Miners (Zidong Xu, Fangping Cao, Jieqian Liu, Zheyuan Hu, Jieyi Sun)
Dr. Purna Gamage
ANLY 511
December 6, 2021

Data Analysis on Bike-sharing System in Chicago

Content

Introduction

Not everyone can be a regular bike commuter. Many people enjoy biking but don’t own a bike. The appearance of bike-sharing solved these problems. The concept of bike-share programs is aimed at providing a fun, healthy, low-cost transportation option for locals and visitors. A bike-sharing system is a shared transport service in which bikes are made available for shared use to individuals in a short term for a price or free. In recent years, the bike-sharing system has become more and more popular in an increasing number of cities all over the world. Therefore, improving the bike-sharing system based on the previous data is meaningful and necessary.

How do bike-share programs work? Bikes and stations are the most important physical elements of any bike-sharing system. A bike-sharing system usually provides several rideable types of bikes for customers to choose from. Each station has a number of docks used to store and lock bikes. For profitable bike-sharing companies or runners, membership and usage fees are both important factors of profitability. Bike-share systems generally have these two types of user fees. One is the flat fee which can be divided into longer-term (membership users) and shorter-term (casual users). Flat fee grants access to the shared bikes for a specific period of time. Another is a usage fee based on the amount of time or riding distance that users actually spend on the bike.

A good bike-sharing system is expected to not only provide users the rideable types of bikes they need, rationale charging standards and proper station arrangement, but also make bike-sharing companies profitable.

In light of all these, the following questions were formulated:

1. Will the peak period of station usage be related to whether it is a working day? If not, what the period of station usage depends on?
2. Can we predict the usage duration and riding distance with the probabilities obtained from large samples?
3. Can we predict the rideable type through the time of using the bikes and the stations?
4. Does the membership affect the usage of time?

Methodology and Results:

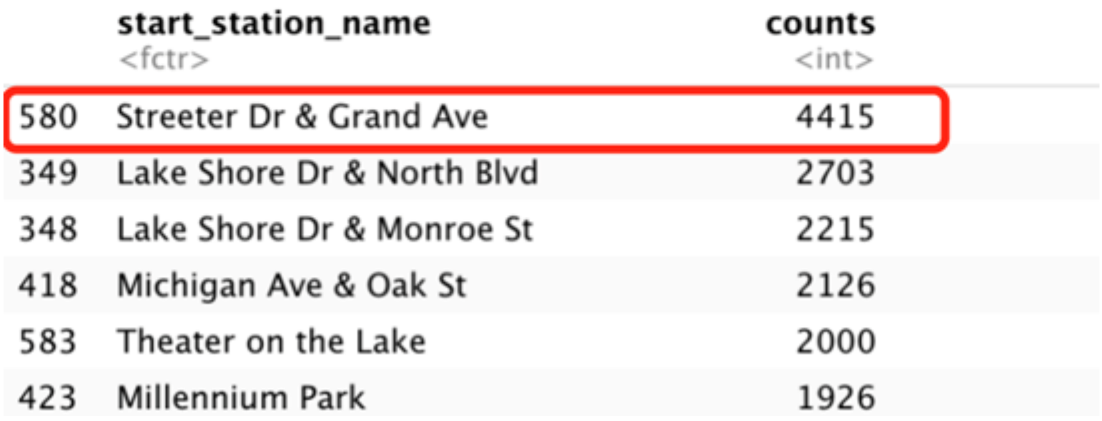
Dataset used in the study

In this project, we downloaded the record data set of the bike-sharing system in Chicago from Kaggle and sorted out the data for the first week of July 2021. The original data set can be found in: https://www.kaggle.com/daudewey/case-study-1-bikeshare. The data set provided us with the information of rideable type, usage time ('started_at' and 'ended_at'), station ('start_station_name', 'start_station_id', 'end_station_name', 'end_station_id') and location ('start_lat', 'start_lng', 'end_lat', 'end_lng') and users ('member_casual'). Based on the record data set, we proposed four statistical questions above and explained them using relevant knowledge of Chi square test, the law of large numbers, central limit theorem, Naïve Bayes, and the t-test method.

1. Research Question 1: Will the peak period of station usage be related to whether it is a working day? If not, what the peak period depends on?

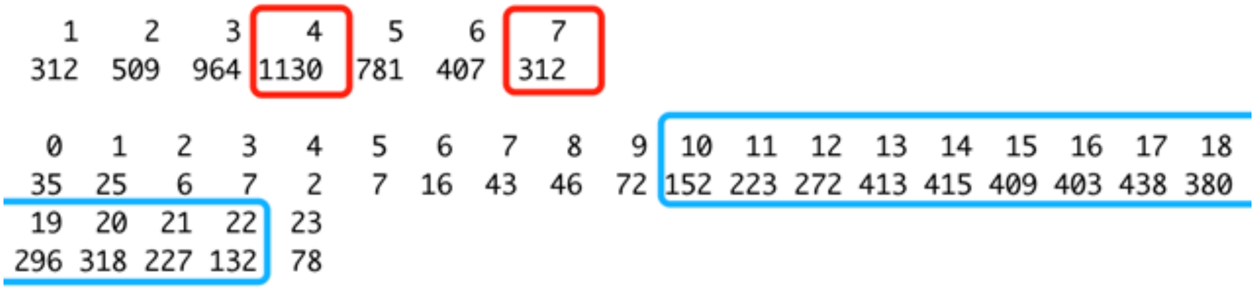
1.1 Data processing

The number of times each station used was count, and sorted in decrease order as below:



	start_station_name	counts
	<fctr>	<int>
580	Streeter Dr & Grand Ave	4415
349	Lake Shore Dr & North Blvd	2703
348	Lake Shore Dr & Monroe St	2215
418	Michigan Ave & Oak St	2126
583	Theater on the Lake	2000
423	Millennium Park	1926

We selected the most popular stations for further research. We also counted the frequency of use of the "Streeter Dr & Grand Ave" station in days, hours, and minutes from July 1 to July 7. The results were shown in the figure below.



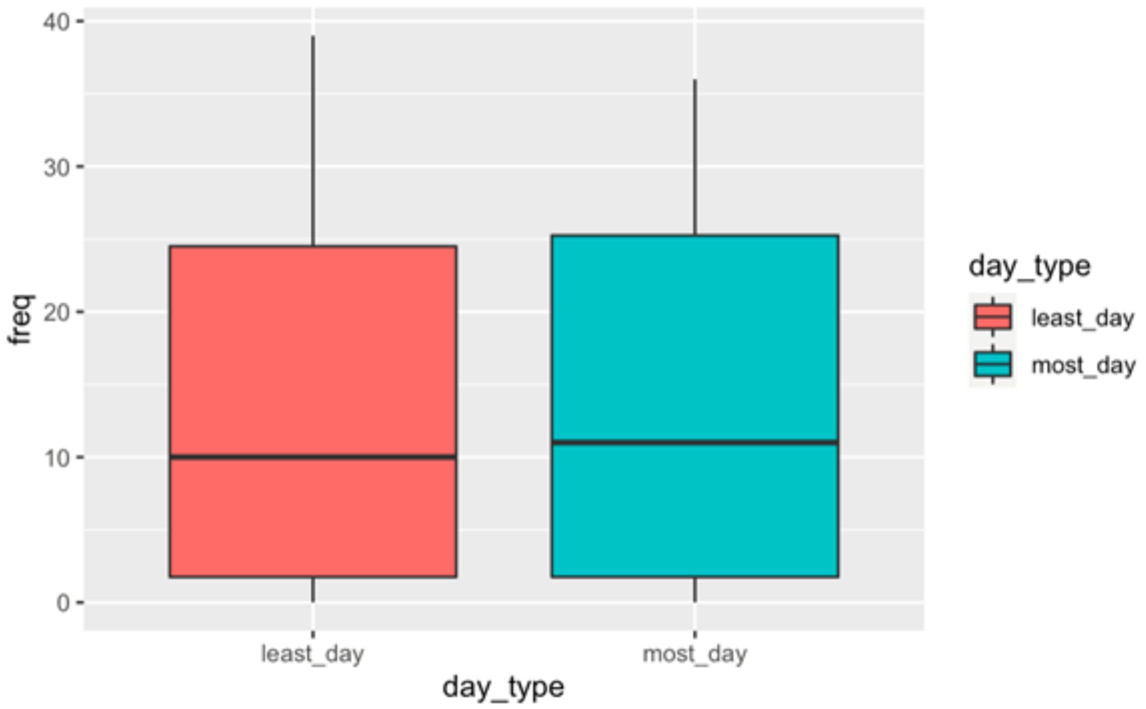
```
   1    2    3    4    5    6    7
 312  509  964 1130  781  407  312

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
  35   25    6    7    2    7   16   43   46   72  152  223  272  413  415  409  403  438  380
  19   20   21   22   23
 296  318  227  132   78
```

1.2 Data analysis

By observing the frequency statistics results above, we found that the number of users of this station every day within a week is quite different, on July 3 and 4, the station usage rate was significantly higher than other dates. In addition, we had observed that the number of users of this station was significantly higher from 10:00AM to 10:00PM, so we raised our first question in this project. That is, would the peak usage period of the station be related to whether it was a working day? Therefore, our null hypothesis was that the peak usage period of the station had nothing to do with whether it is a working day.

In order to better answer this question, we chose July 4 (Saturday) with the largest number of users and July 7 (Wednesday) with the least number of users to compare whether the peak hours of these two days are the same.

From above frequency statistics results, we could see that there was a big difference between the total number of times the station was used on Wednesday and Saturday. In order to better answer our question, we normalized the usage data of these two days, and then we used box plots to show the usage of the station on Wednesday and Saturday as below.

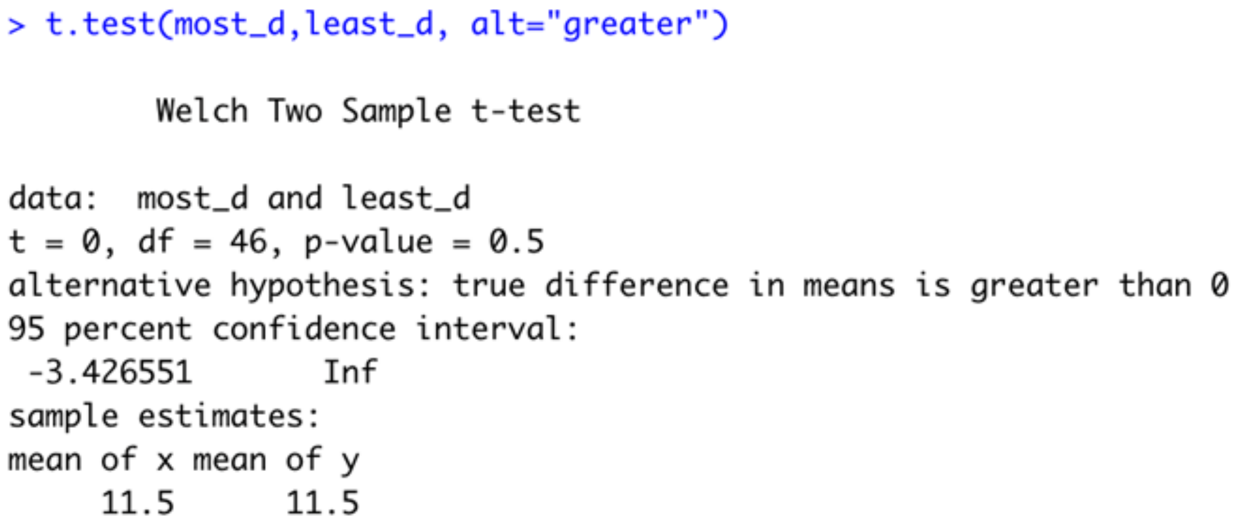


From the box plots, we could see that the usage of bikes on Wednesday and Saturday was basically the same. In order to continue to test the relationship between the peak period of bike use at the station and whether that day was a working day, we used the Chi Square Test for further exploration.

1.3 Result

Chi Square Test (Two-way tables)

H0: The peak time of station usage has nothing to do with whether the day is a working day, the peak usage time of the station does not change depending on the day of the week.



```
> t.test(most_d,least_d, alt="greater")

	Welch Two Sample t-test

data:  most_d and least_d
t = 0, df = 46, p-value = 0.5
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -3.426551       Inf
sample estimates:
mean of x mean of y 
     11.5      11.5 
```

From the result, we can see that p-value=.5 (>.05), thus at 5% significance level, we cannot reject the null hypothesis. In addition, we can see that the 95 percent confidence interval also contains 0. Therefore, we believe that the peak time of station usage has nothing to do with whether it is a working day or not.

2. Research Question 2: Can we predict the usage duration and riding distance with the probabilities obtained from large samples?

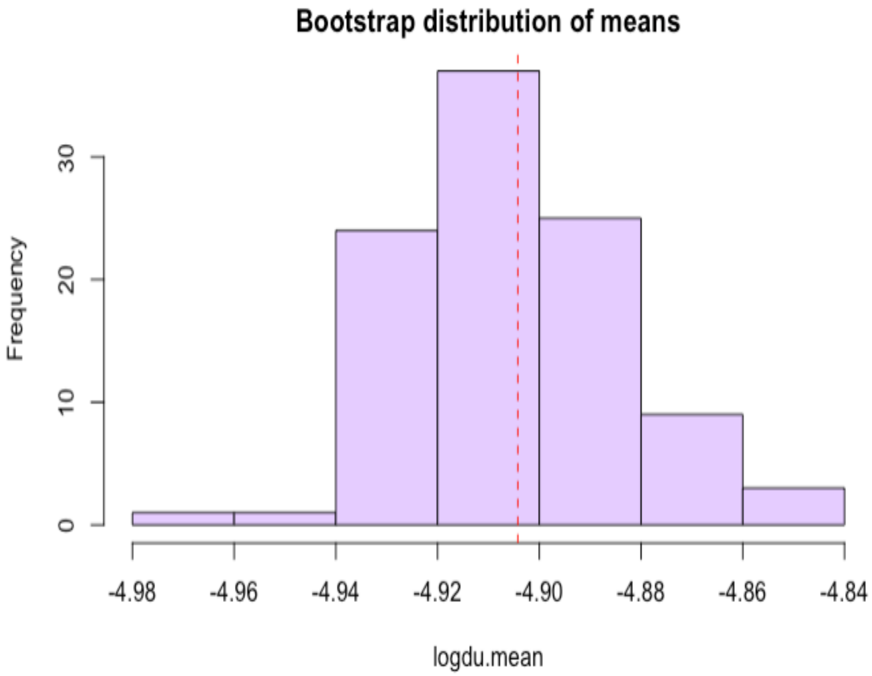
2.1 Data processing

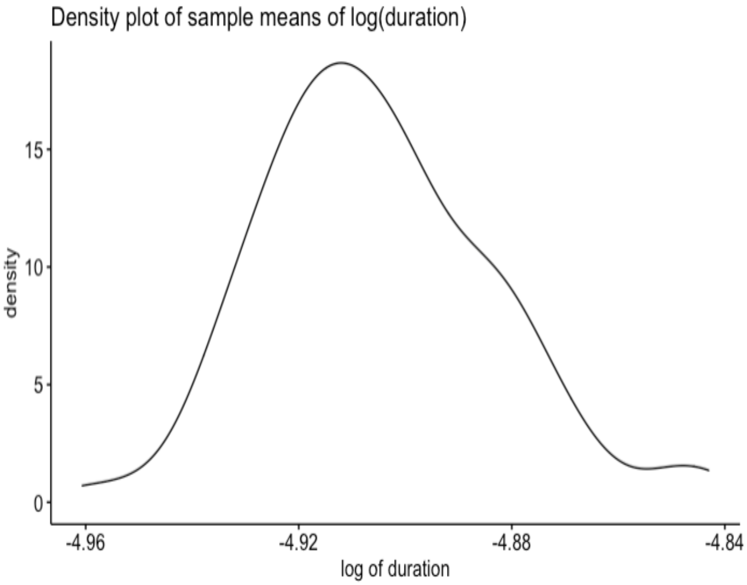
The character variables represented time and distance were converted to quantitative variables. Then the usage time between two stations was calculated and the new variable ‘duration’ was created. The library(geosphere) was used to calculate the riding distance based on latitude and longitude of start and end locations.

