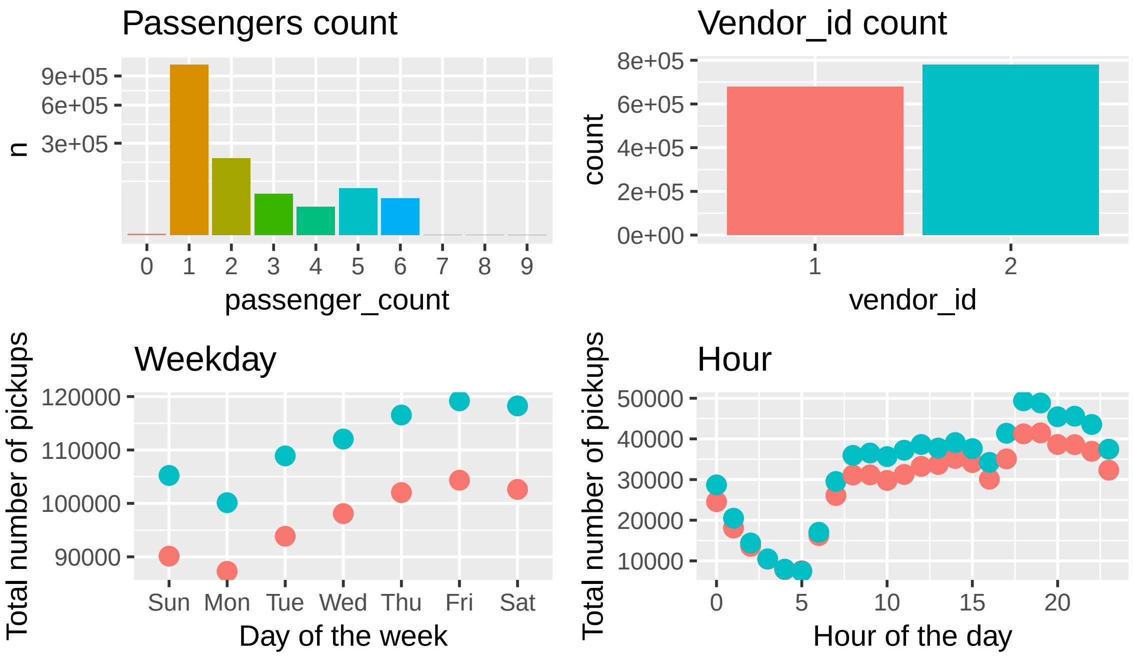
2 Plot by single parameter

Now in order for us to get a better understanding of the data, we will begin by having a look at the distributions of the individual data features. First of all, let's plot the target feature trip_duration.

```r
library('ggplot2')
p1 <- ggplot(train, aes(train$trip_duration)) +
  geom_histogram(fill = "steelblue", bins = 150) +
  scale_x_log10() + scale_y_sqrt()
p1
```



Comments: Most trips will ends in nearly 1000 seconds, but there will also be some exceptions.
Then we can also plot the distribution of passenger_count,Vendor_id,day of the week, hour of the day



Comments: Most trips only have 1 passenger; Thursday,Friday and Satuaday are the most busy days; there is a strong dip during the early morning hours and another dip around 4pm.