2.1.2 Data analysis (usage time)

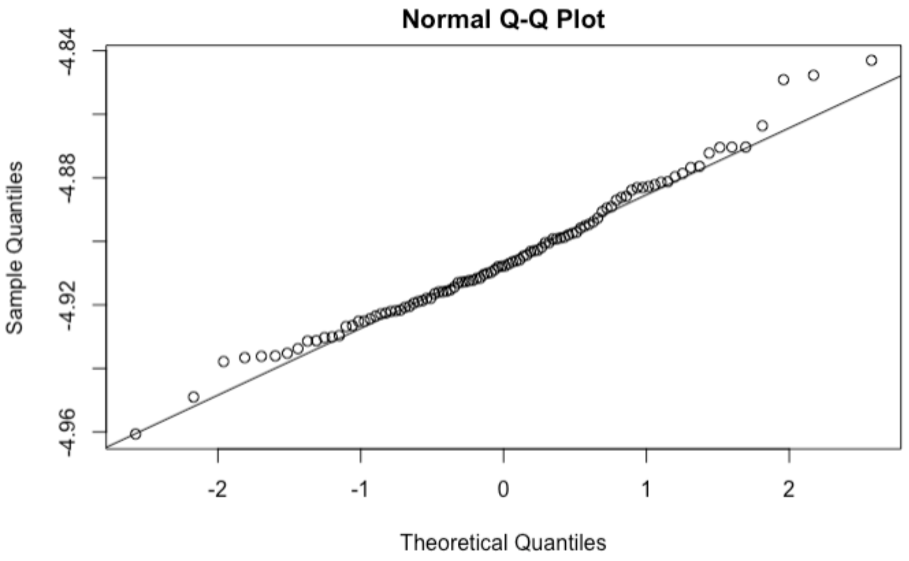
Log transformation was used to change the scale of the variable ‘duration’. We made several visualizations to show the bootstrap distributions of log(duration).

2.1.3 Results (usage time)





The graphs indicated that the distribution of sample mean of log(duration) was approximately bell-shaped. The mean value of the log(duration) was around -4.9, which meant that the time value was equal to 0.0074 and time duration was 10 minutes and 39 seconds.



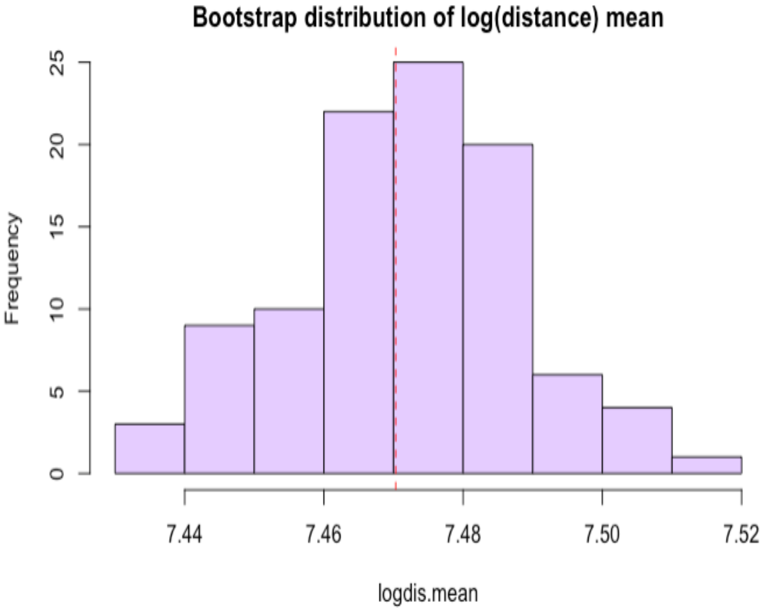
The nearly straight line in the qqplot graph showed that the distribution of sample mean of log(duration) was normally distributed with the points clustering on the 45-degree line.

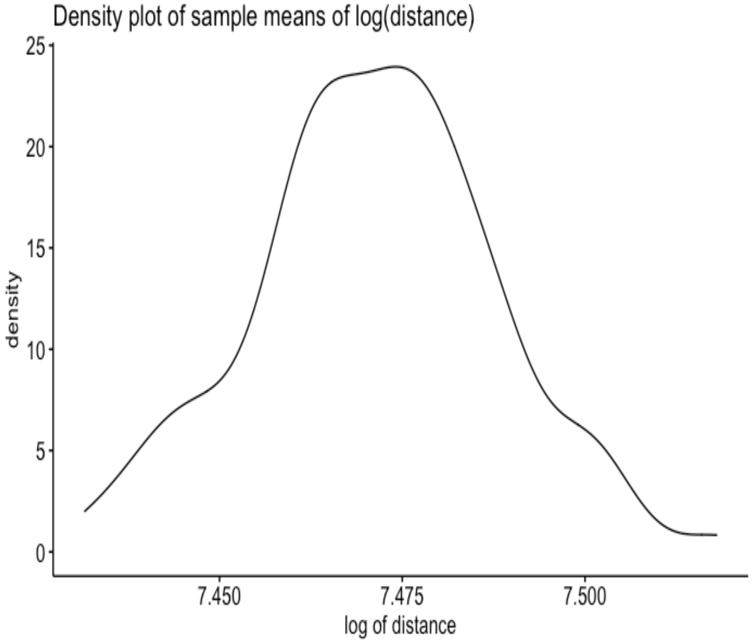
By applying the z-test, we calculated the z statistic and p-value of log(distance)'s sample mean, which were 1.124169 and 0.26. So, we failed to reject the null hypothesis and conclude that the sample mean of log(distance) is statistically equal to the population mean. Besides, the mean and bias of bootstrap mean are -4.9 and -0.01 respectively. Based on the sample mean and standard deviation, we calculate the 95% confidence interval, [-5, -4.8], which means that the riding time interval is (9:39, 14:24).

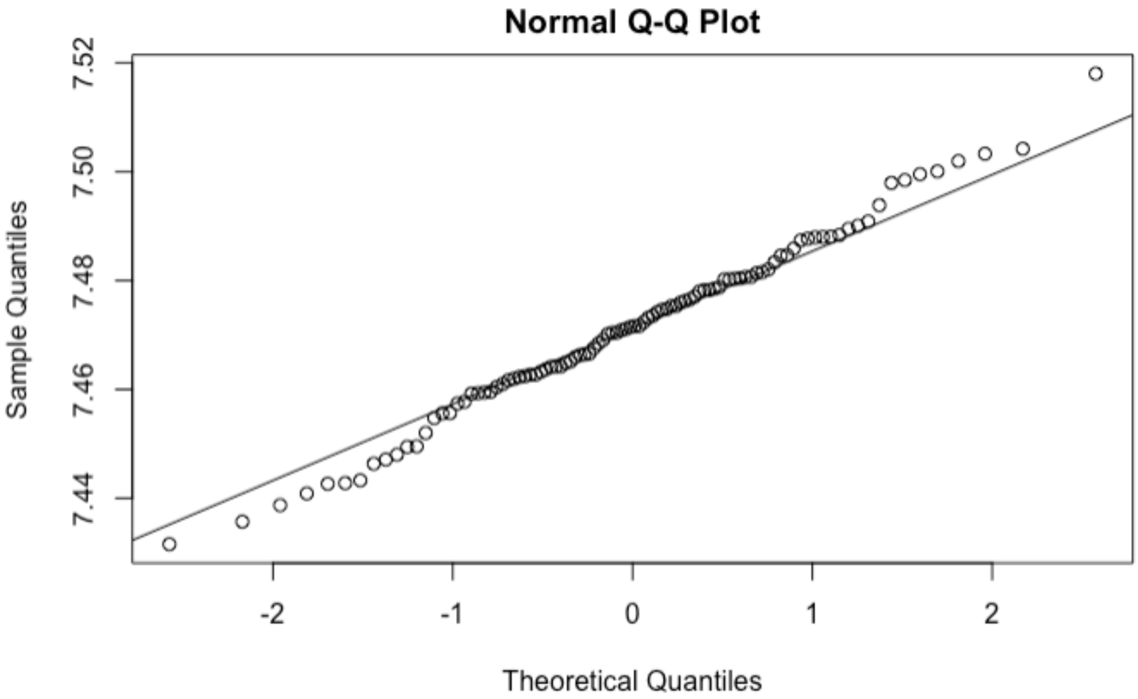
2.2.1 Distance analysis (distance)

Using log transformation to change the scale of the distance variable. We make several visualizations to show the distributions of it.

2.2.2 Results (distance)







The graphs indicated that the distribution of log(distance) was approximately bell-shaped. The mean value of the log(distance) was around 7.47, which meant that the riding distance was approximately 1.8 kilometers.

The almost straight line in the qqplot graph showed that the distribution of log(distance) was normally distributed with the points clustering on the 45-egree line.

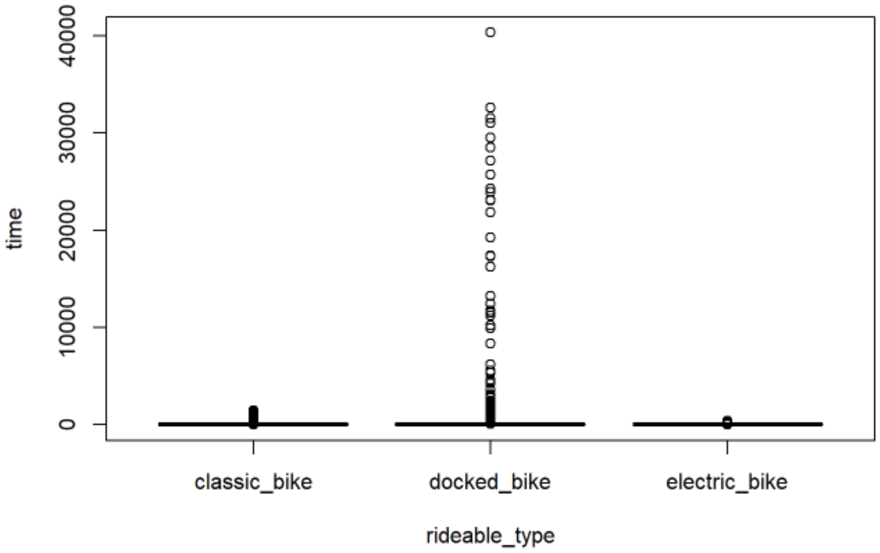
By applying the z-test, we calculated the z statistic and p-value of log(distance)'s sample mean, which were 0.1350135 and 0.8926012. So, we failed to reject the null hypothesis and got the conclusion that the sample mean of log(distance) was statistically equal to the population mean. Besides, the mean and bias of bootstrap mean were 7.47 and 0.011 respectively. Based on the sample mean and standard deviation, we calculated the 95% confidence interval, [7.38,7.5], which meant that the riding time interval was (1.6,1.8) kilometers.

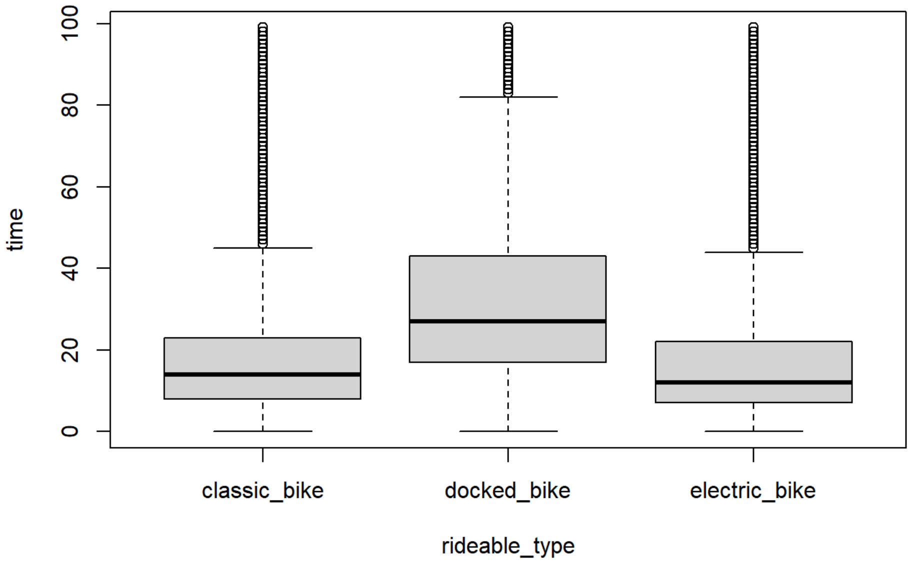
3. Research Question 3: Can we predict the rideable type through the time of using the bikes and the stations?

3.1 Data processing

The usage time of sharing bikes was calculated by subtracting the start time(time_start) from the end time(time_end) in the raw dataset.

Then we used the boxplot to see the overview distribution of different rideable types of these bikes with time.



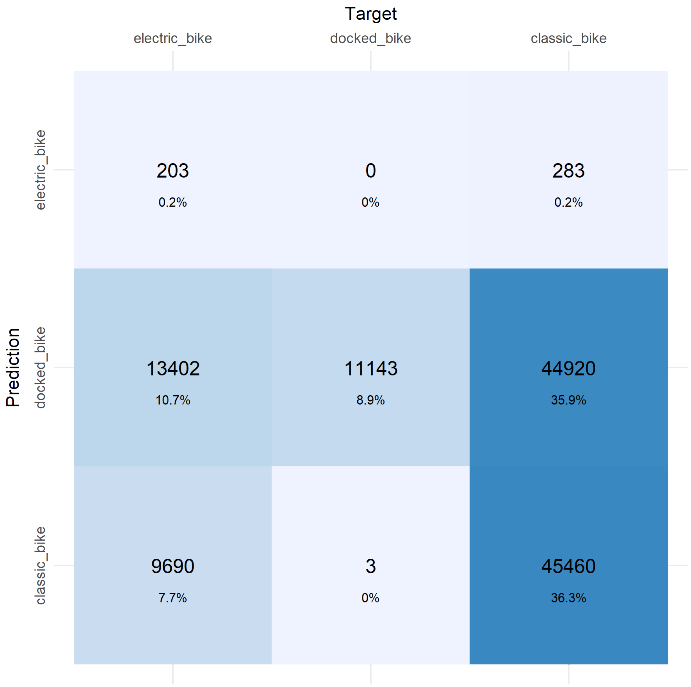


Through the first plot, we can see that the docked bikes had many outliers with impossible length of time. Therefore, we moved those outliers which were larger than 100 minutes and plotted again. And the second box plot showed the data after removing those outliers.