3 Relations

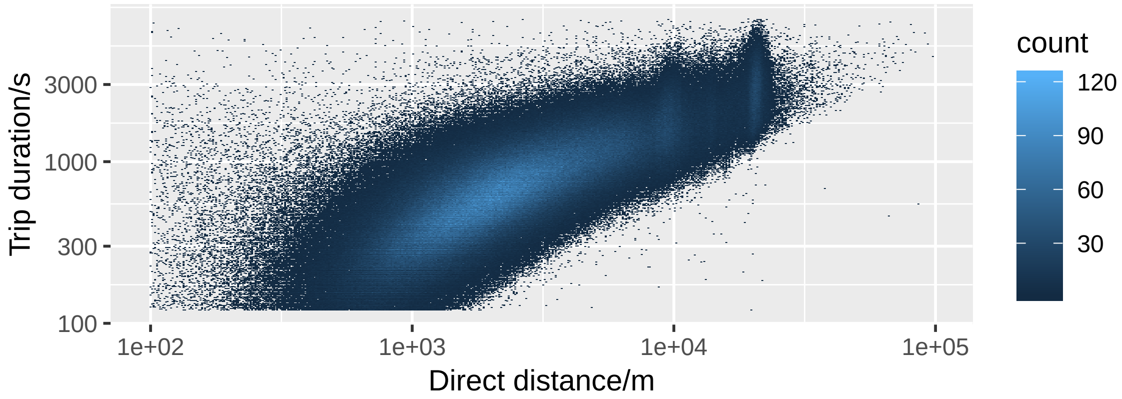
While the previous section looked primarily at the distributions of the individual features, here we will examine in more detail how those features are related to each other and to our target trip_duration. In this project, we will assume that the trip_duration is only related to **Trip distance**, **passenger numbers**, **vender_id**, **day of the week**, **hour of the day**.

3.1 Trip distance vs trip_duration

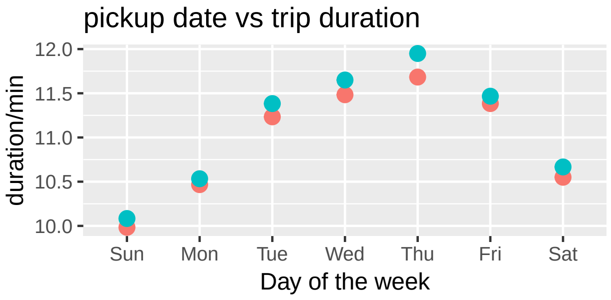
First, we need to calculate the exact trip distance by the pickup and dropoff location, this is a very important parameter in our estimation.

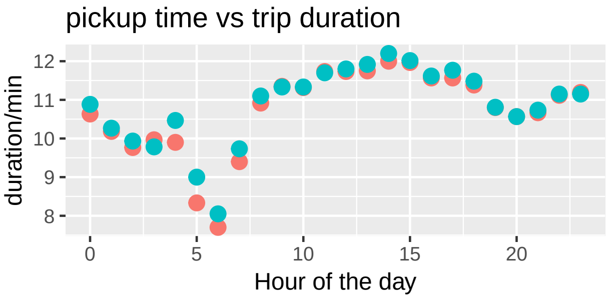
```
library('geosphere')
pick_coord <- train %>%
  select(pickup_longitude, pickup_latitude)
drop_coord <- train %>%
  select(dropoff_longitude, dropoff_latitude)
train$dist <- distCosine(pick_coord, drop_coord)
```

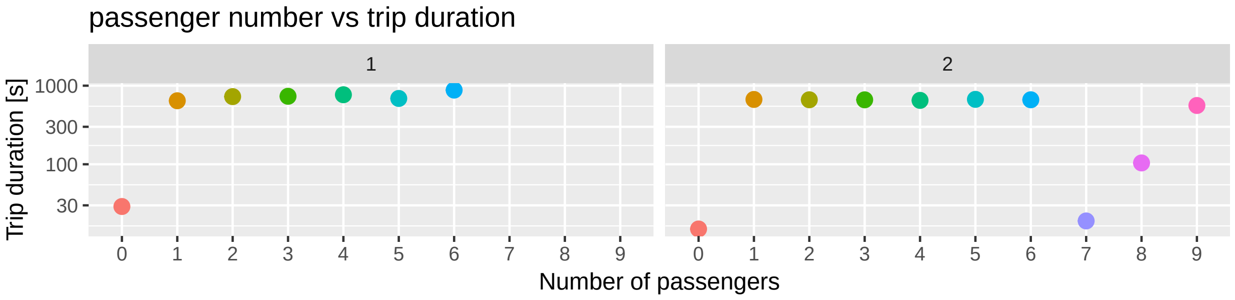
Then plot the Trip distance vs trip_duration distribution.