3.2 Data Analysis

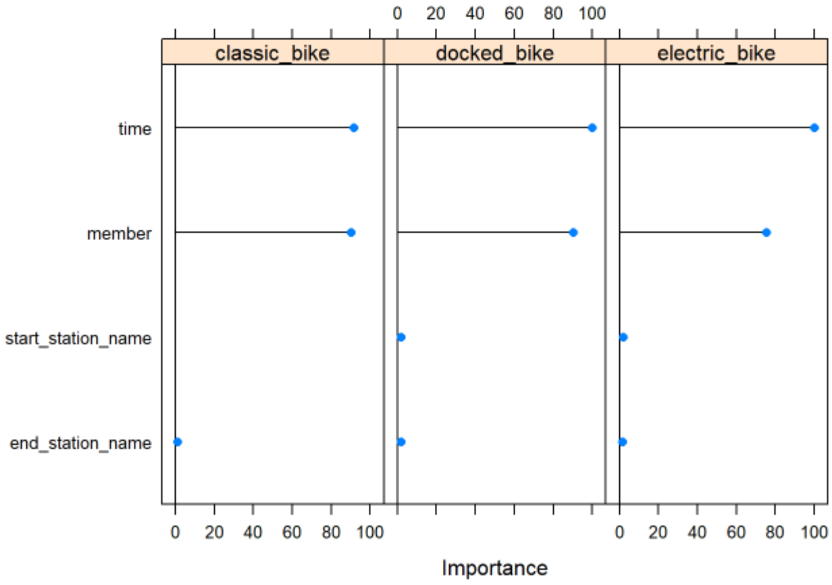
The problem was how to predict the rideable type through the time of using the bikes and the stations. To solve it, firstly, we separated the data to a train dataset with 80% data and a test dataset with the remaining 20%.

We built a naive bayes model to predict the rideable type. The confusion matrix was shown below:



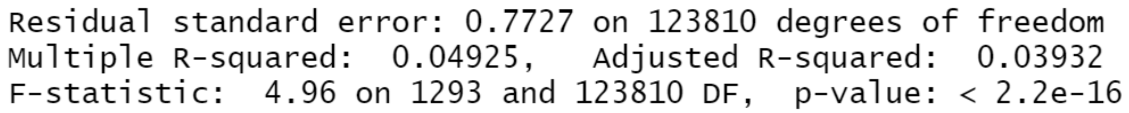
3.3 Results

Therefore, we calculated that the accuracy was just 51.29% and the Kappa value was only 0.1339. So, we could say that the Naive Bayes model did not perform well. There might be other factors that do not include.



According to the variable importance plot, we could see that the usage time of bikes and the membership were important factors in the prediction model. By contrast, the start and end station locations were not important, which was not the same as what we previously assumed.

Then we tried to use a linear regression model to predict it again and compare the two models. For the linear model, the overview statistics were shown below:



```
Residual standard error: 0.7727 on 123810 degrees of freedom
Multiple R-squared:  0.04925,   Adjusted R-squared:  0.03932
F-statistic:  4.96 on 1293 and 123810 DF,  p-value: < 2.2e-16
```

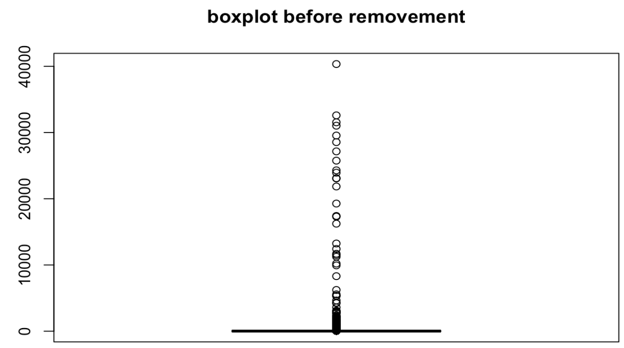
We could conclude that the linear model also did not perform well because the residual standard error was too high and the adjusted R-squared was too low. It might be because there still existed some error data or the model might be overfitting due to too many variables.

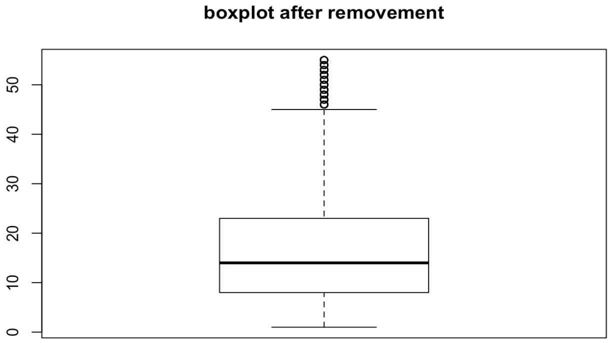
4. Research Question 4: Does the membership affect the usage of time?

4.1 Data processing

The original data set only had the time_start and time_end as both sides of the time stamps. In question 4, we needed to have a variable showing how long the riding lasted for. The data processing in this question was quite similar to what we did in question 3.

Also, we checked the boxplot of the time used so that we could have an overview on the distribution of the data. The boxplot looks as left one:





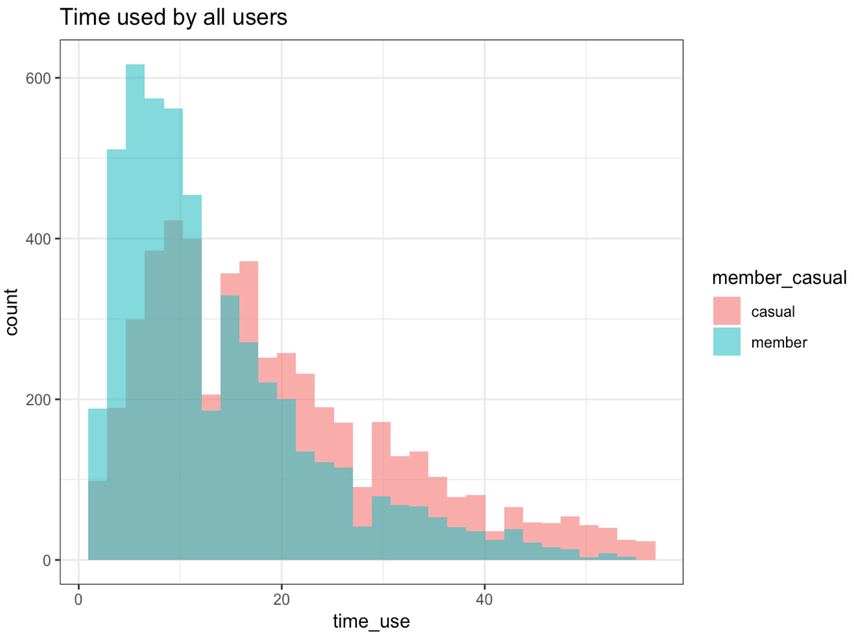
The box plot indicated that there were many outliers with an impossible time record. Some of them even reached 40000 minutes. These outcomes could owe to malfunction of the timer, misuse or system crushing. We removed the outliers and the second box plot showed the distribution after removing the incorrect data.

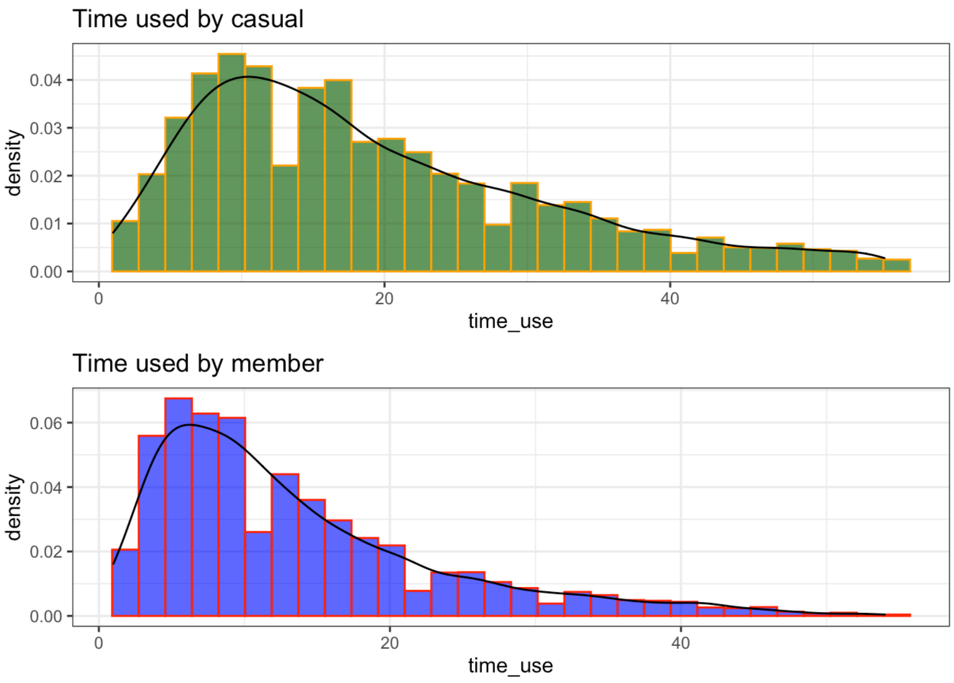
4.2 Data Analysis

The problem here was whether the members spend more time for each ride than casual users or not. Since the time spent was a continuous variable, t-test became a good choice to do the analysis. Before performing the t-test, we set up the hypothesis first according to our common sense.

The null hypothesis was that members tend to spend less or the same time on riding as the casual users. The alternative hypothesis was that members tend to spend more time on riding than the casual users.

In order to make that analysis, we needed to take samples to reduce the size of the data set. We chose 5000 member sampling and 5000 casuals sampling. We also created two visualizations for the sample, showing the distribution of the sampled data.


Time used by all users
count
600
400
200
0
0
20
40
time_use
member_casual
🟥 casual
🟦 member


Time used by casual
density
0.04
0.03
0.02
0.01
0.00
0
20
40
time_use
Time used by member
density
0.06
0.04
0.02
0.00
0
20
40
time_use

4.3 Results

We did the two samples t-tests for the data set that we sampled. The results were shown below:

<table>
<tr><td colspan="3"><b>Welch Two Sample t-test</b></td></tr>
<tr><td colspan="3"><b>data: member_rider$time_use and casual_rider$time_use</b></td></tr>
<tr><td><b>t = -25.648</b></td><td><b>df = 9574.2</b></td><td><b>p-value = 1</b></td></tr>
<tr><td colspan="3"><b>95 percent confidence interval:</b></td></tr>
<tr><td><b>-6.093892</b></td><td colspan="2"><b>Inf</b></td></tr>
</table>

The p-value was 1 which meant we should accept the null hypothesis. Members tend to spend less or equal time with the casuals. This result was opposed to our initial assumption since the common sense for people was that they would have the membership only if they spent more time on shared bikes than normal.

In order to verify whether the casuals spent more time on the shared bikes or not, we did another t-test. The null hypothesis was that the members tend to spend greater or equal time than the casuals. The alternative hypothesis is that the members tend to spend less time than the casuals. The results were shown below:

<table>
<tr><td colspan="3"><b>Welch Two Sample t-test</b></td></tr>
<tr><td colspan="3"><b>data: member_rider$time_use and casual_rider$time_use</b></td></tr>
<tr><td><b>t = -25.648</b></td><td><b>df = 9574.2</b></td><td><b>p-value = 2.2e-16</b></td></tr>
<tr><td colspan="3"><b>95 percent confidence interval:</b></td></tr>
<tr><td><b>-Inf</b></td><td colspan="2"><b>-5.359308</b></td></tr>
</table>

The p-value was approximately 0, which meant that we should reject the null hypothesis and get the conclusion that the time members spent on each riding was less than the casuals.

Conclusion and Discussion

Chicago’s bike-sharing program is one of the most mature bike-sharing systems in the world. It receives grants from local transportation authorities, municipal governments, or private companies. As a solid alternative to a car, bike sharing has many advantages. It is cheaper than transit or car rental. It eliminates the need for personal bike ownership. It is healthier than driving or riding transit.

However, after doing some research we found that there still exist some disadvantages needed to improve. For some rideable types of bikes, their frames are bulkier than typical road bikes. And some rideable types of bikes might be too popular to satisfy all users’ demand. As a novice bike-share user, you might face acclimation period during which you are less satisfied than you are on other bikes. Dock shortages can also be a problem. The likelihood of being shut out of a station grows when you are traveling to the same destination as everyone else. Many users, both members and casuals are complaining about charging of the bikes. Bike sharing is not a great choice for a long, leisurely ride along country roads or rural trails. Although it is hard to eliminate all these problems because bike sharing does not always make sense for everyone, we can improve bike-sharing management through our data analysis.

To help users avoid dock shortage as possible, it is worthwhile to find out the peak period of station usage and make corresponding measures. According to our common sense, the peak period might depend on working days and weekends. Based on the preview of frequency count of the busiest station, the busiest day is Thursday and the most vacant the is Sunday. However, the Chi Square Test result shows that the peak time of station usage has nothing to do what the day is of the week. Through frequency analysis, we also found that the peak time of a day is from 10:00 AM to 10:00 PM. So, we can arrange shared bikes staff and replenish the bikes needed at the station during this period every day.

Our common sense just tells us that bike sharing is not a good choice for a long-distance travel, but how to define ‘long’? It also depends on many other complicate matters like geography and climate. This analysis results can help users to make right decision when they hesitate choosing bike sharing.

There is no doubt that in the bike-sharing system, the most important physical element is the bike. When choosing the bike type, users need to consider their plan to ride.

Bikes of different rideable types usually cost different, and those with higher-performance components or frame materials are expected to cost more than other bikes. Bikes also come in a range of sizes. Although most bikes' height can be adjusted within a range, fine-tune fit issues still arise from time to time. Predicting the rideable type is therefore a necessary approach knowing users' demand. In our project, we have tried to use Naïve Bayes model and linear regression to predict the rideable type through the tike of using the bikes and the stations, but these two models did not perform well. The reason could be that too many variables in the data set made the models overfitting.

The difference of charging criteria of members and casuals result in different user behaviors. By investigating how membership affect the usage of time, we got the result deviating from our common sense. One of the reasons might be that divvy uses a different charging mechanism. The casuals would be charged within limited times of using in a certain period while members did not have any limitation. Because of that, the casuals were inclined to spend more time on each use of the shared bikes while members would use whenever they needed. That made the situations that we found from the t-test analysis.