3.2 Pickup date/time and Passenger numbers vs trip_duration







From this plot, we can find the trip_duration doesn't have a strong relationship with the passenger numbers, the difference in the picture may only reveal the impact of different distance. For those passenger numbers = 0, 7, 8, 9, we don't think that they are reasonable, so we will delete those abnormal data.

4 Prediction by Maximum likelihood estimation

In this project, we assume the trip_duration distribution function has three parameters: trip_distance d, pickup date w, pickup time t. We estimate the duration function has the following form.

$$T = \frac{dis}{\bar{V}}(1+r) * Date[wday] * Hour[hour]$$

$\bar{V}$ is the average speed, dis is the distance by the pickup and dropoff location, r is unknown distribution function which we need to estimate using the Maximum likelihood estimation, $DATE$ and $HOUR$ are functions based on the pickup_date and pickup_hour.

First of all, let's delete some abnormal data, then calculate the average driving speed $\bar{V}$

```
train$speed <- train$dist/train$trip_duration*3.6
train$wday <- wday(train$pickup_datetime, label = FALSE)
train$hour <- hour(train$pickup_datetime)
train <- train %>%
  filter(trip_duration < 7600 & trip_duration > 40) %>% #delete abnormal trip_duration time: T>2hours a
  filter(dist > 100 & dist < 100e3) %>%                 #delete abnormal distance: d>100km and d<100m
  filter(speed < 100 & speed > 1)                       #deleta speed which is too fast or too slow
average_speed <- mean(train$speed)
average_speed
```

```
## [1] 14.52126
```

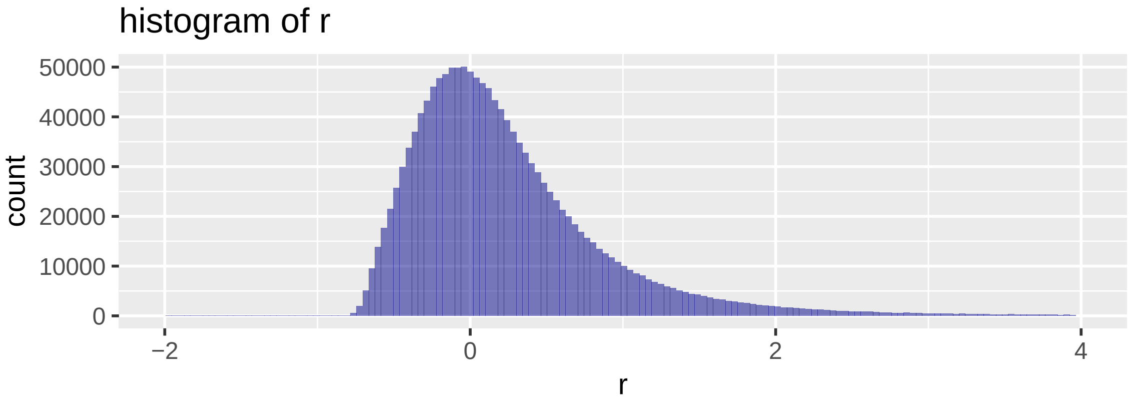
We get the average speed $\bar{V} = 14.52126$, it's not hard to calculate$\frac{Distance}{\bar{V}}$
Then, define the $DATE$ and $HOUR$ function below, we can use them to generate the r distribution:

```
wdaydata <- train %>%                                     # get pickup date median duration
  mutate(wday = wday(pickup_datetime, label = FALSE)) %>%
  group_by(wday) %>%
  summarise(median_duration = median(trip_duration))
hpickdata <- train  %>%                                   # get pickup hour median duration
  mutate(hour = hour(pickup_datetime)) %>%
  group_by(hour) %>%
  summarise(median_duration = median(trip_duration))
DATE <- function(x){wdaydata$median_duration[x]/mean(wdaydata$median_duration)}
HOUR <- function(x){hpickdata$median_duration[x+1]/mean(hpickdata$median_duration)}
```