Reference

Dau, D. (2021). Kaggle case study data. Retrieved from：
https://www.kaggle.com/daudewey/case-study-1-bikeshare

511 final project

Zidong Xu

11/18/2021

```
# Problem: Will the peak period of station usage be related to whether it is a workin
g day?

data <- read.csv("511 data.csv")
head(data)
```

```
##            ride_id rideable_type  started_at    ended_at
## 1 B68C8C61487B5C15  classic_bike 7/1/21 0:00 7/1/21 0:46
## 2 3FBF4B489DE3F24E   docked_bike 7/1/21 0:00 7/1/21 0:10
## 3 8A1D30385234151D  classic_bike 7/1/21 0:00 7/1/21 0:28
## 4 D5AA171FAE3335A7  classic_bike 7/1/21 0:00 7/1/21 0:16
## 5 D493F6B96A3117A2  classic_bike 7/1/21 0:01 7/1/21 0:14
## 6 DF64E952788CAA2C electric_bike 7/1/21 0:02 7/1/21 0:09
##          start_station_name start_station_id            end_station_name
## 1 Bissell St & Armitage Ave            13059 Lincoln Ave & Diversey Pkwy
## 2   Clark St & Winnemac Ave     TA1309000035    Broadway & Thorndale Ave
## 3    Michigan Ave & Lake St     TA1305000011      Wabash Ave & Wacker Pl
## 4    Wabash Ave & Wacker Pl     TA1307000131       Wells St & Concord Ln
## 5      Halsted St & 18th St            13099         Morgan St & 31st St
## 6    Clark St & Elmdale Ave     KA1504000148    Broadway & Granville Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   TA1307000064  41.91844 -87.65222 41.93223 -87.65862        casual
## 2          15575  41.97335 -87.66786 41.98974 -87.66014        casual
## 3   TA1307000131  41.88602 -87.62412 41.88688 -87.62603        casual
## 4   TA1308000050  41.88688 -87.62603 41.91213 -87.63466        casual
## 5   TA1308000046  41.85750 -87.64628 41.83780 -87.65114        casual
## 6          15571  41.99085 -87.66965 41.99483 -87.66021        member
```

```
str(data)
```

```
## 'data.frame':    161010 obs. of  13 variables:
##  $ ride_id           : chr  "B68C8C61487B5C15" "3FBF4B489DE3F24E" "8A1D30385234151
D" "D5AA171FAE3335A7" ...
##  $ rideable_type     : chr  "classic_bike" "docked_bike" "classic_bike" "classic_b
ike" ...
##  $ started_at        : chr  "7/1/21 0:00" "7/1/21 0:00" "7/1/21 0:00" "7/1/21 0:00
" ...
##  $ ended_at          : chr  "7/1/21 0:46" "7/1/21 0:10" "7/1/21 0:28" "7/1/21 0:16
" ...
##  $ start_station_name: chr  "Bissell St & Armitage Ave" "Clark St & Winnemac Ave"
"Michigan Ave & Lake St" "Wabash Ave & Wacker Pl" ...
##  $ start_station_id  : chr  "13059" "TA1309000035" "TA1305000011" "TA1307000131" .
..
##  $ end_station_name  : chr  "Lincoln Ave & Diversey Pkwy" "Broadway & Thorndale Av
e" "Wabash Ave & Wacker Pl" "Wells St & Concord Ln" ...
##  $ end_station_id    : chr  "TA1307000064" "15575" "TA1307000131" "TA1308000050" .
..
##  $ start_lat         : num  41.9 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
summary(data)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:161010      Length:161010      Length:161010      Length:161010     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:161010      Length:161010      Length:161010      Length:161010     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##    start_lat       start_lng         end_lat         end_lng      
##  Min.   :41.65   Min.   :-87.77   Min.   :41.65   Min.   :-87.77  
##  1st Qu.:41.88   1st Qu.:-87.65   1st Qu.:41.88   1st Qu.:-87.65  
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64  
##  Mean   :41.90   Mean   :-87.64   Mean   :41.91   Mean   :-87.64  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :42.06   Max.   :-87.53   Max.   :42.06   Max.   :-87.53  
##  member_casual     
##  Length:161010     
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

```
# Count the number of times each station is used, and sort them in decrease order.
count_start_station_name <- as.data.frame(table(data$start_station_name))
colnames(count_start_station_name) <- c("start_station_name","counts")
count_start_station_name <- count_start_station_name[order(count_start_station_name$c
ounts,decreasing = TRUE),]
head(count_start_station_name)
```

```
##                start_station_name counts
## 580    Streeter Dr & Grand Ave   4415
## 349 Lake Shore Dr & North Blvd   2703
## 348  Lake Shore Dr & Monroe St   2215
## 418      Michigan Ave & Oak St   2126
## 583        Theater on the Lake   2000
## 423            Millennium Park   1926
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

# Select the most popular station for further study.
most_freq_start_station_data <- data[data$start_station_name=="Streeter Dr & Grand Ave",c("start_station_name","started_at")]

# Convert time data to POSIXlt format, and add date, hour, and minute variables to facilitate subsequent frequency statistics and other analysis.
most_freq_start_station_data$started_at <-
  strptime(most_freq_start_station_data$started_at,"%m/%d/%y %H:%M")
most_freq_start_station_data$day <-
  day(most_freq_start_station_data$started_at)
most_freq_start_station_data$hour <-
  hour(most_freq_start_station_data$started_at)
most_freq_start_station_data$minute <-
  minute(most_freq_start_station_data$started_at)
str(most_freq_start_station_data)
```

```
## 'data.frame':    4415 obs. of  5 variables:
##  $ start_station_name: chr  "Streeter Dr & Grand Ave" "Streeter Dr & Grand Ave" "S
treeter Dr & Grand Ave" "Streeter Dr & Grand Ave" ...
##  $ started_at        : POSIXlt, format: "2021-07-01 00:02:00" "2021-07-01 00:04:00
" ...
##  $ day               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hour              : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ minute            : int  2 4 6 8 11 12 47 47 2 39 ...
```

```
# July 1-7, the result of the frequency statistics of "Streeter Dr & Grand Ave" stati
on in units of days, hours, and minutes.
(table(most_freq_start_station_data$day))
```

```
##
##    1    2    3    4    5    6    7
##  312  509  964 1130  781  407  312
```

```
(table(most_freq_start_station_data$hour))
```

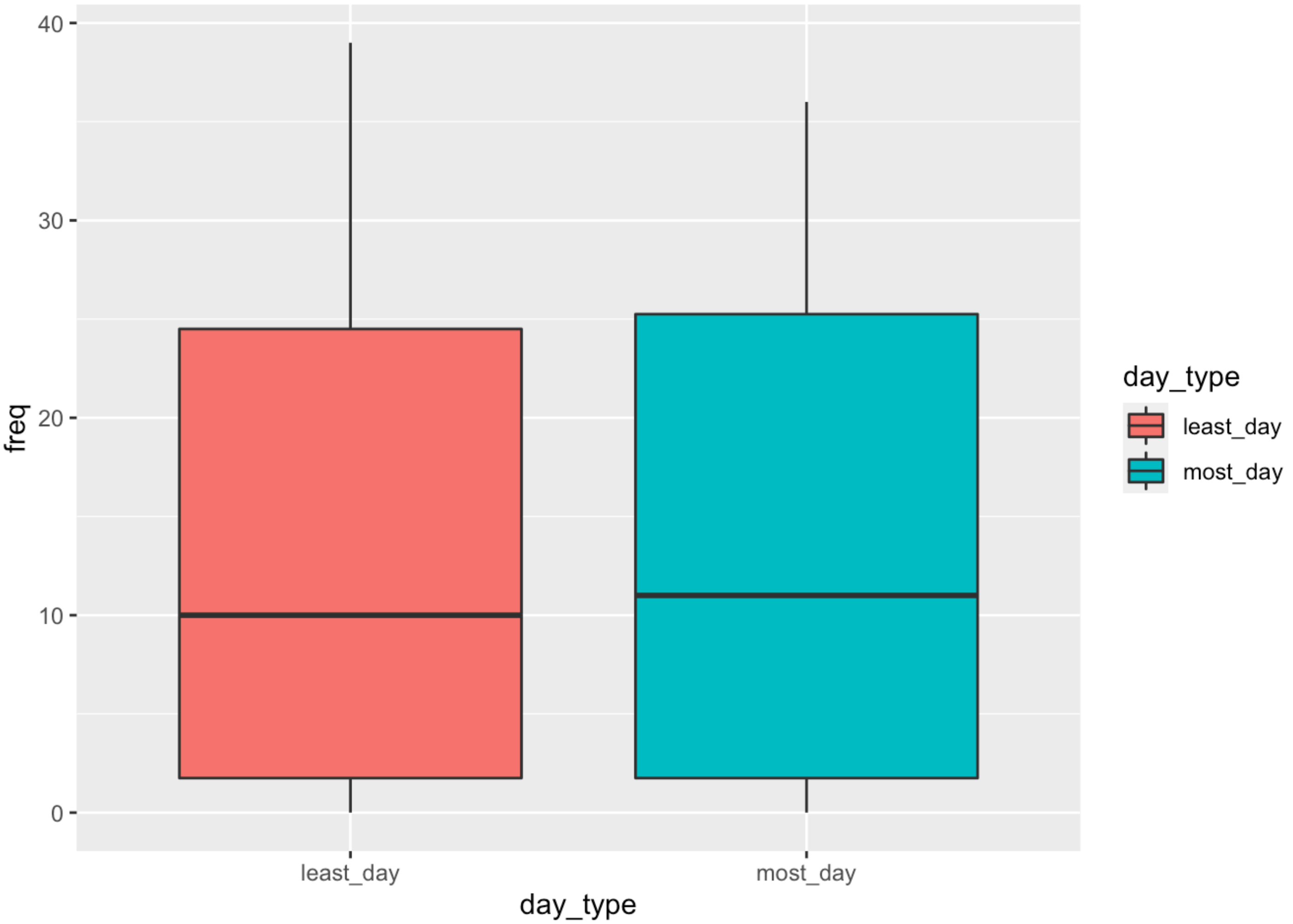
```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  35  25   6   7   2   7  16  43  46  72 152 223 272 413 415 409 403 438 380 296
##  20  21  22  23
## 318 227 132  78
```

```
(table(most_freq_start_station_data$minute))
```

```
##
##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## 89 74 71 76 81 67 91 91 72 78 82 81 63 64 66 62 71 54 70 65 79 77 75 70 85 70
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## 66 87 76 84 62 67 70 62 74 76 64 66 64 93 76 59 82 76 76 95 87 81 63 68 88 80
## 52 53 54 55 56 57 58 59
## 66 81 67 62 69 60 74 70
```

```
# By observing the frequency statistics results above, we found that the number of us
ers of this site every day within a week is quite different. In addition, we have obs
erved that the number of users of this site is significantly higher from 10:00AM to 1
0:00PM, so we raise the question in this section. That is, will the peak usage period
 of the station be related to whether it is a working day?
```

```
# In order to better answer this question, we choose July 4 (Saturday) with the largest number of users and July 7 (Wednesday) with the least number of users to compare whether the peak hours of the two days are the same.

# H0: The peak usage period of the station has nothing to do with whether it is a working day.

day_most<-most_freq_start_station_data[most_freq_start_station_data$day==4,]
day_least<-most_freq_start_station_data[most_freq_start_station_data$day==7,]

b_hour_most <- as.data.frame(table(day_most$hour))
colnames(b_hour_most) <- c("Hour","most_day_freq")

b_hour_least <-as.data.frame(table(day_least$hour))
colnames(b_hour_least) <- c("Hour","least_day_freq")

# Normalization
# In the above frequency statistics, we find that there is a big difference between the number of times the station is used on Wednesday and Saturday. In order to better study the relationship between the peak period of site use and whether it is a working day, we normalize the usage data of these two days.

mm <- mean(b_hour_most$most_day_freq)
lm <- mean(b_hour_least$least_day_freq)
b_hour_most$most_day_freq <- b_hour_most$most_day_freq/(mm/lm)
b_hour_most$most_day_freq <- round(b_hour_most$most_day_freq,0)

hour <- data.frame(c(0:23))
colnames(hour) <- "Hour"

compare1.1 <- merge(hour,b_hour_most,by="Hour",all.x = TRUE)
compare1.1[is.na(compare1.1)] <- 0
compare1.2 <- merge(hour,b_hour_least,by="Hour",all.x = TRUE)
compare1.2[is.na(compare1.2)] <- 0
compare1.1$most_day <- "most_day"
colnames(compare1.1) <- c("Hour","freq","day_type")
compare1.2$least_day <- "least_day"
colnames(compare1.2) <- c("Hour","freq","day_type")
compare1 <- rbind(compare1.1,compare1.2)

# Use box plots to show the usage of the station on Wednesday and Saturday.
compare1 %>%
ggplot(aes(x = day_type, y = freq, fill=day_type)) +
geom_boxplot()
```



```
most_d <- subset(compare1, select = Hour, subset = day_type=="most_day")
least_d <- subset(compare1, select = Hour, subset = day_type=="least_day")

# Chi Square Test (Two-way tables)
# H0: The peak time of station usage has nothing to do with whether the day is a work
ing day, the peak usage time of the site does not change depending on the day of the w
eek.

t.test(most_d,least_d, alt="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  most_d and least_d
## t = 0, df = 46, p-value = 0.5
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -3.426551       Inf
## sample estimates:
## mean of x mean of y
##      11.5      11.5
```

```
# We can see the result below, which shows that p-value=.5 (>.05), thus at 5% signifi
cance level, we can not reject the null hypothesis. In addition, we can see that the
95 percent confidence interval also contains 0. Therefore, we believe that the peak t
ime of station usage has nothing to do with whether it is a working day or not.

# Our results show that no matter what day of the week, we can arrange for personnel
to replenish the bike needed on station at the same time period.

# Therefore, the results of this time can effectively help the bike-sharing staff to
regularly check whether there are enough bikes on the station.
```

Data preparation

```
data = read.csv("/Users/caofangping/511_update.csv")
data = data.frame(data)
data = subset(data,select = c(X511.data,X,X.1,X.2,X.3,X.4,X.5,X.6))
data <- data[-c(1),]
colnames(data)<- c("rideable","start_at","end_at","start_lat","start_lng","end_lat","
end_lng","member_casual")
#str(data)
data$rideable = as.factor(data$rideable)
data$member_casual = as.factor(data$member_casual)
data$start_at = as.numeric(data$start_at)
data$end_at = as.numeric(data$end_at)
```

```
## Warning: 强制改变过程中产生了NA
```

```
data$start_lat = as.numeric(data$start_lat)
data$start_lng = as.numeric(data$start_lng)
data$end_lat = as.numeric(data$end_lat)
data$end_lng = as.numeric(data$end_lng)
#str(data)
data = na.omit(data)
```

#compute variables duration and distance

```
duration = data$end_at - data$start_at
data = data.frame(data,duration)
```

```
library(geosphere)
library(dplyr)
```

```
##
## 载入程辑包：'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data = data %>% rowwise() %>%
  mutate(distance = distHaversine(c(start_lng, start_lat), c(end_lng, end_lat)))
data$distance = as.numeric(data$distance)
data = data.frame(data)
#str(data)
#head(data)
```

```
#transformation: log of duration
duration_log = log(data$duration)
```

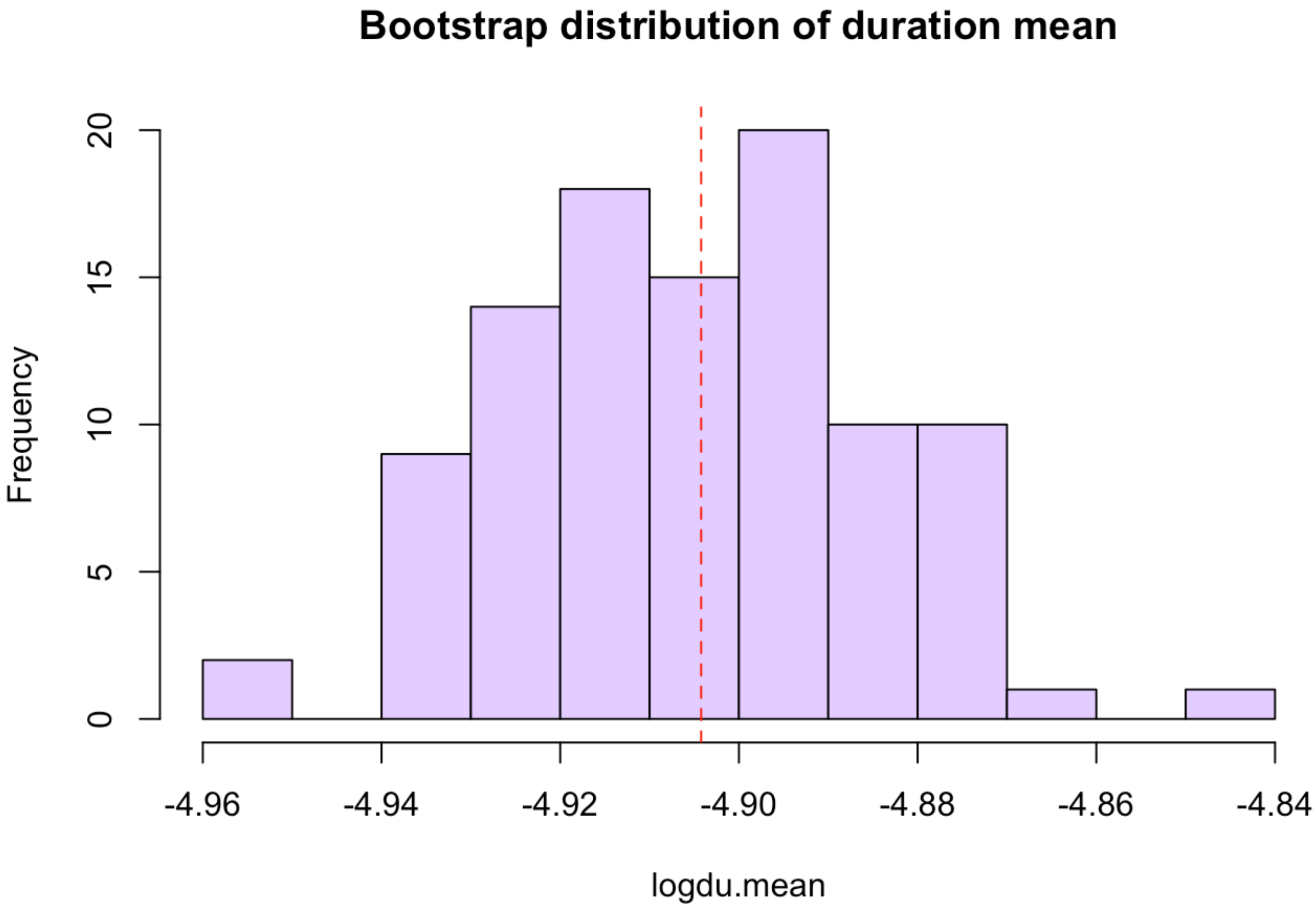
```
## Warning in log(data$duration): 产生了NaNs
```

```
data = data.frame(data,duration_log)
```

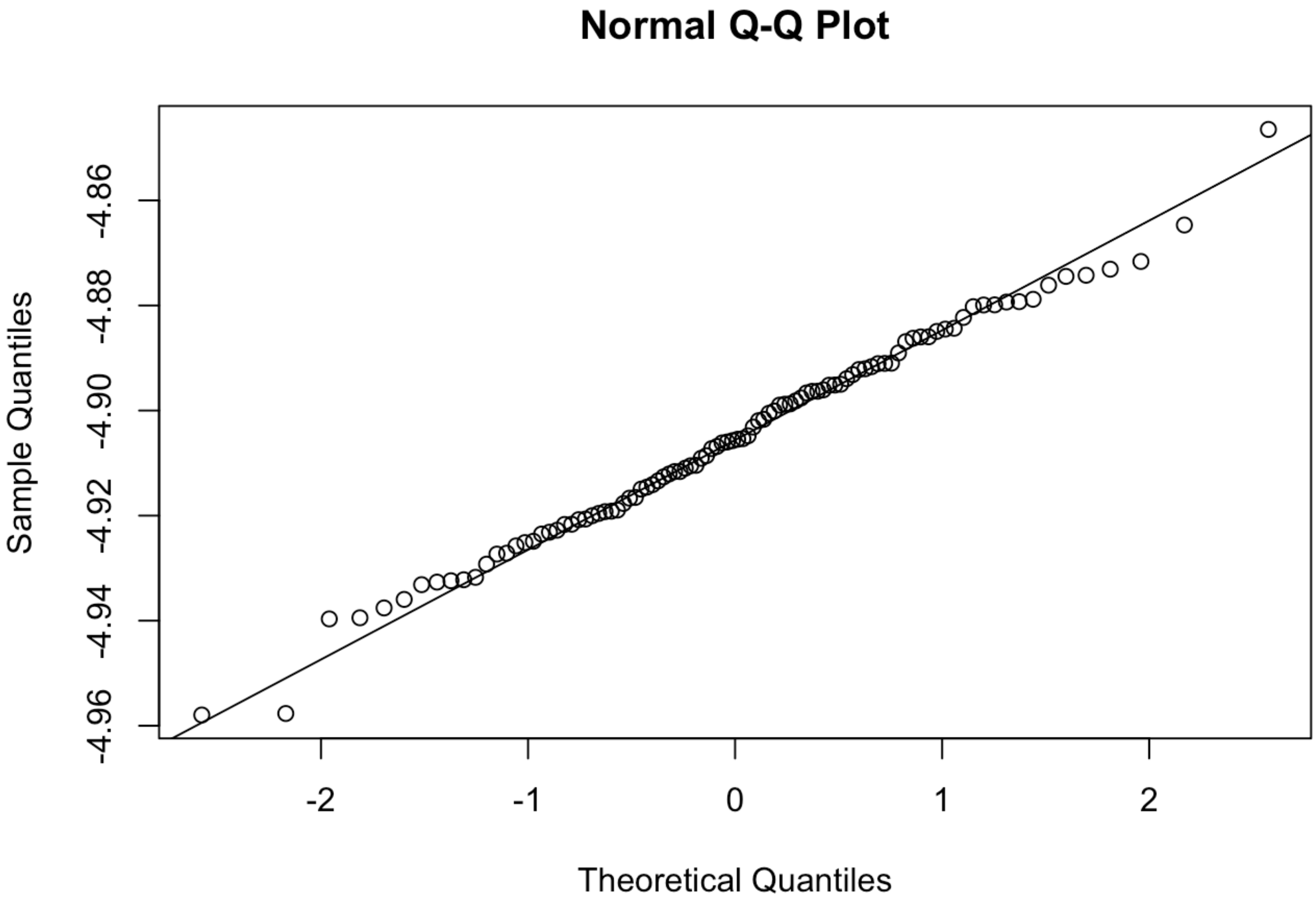
```
#transformation: log of distance
distance_log = log(data$distance)
data = data.frame(data,distance_log)
write.csv(data,"511_update1.csv")
```

Duration Analysis

```
#remove outliers and plots
data1<-data[!(data$duration_log <=-10 | data$duration_log >=0),]
data1 = na.omit(data1)
n <- length(data$duration_log)
N <- 100
logdu.mean <- numeric(N)
for (i in 1:N)
{
  x <- sample(data1$duration_log, n, replace = TRUE)
  logdu.mean[i] <- mean(x) #bootstrap sample mean
}
hist(logdu.mean, main = "Bootstrap distribution of duration mean",col = '#E5CCFF')
abline(v = mean(data1$duration_log), col = "red", lty = 2)
```



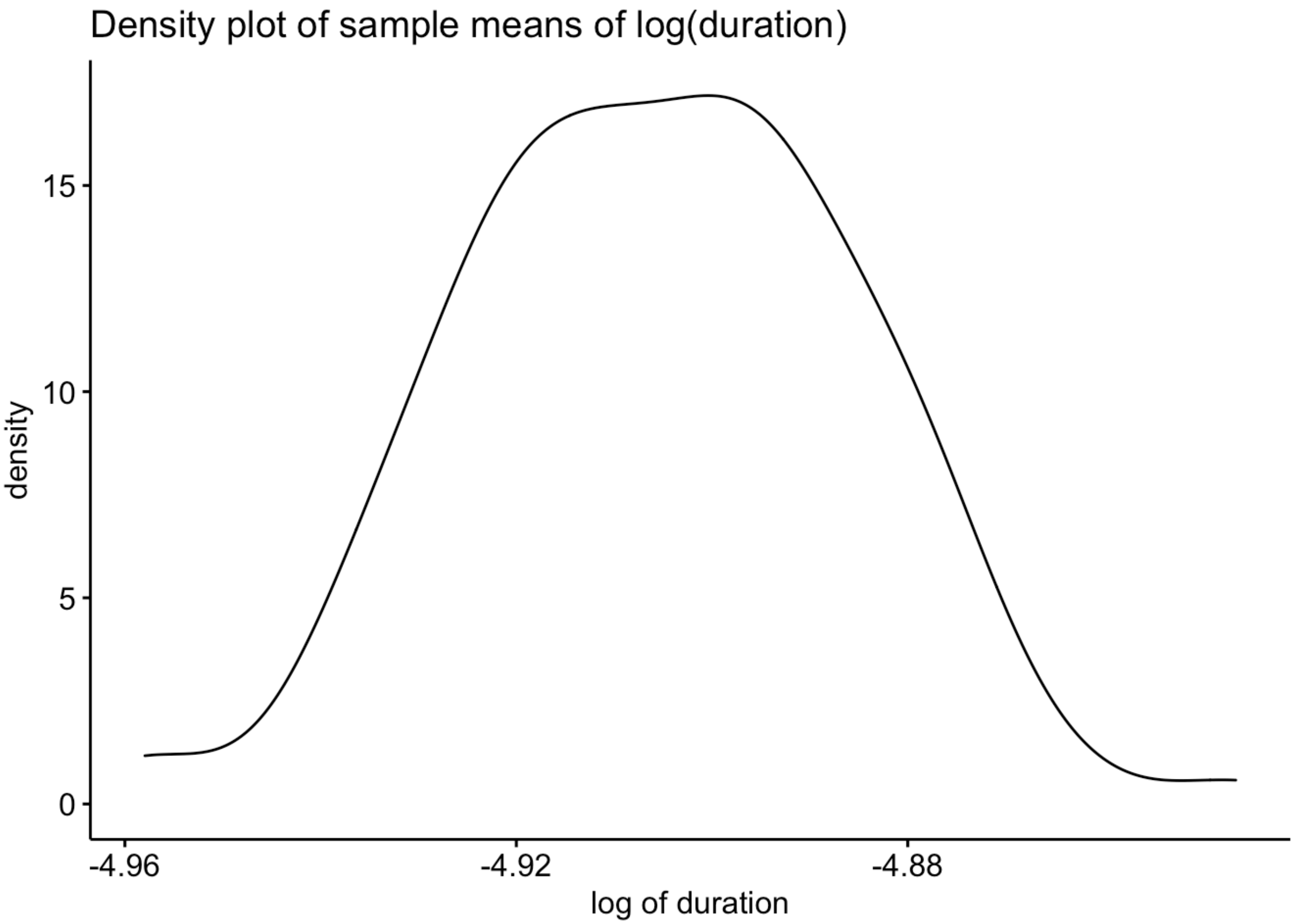
```
qqnorm(logdu.mean)
qqline(logdu.mean)
```



```
library("ggpubr")
```

```
## 载入需要的程辑包：ggplot2
```

```
ggdensity(logdu.mean,
          main = "Density plot of sample means of log(duration)",
          xlab = "log of duration")
```


Density plot of sample means of log(duration)

```
#mean and sd
boot.dumean= mean(logdu.mean)
boot.dusd=sd(logdu.mean)
bias.du = boot.dumean - mean(data1$duration_log)
boot.dumean
```

```
## [1] -4.905289
```

```
bias.du
```

```
## [1] -0.001063335
```

```
#95% confidence interval
LL.du=boot.dumean-1.96*boot.dusd
UL.du=boot.dumean+1.96*boot.dusd
LL.du
```

```
## [1] -4.944768
```

```
UL.du
```

```
## [1] -4.86581
```

```
#z test
library(distributions3)
```

```
##
## 载入程辑包：'distributions3'
```

```
## The following objects are masked from 'package:stats':
##
##     Gamma, quantile
```

```
## The following object is masked from 'package:grDevices':
##
##     pdf
```

```
mean_dur = mean(data1$duration_log)
sd_dur = sd(data1$duration_log)
Size1 <- as.integer(nrow(data1)/8)
sample1 <- sample(nrow(data1), Size1, replace = FALSE)
sample_dur<-data1[sample1, ]
#Calculate z-statistics and two-sided p-value
n1 <- length(sample_dur$duration_log)
z_stat1 <- (mean(sample_dur$duration_log) - mean_dur) / (sd_dur / sqrt(n1))
z_stat1
```