After that, we got the estimated r data

$$r = \frac{T\bar{V}}{Date[wday] * Hour[hour] * Distance} - 1$$

```
n <- function(data){
  (data$trip_duration/(DATE(data$wday)*HOUR(data$hour))-data$dist/average_speed*3.6)/(data$dist/average_
}
error <- n(train)
p10 <- ggplot(data.frame(x=error),aes(x=x)) +             # histogram of distribution r
  geom_histogram(fill="darkblue", bins = 150,position="identity", alpha=0.5)+
  xlim(-2,4)+labs(title = "histogram of r", x = "r")
p10
```



There will be lots of methods to estimate the r distribution function given the data above. In this project, we assume r is a normal distribution, with density function below

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Compute log-likelihood function:

$$l(x|\mu,\sigma) = -\frac{n}{2} ln(2\pi) - nln\sigma - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$$

derivative of μ and σ:

$$\frac{\partial l}{\partial \mu} = -\sum_{i=1}^{n} \frac{x_i-\mu}{\sigma^2} = 0$$

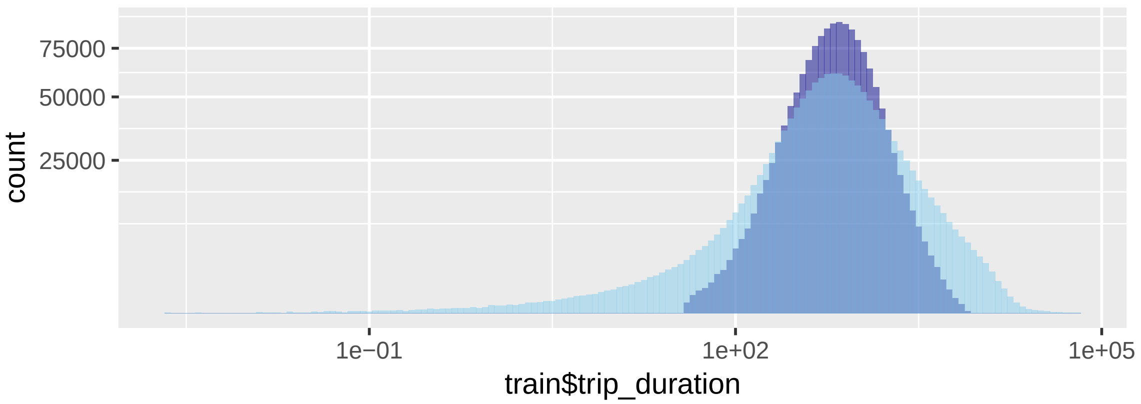
$$\mu = \sum_{i=1}^{n} \frac{x_i}{n}$$

$$\frac{\partial l}{\partial \sigma} = -\frac{n}{\sigma} + \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{\sigma^3} = 0$$

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{n}$$

when we get the r distribution, let's calculate the estimated trip_duration **T**

```
sd <- sd(n(train))
mean <- mean(n(train))
T <- function(data,mean,sd){
  res <- 0
  for(i in 1:length(data$dist)){
    res[i] <- data$dist[i]/average_speed*3.6*(1+rnorm(1,mean,sd))*(DATE(data$wday[i])*HOUR(data$hour[i]
  }
  res
}
train$estimate <- T(train,mean,sd)                 # estimated trip duration time
p9 <- ggplot(train) +                              # plot the true ditribution and estimated distributio
  geom_histogram(aes(train$trip_duration),fill="darkblue", bins = 150,position="identity", alpha=0.5) +
  geom_histogram(aes(train$estimate),fill="skyblue", bins = 150,position="identity", alpha=0.5) +
  scale_x_log10() +
  scale_y_sqrt()
p9
```



5 Comments and Improvement

In the plot above, we plot the true trip_duration distribution and the trip_duration distribution generated by our estimate. In the general shape, the estimated results fits good, but in the area far from the mean, our estimation will generate much more results then it should be.

In order to get a more accurate result, maybe we need to **examine potential outliers** like the rainy days, the snowy days to reduce the standard deviation of r distribution. Also, we can make **a better assumption of r** since its graph has a positive skewness, not simply assume it become a normal distribution.