```
## [1] 0.07157019
```

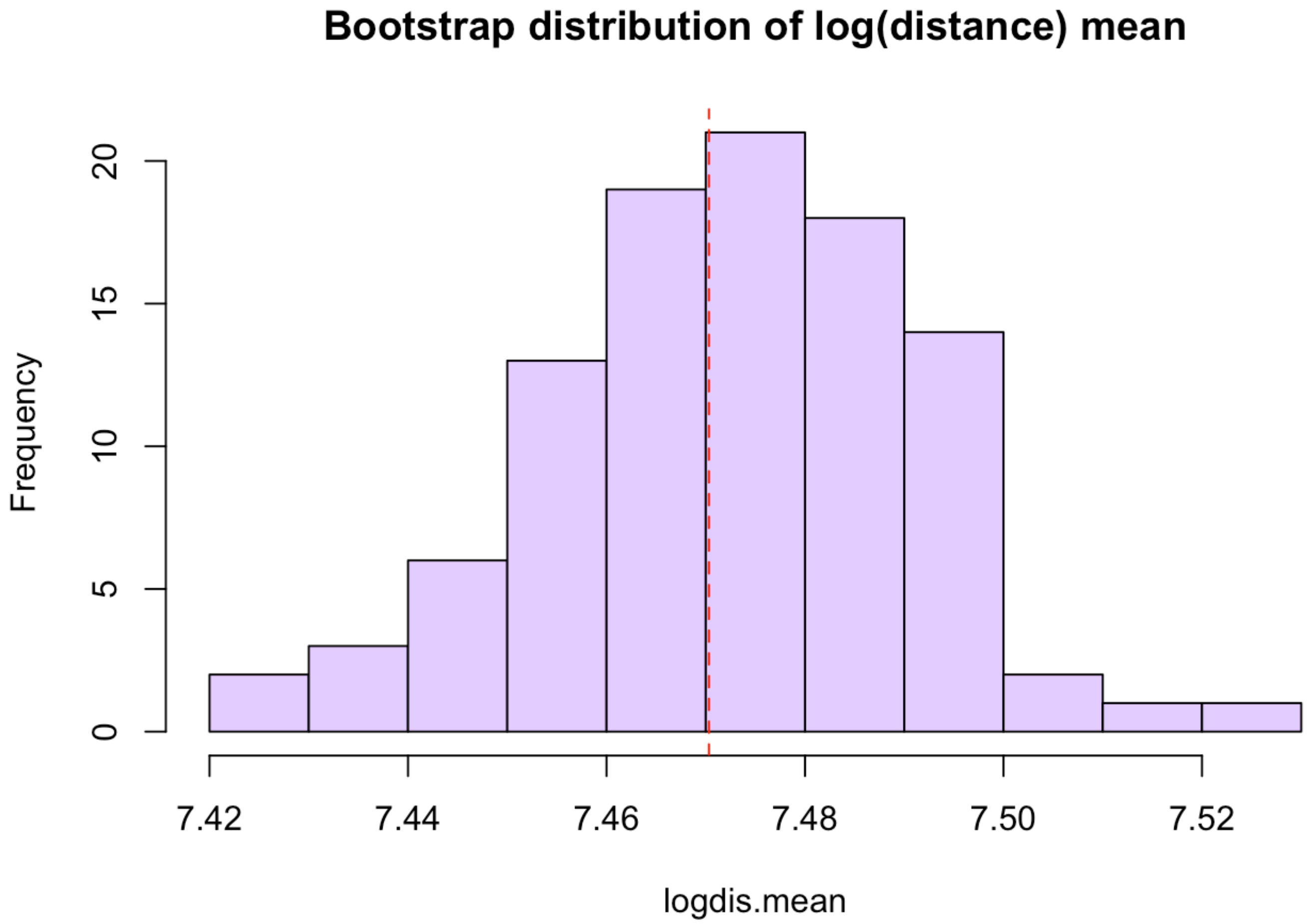
```
Z <- Normal(0, 1)
p_dur = 1 - cdf(Z, abs(z_stat1)) + cdf(Z, -abs(z_stat1))
p_dur# fail to reject null hypothesis, sample mean is equal to population mean at 5%
significance level.
```

```
## [1] 0.942944
```

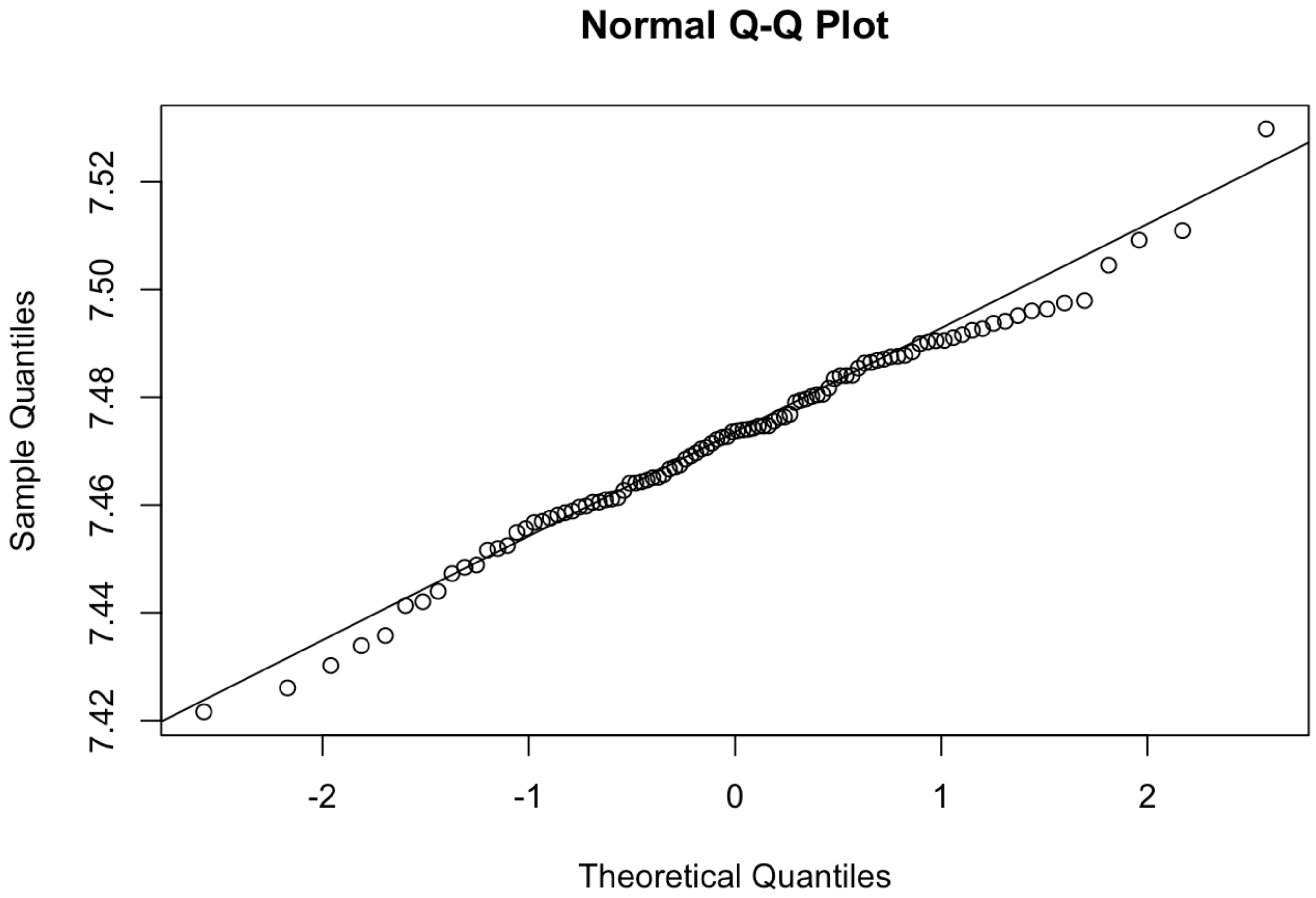
Distance Analysis

```
#remove outliers and plots
#transformation: log of distance
distance_log = log(data$distance)
data = data.frame(data,distance_log)
#remove outliers
data2<-data[!(data$distance_log <=5 | data$distance_log >=12),]
data2 = na.omit(data2)
n2 <- length(data2$distance_log)
logdis.mean <- numeric(N)
for (i in 1:N)
{
  x <- sample(data2$distance_log, n, replace = TRUE)
  logdis.mean[i] <- mean(x) #bootstrap sample mean
}
```

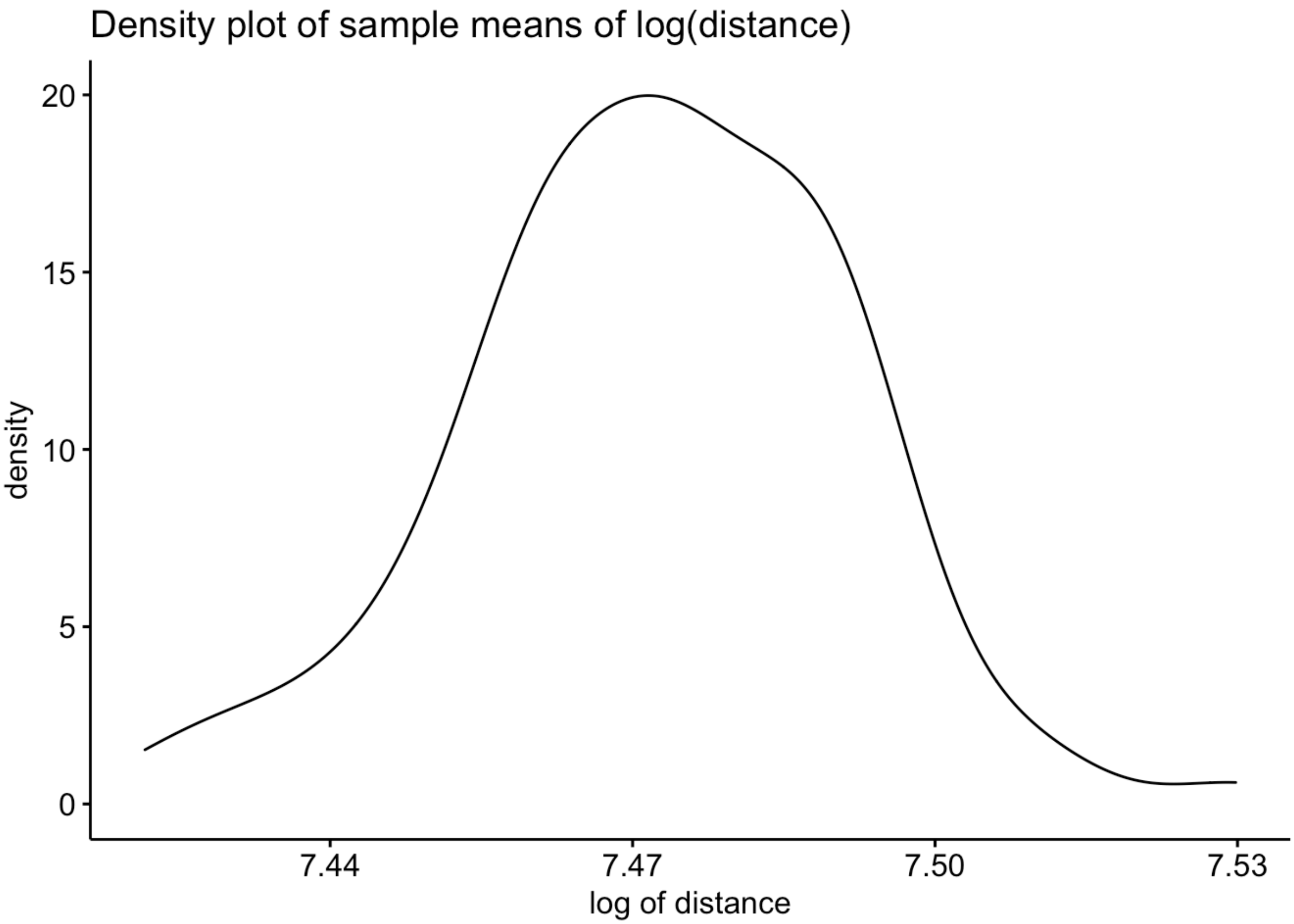
```
hist(logdis.mean, main = "Bootstrap distribution of log(distance) mean",col = '#E5CCFF')
abline(v = mean(data2$distance_log), col = "red", lty = 2)
```



```
qqnorm(logdis.mean)
qqline(logdis.mean)
```



```
#library("ggpubr")
ggdensity(logdis.mean,
          main = "Density plot of sample means of log(distance)",
          xlab = "log of distance")
```



```
#mean and sd
boot.dismean= mean(logdis.mean)
boot.dissd=sd(logdis.mean)
bias.dis = boot.dismean - mean(data2$distance_log)
boot.dismean
```

```
## [1] 7.472443
```

```
bias.dis
```

```
## [1] 0.002125202
```

```
#95% confidence interval
LL.dis=boot.dismean-1.96*boot.dissd
UL.dis=boot.dismean+1.96*boot.dissd
LL.dis
```

```
## [1] 7.43541
```

```
UL.dis
```

```
## [1] 7.509475
```

final project

Jieqian Liu

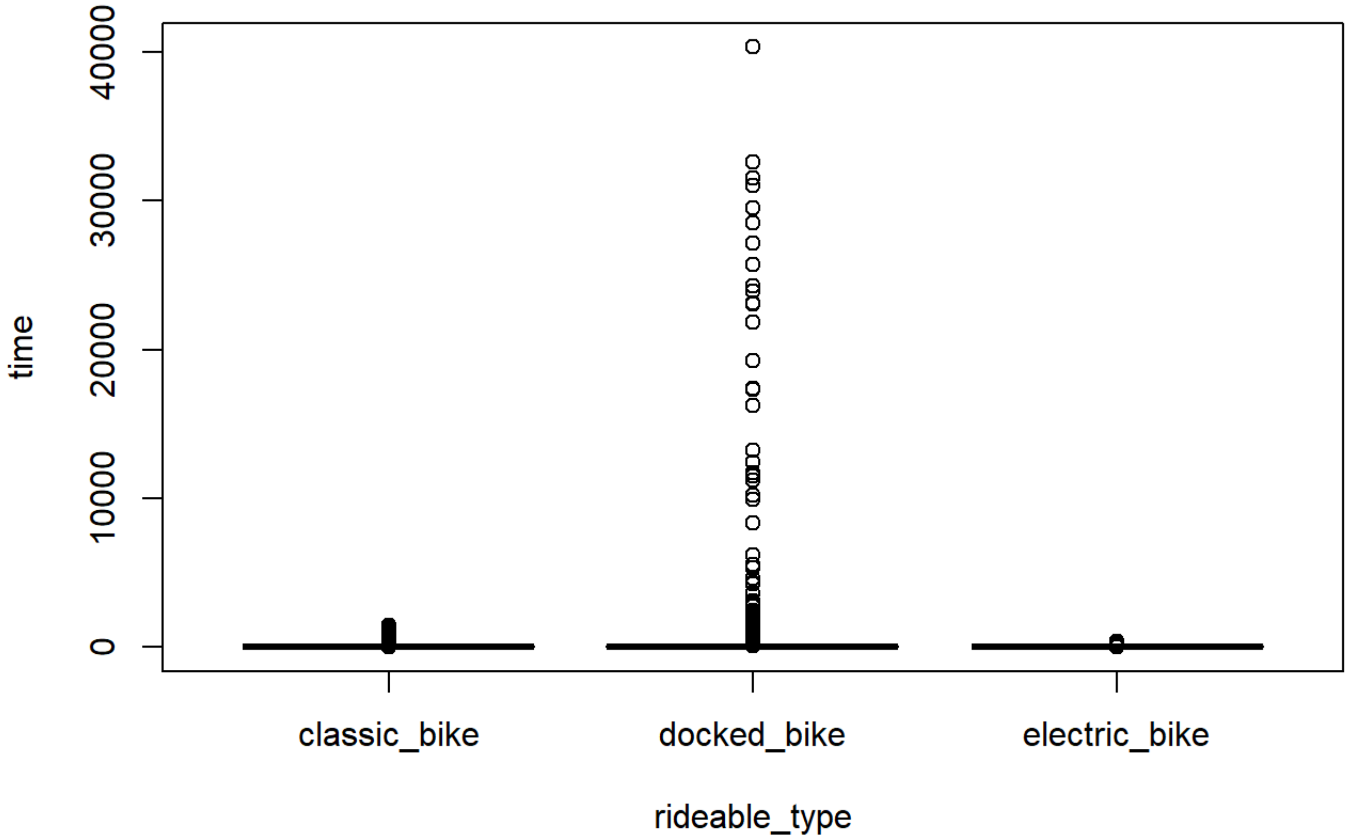
11/23/2021

```
library(dplyr)
library(ggplot2)
library(ggthemes)
library(psych)
library(caTools)
library(e1071)
library(tidyverse)
library(caret)
library(cvms)
```

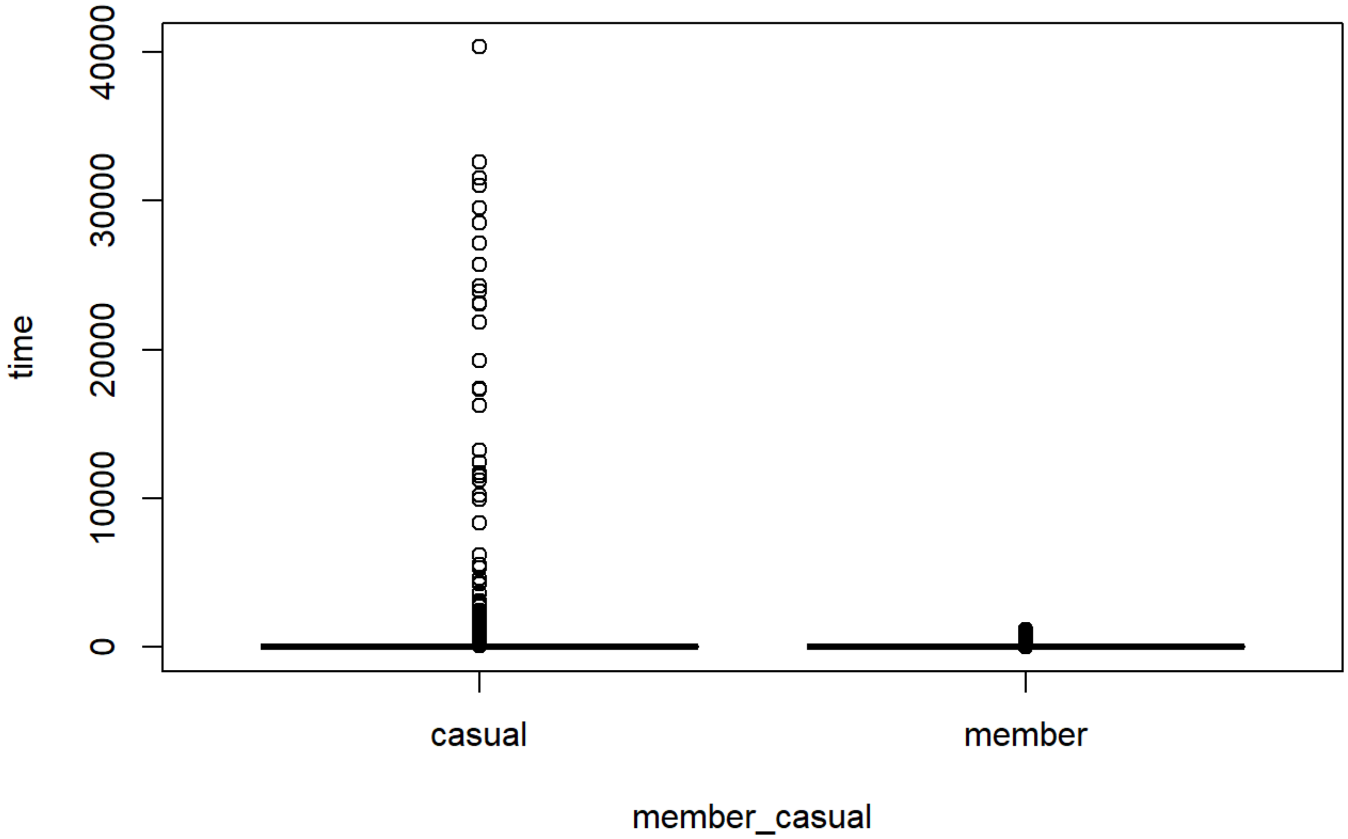
```
# read data
data <- read.csv("511 data.csv")
start_time <- as.POSIXct(paste(data$started_at, data$started_at_time))
end_time <- as.POSIXct(paste(data$ended_at, data$ended_at_time))
data$time <- difftime(end_time, start_time, units = 'mins')
data <- data[,-c(1,3,4,5,6)]
head(data)
```

```
##   rideable_type        start_station_name            end_station_name
## 1  classic_bike Bissell St & Armitage Ave Lincoln Ave & Diversey Pkwy
## 2   docked_bike   Clark St & Winnemac Ave    Broadway & Thorndale Ave
## 3  classic_bike    Michigan Ave & Lake St      Wabash Ave & Wacker Pl
## 4  classic_bike    Wabash Ave & Wacker Pl       Wells St & Concord Ln
## 5  classic_bike      Halsted St & 18th St         Morgan St & 31st St
## 6 electric_bike    Clark St & Elmdale Ave    Broadway & Granville Ave
##   member_casual    time
## 1        casual 46 mins
## 2        casual 10 mins
## 3        casual 28 mins
## 4        casual 16 mins
## 5        casual 13 mins
## 6        member  7 mins
```

```
boxplot(time ~ rideable_type, data = data)
```



```
boxplot(time ~ member_casual, data = data)
```



```
data$type = ""
data[data$rideable_type=="classic_bike",]$type <- 0
data[data$rideable_type=="docked_bike",]$type <- 1
data[data$rideable_type=="electric_bike",]$type <- 2

data$member = ""
data[data$member_casual=="casual",]$member <- 0
data[data$member_casual=="member",]$member <- 1

data$time <- as.numeric(data$time)
data$type <- as.numeric(data$type)
data$member <- as.numeric(data$member)
data$start_station_name <- as.factor(data$start_station_name)
data$end_station_name <- as.factor(data$end_station_name)
```

```
(DataSize=nrow(data))
```

```
## [1] 161010
```

```
TrainingSet_Size<-floor(DataSize*(4/5))
TestSet_Size <- DataSize - TrainingSet_Size
# Sample for row numbers for training set
MyTrainSample <- sample(nrow(data), TrainingSet_Size, replace=FALSE)

train <- data[MyTrainSample,]
# Select testing set using data not in training set
test <- data[-MyTrainSample,]

y.train <- train$rideable_type
y.test <- test$rideable_type

X.train <- train %>% select(-c(rideable_type, type, member_casual))
X.test <- test %>% select(-c(rideable_type, type, member_casual))
```

```
# build the model
model1 <- lm(time~type+member+start_station_name+end_station_name,data=train)
summary(model1)
```

```
## 
## Call:
## lm(formula = time ~ type + member + start_station_name + end_station_name, 
##     data = train)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -3094    -17     -4      8  40172 
## 
## Coefficients: (16 not defined because of singularities)
##                                                                               Estimate
## (Intercept)                                                                  2.481e+01
## type                                                                         2.761e+00
## member                                                                      -2.049e+01
## start_station_name63rd St Beach                                              9.297e+01
## start_station_name900 W Harrison St                                          1.056e+01
## start_station_nameAberdeen St & Jackson Blvd                                 8.982e+00
## start_station_nameAberdeen St & Monroe St                                    3.824e+01
## start_station_nameAberdeen St & Randolph St                                  1.285e+01
## start_station_nameAda St & 113th St                                          2.467e+01
## start_station_nameAda St & Washington Blvd                                   6.203e+00
## start_station_nameAdler Planetarium                                          2.385e+01
## start_station_nameAlbany Ave & 26th St                                      -5.233e+02
## end_station_nameWolcott Ave & Fargo Ave
## end_station_nameWolcott Ave & Polk St
## end_station_nameWood St & 35th St
## end_station_nameWood St & Augusta Blvd
## end_station_nameWood St & Chicago Ave
## end_station_nameWood St & Hubbard St
## end_station_nameWood St & Milwaukee Ave
## end_station_nameWood St & Taylor St (Temp)
## end_station_nameWoodlawn Ave & 55th St
## end_station_nameWoodlawn Ave & 75th St
## end_station_nameWoodlawn Ave & Lake Park Ave
## end_station_nameYates Blvd & 75th St
## end_station_nameYates Blvd & 93rd St
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7685 on 127500 degrees of freedom
## Multiple R-squared:  0.04944,    Adjusted R-squared:  0.0397
## F-statistic: 5.074 on 1307 and 127500 DF,  p-value: < 2.2e-16
```

```
# make predictions
# predictions1 <- predict(model1, X.test)
# Prediction error, RMSE (Root mean square error)
# RMSE(predictions1, y.test)
```

```
nb.model <- naiveBayes(X.train, y.train)
y.pred.nb <- predict(nb.model, X.train)
y.train <- as.factor(y.train)
(confMat <- table(y.pred.nb, y.train))
```

```
##                y.train
## y.pred.nb       classic_bike docked_bike electric_bike
##   classic_bike         53512         675         11989
##   docked_bike          38201       12005         11051
##   electric_bike          783          41           551
```

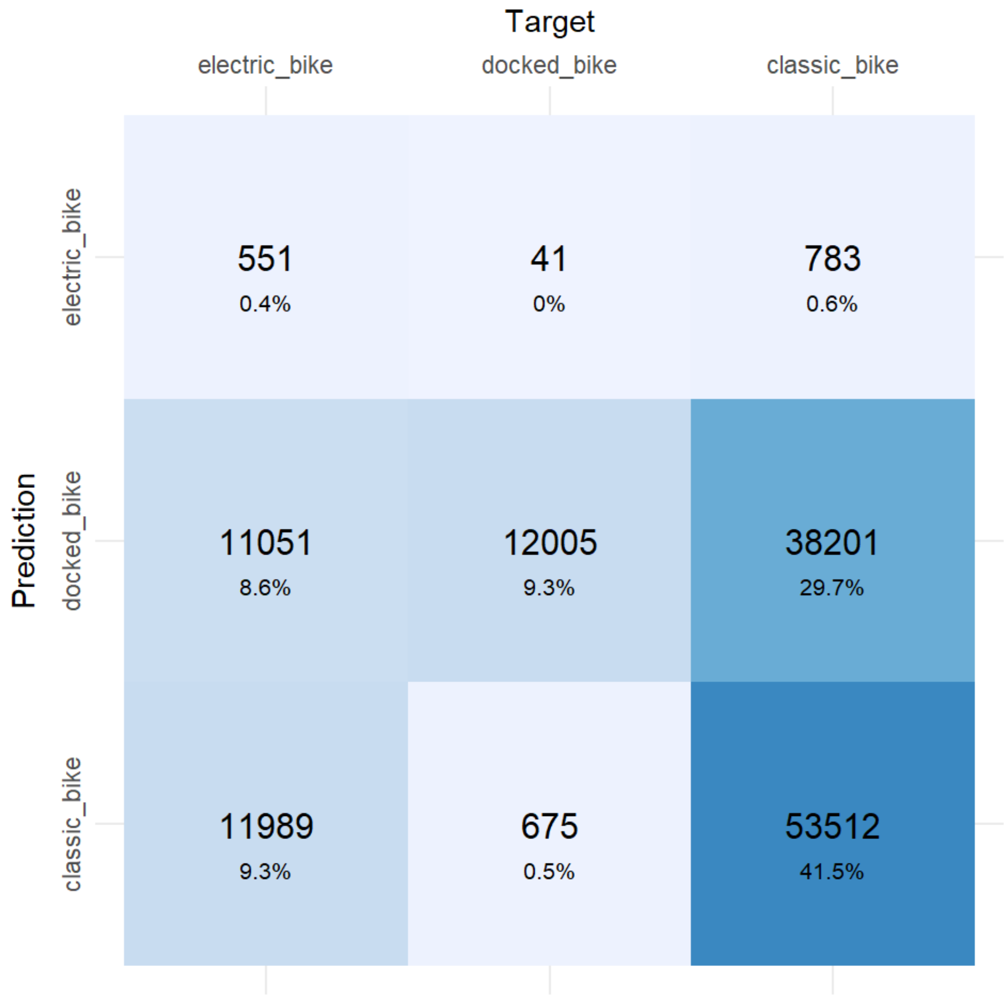
```
# Calculate accuracy
(accuracy <- (confMat[1, 1] + confMat[2, 2] + confMat[3, 3])/sum(colSums(confMat)))
```

```
## [1] 0.5129185
```

```
# Plot Confusion Matrix
(cm1 <- caret::confusionMatrix(y.pred.nb, y.train, positive="true"))
```

```
## Confusion Matrix and Statistics
## 
##                Reference
## Prediction      classic_bike docked_bike electric_bike
##   classic_bike         53512         675         11989
##   docked_bike          38201       12005         11051
##   electric_bike          783          41           551
## 
## Overall Statistics
## 
##                Accuracy : 0.5129
##                  95% CI : (0.5102, 0.5157)
##     No Information Rate : 0.7181
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.1633
## 
##  Mcnemar's Test P-Value : <2e-16
## 
## Statistics by Class:
## 
##                      Class: classic_bike Class: docked_bike
## Sensitivity                       0.5785            0.94372
## Specificity                       0.6512            0.57573
## Pos Pred Value                    0.8086            0.19598
## Neg Pred Value                    0.3776            0.98940
## Prevalence                        0.7181            0.09876
## Detection Rate                    0.4154            0.09320
## Detection Prevalence              0.5138            0.47557
## Balanced Accuracy                 0.6149            0.75972
##                      Class: electric_bike
## Sensitivity                      0.023356
## Specificity                      0.992169
## Pos Pred Value                   0.400727
## Neg Pred Value                   0.819199
## Prevalence                       0.183149
## Detection Rate                   0.004278
## Detection Prevalence             0.010675
## Balanced Accuracy                0.507762
```

```
cmDF1 <- as.data.frame(cm1$table)
plot_confusion_matrix(cmDF1,
                      target_col = "Reference",
                      prediction_col = "Prediction",
                      counts_col = "Freq",
                      add_row_percentages = FALSE,
                      add_col_percentages = FALSE,
                      rm_zero_percentages = FALSE,
                      rm_zero_text = FALSE,
                      add_zero_shading = TRUE,
                      counts_on_top = TRUE)
```



```
ggsave("nb_train.png", height = 6, width = 6)

test$Label = predict(nb.model, X.test)
```

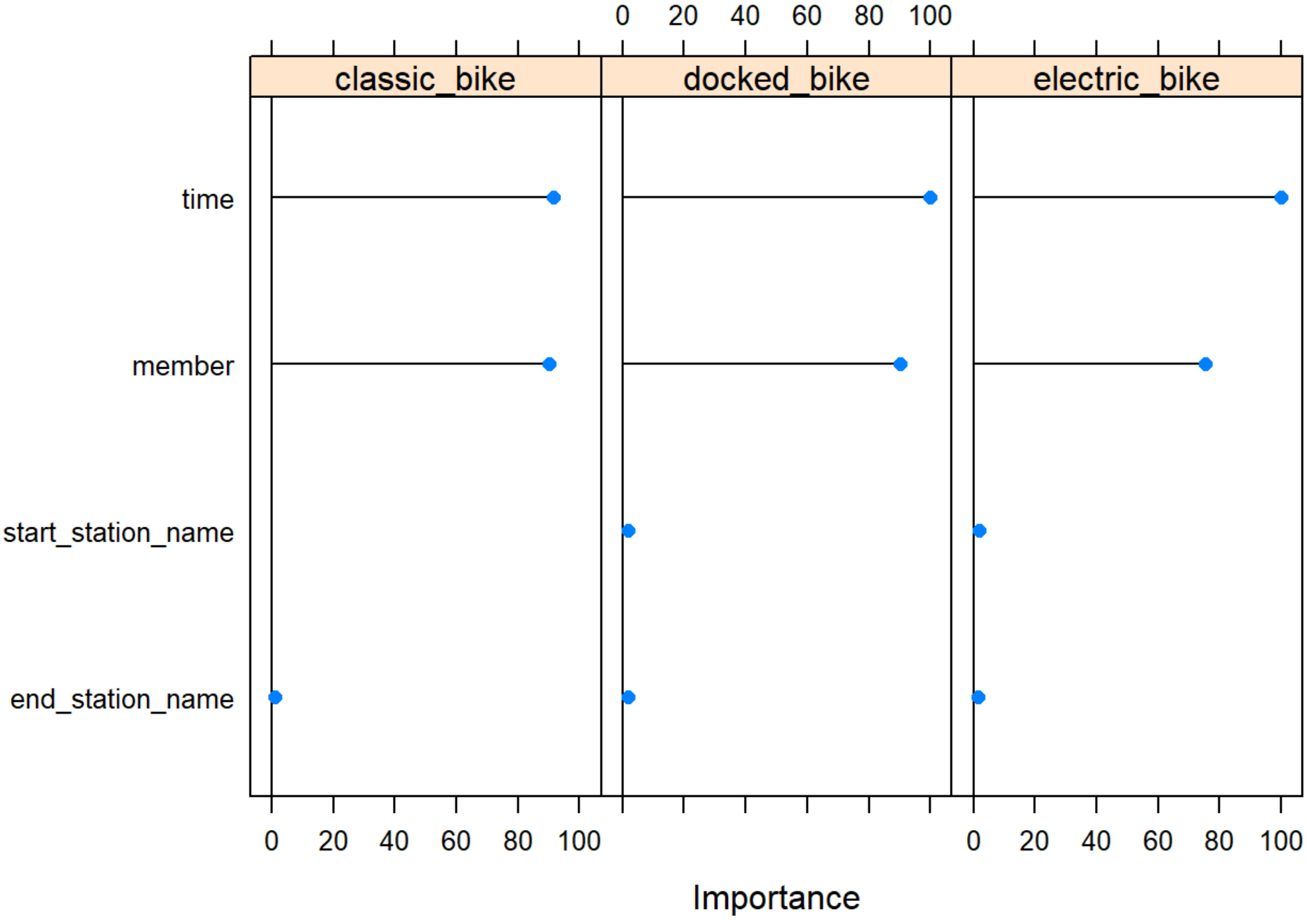
```
model = train(X.train,y.train,'nb')
model$results
```

```
##   usekernel fL adjust  Accuracy     Kappa  AccuracySD     KappaSD
## 1     FALSE  0      1       NaN       NaN          NA          NA
## 2      TRUE  0      1 0.6890352 0.1632561 0.001631307 0.003904021
```

```
Predict <- predict(model, X.test)
table(Predict, y.test)
```

```
##                y.test
## Predict          classic_bike docked_bike electric_bike
##   classic_bike          20647        1681          5295
##   docked_bike            2375        1464           578
##   electric_bike            95          20            47
```

```
X <- varImp(model)
plot(X)
```


classic_bike
docked_bike
electric_bike
time
member
start_station_name
end_station_name
0
20
40
60
80
100
Importance

final_q4

Zheyuan HU

11/26/2021

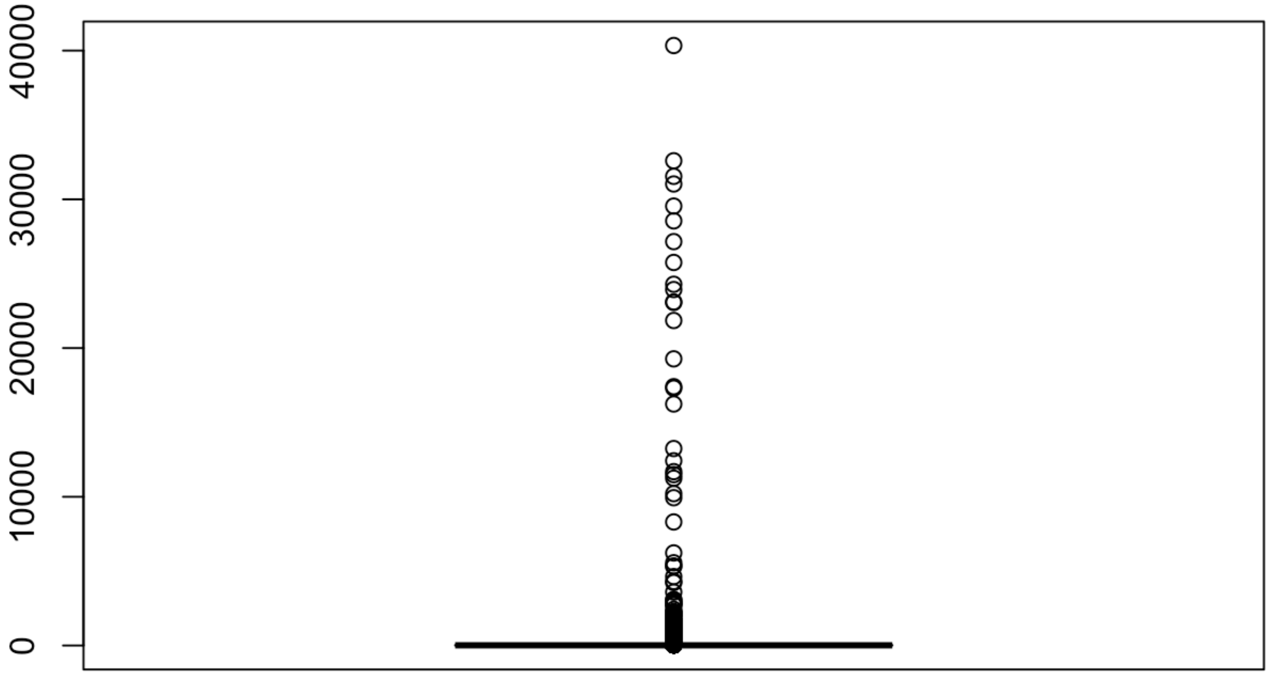
```
library(ggplot2)
library(ggpubr)
set.seed(511)
bikes <- read.csv('511 data.csv')
bikes_c <- bikes  # make a copy
bikes_c$started_at <- as.character(bikes_c$started_at) # transit the time data into s
rting
bikes_c$ended_at <- as.character(bikes_c$ended_at) # transit the time data into srtin
g
str(bikes_c) # info of the dataset
```

```
## 'data.frame':    1048575 obs. of  13 variables:
##  $ ride_id           : Factor w/ 161011 levels "","0000453517CABB51",..: 114914 40
311 87142 134424 133741 140526 132731 131231 14584 81635 ...
##  $ rideable_type     : Factor w/ 4 levels "","classic_bike",..: 2 3 2 2 2 4 2 2 2
2 ...
##  $ started_at        : chr  "7/1/2021 0:00" "7/1/2021 0:00" "7/1/2021 0:00" "7/1/2
021 0:00" ...
##  $ ended_at          : chr  "7/1/2021 0:46" "7/1/2021 0:10" "7/1/2021 0:28" "7/1/2
021 0:16" ...
##  $ start_station_name: Factor w/ 665 levels "","2112 W Peterson Ave",..: 50 150 41
7 608 278 134 421 134 390 581 ...
##  $ start_station_id  : Factor w/ 664 levels "","13001","13006",..: 22 643 514 573
38 465 2 465 487 9 ...
##  $ end_station_name  : Factor w/ 672 levels "","2112 W Peterson Ave",..: 387 63 61
5 624 444 60 426 60 522 588 ...
##  $ end_station_id    : Factor w/ 670 levels "","13001","13006",..: 562 154 579 624
621 153 2 153 53 9 ...
##  $ start_lat         : num  41.9 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : Factor w/ 3 levels "","casual","member": 2 2 2 2 2 3 2 3 2
2 ...
```

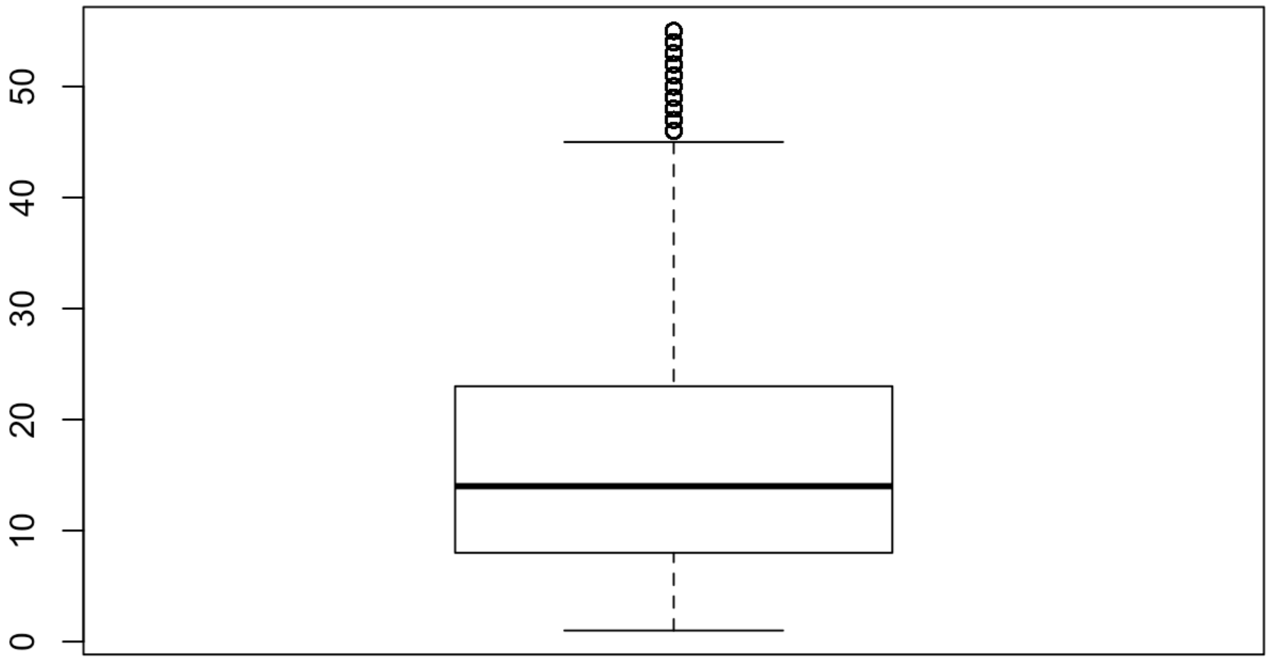
```
############
# clean the data
bikes_c$start_time <- strptime(bikes_c$started_at, '%m/%d/%Y %H:%M')
bikes_c$end_time <- strptime(bikes_c$ended_at, '%m/%d/%Y %H:%M')
bikes_c$time_use <- difftime(bikes_c$end_time, bikes_c$start_time, units = 'mins') # calculate the time lasted
bikes_c$time_use <- as.numeric(bikes_c$time_use)
# drop the NA value for member_casual and 0 for time_use
ind <- which(bikes_c$member_casual == '' | bikes_c$time_use == 0)
# remove
bikes_c <- bikes_c[-ind,]
bikes_c$member_casual <- droplevels(bikes_c$member_casual)
nrow(bikes_c)
```

```
## [1] 159776
```

```
# boxplot
boxplot(bikes_c$time_use) # we need to remove the outlier since some of them are probably wrong
out <- boxplot(bikes_c$time_use)$out
```



```
bikes_c <- bikes_c[!bikes_c$time_use %in% out,]
# boxplot after removing the outliers
boxplot(bikes_c$time_use)
```



```
# check the info
str(bikes_c)
```

```
## 'data.frame':    147170 obs. of  16 variables:
##  $ ride_id           : Factor w/ 161011 levels "","0000453517CABB51",..: 114914 40
311 87142 134424 133741 140526 132731 131231 14584 118572 ...
##  $ rideable_type     : Factor w/ 4 levels "","classic_bike",..: 2 3 2 2 2 4 2 2 2
2 ...
##  $ started_at        : chr  "7/1/2021 0:00" "7/1/2021 0:00" "7/1/2021 0:00" "7/1/2
021 0:00" ...
##  $ ended_at          : chr  "7/1/2021 0:46" "7/1/2021 0:10" "7/1/2021 0:28" "7/1/2
021 0:16" ...
##  $ start_station_name: Factor w/ 665 levels "","2112 W Peterson Ave",..: 50 150 41
7 608 278 134 421 134 390 278 ...
##  $ start_station_id  : Factor w/ 664 levels "","13001","13006",..: 22 643 514 573
38 465 2 465 487 38 ...
##  $ end_station_name  : Factor w/ 672 levels "","2112 W Peterson Ave",..: 387 63 61
5 624 444 60 426 60 522 444 ...
##  $ end_station_id    : Factor w/ 670 levels "","13001","13006",..: 562 154 579 624
621 153 2 153 53 621 ...
##  $ start_lat         : num  41.9 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : Factor w/ 2 levels "casual","member": 1 1 1 1 1 2 1 2 1 1 .
..
##  $ start_time        : POSIXlt, format: "2021-07-01 00:00:00" "2021-07-01 00:00:00
" ...
##  $ end_time          : POSIXlt, format: "2021-07-01 00:46:00" "2021-07-01 00:10:00
" ...
##  $ time_use          : num  46 10 28 16 13 7 41 6 20 15 ...
```

```
###############
# cleaning finished
table(bikes_c$member_casual)
```

```
##
## casual member
##  79134  68036
```

```
################
# sample for casual and member, make sure they are balanced
samp_c <- sample(which(bikes_c$member_casual == 'casual'), size = 5000)
samp_m <- sample(which(bikes_c$member_casual == 'member'), size = 5000)
total_samp <- rbind(bikes_c[samp_c,], bikes_c[samp_m,])
casual_rider <- bikes_c[samp_c,]
member_rider <- bikes_c[samp_m,]
nrow(total_samp)
```

```
## [1] 10000
```

```
aggregate(total_samp$time_use, by = list(status = total_samp$member_casual), FUN = me
an)
```

```
##   status       x
## 1 casual 19.2654
## 2 member 13.5388
```

```
aggregate(total_samp$time_use, by = list(status = total_samp$member_casual), FUN = sd
)
```

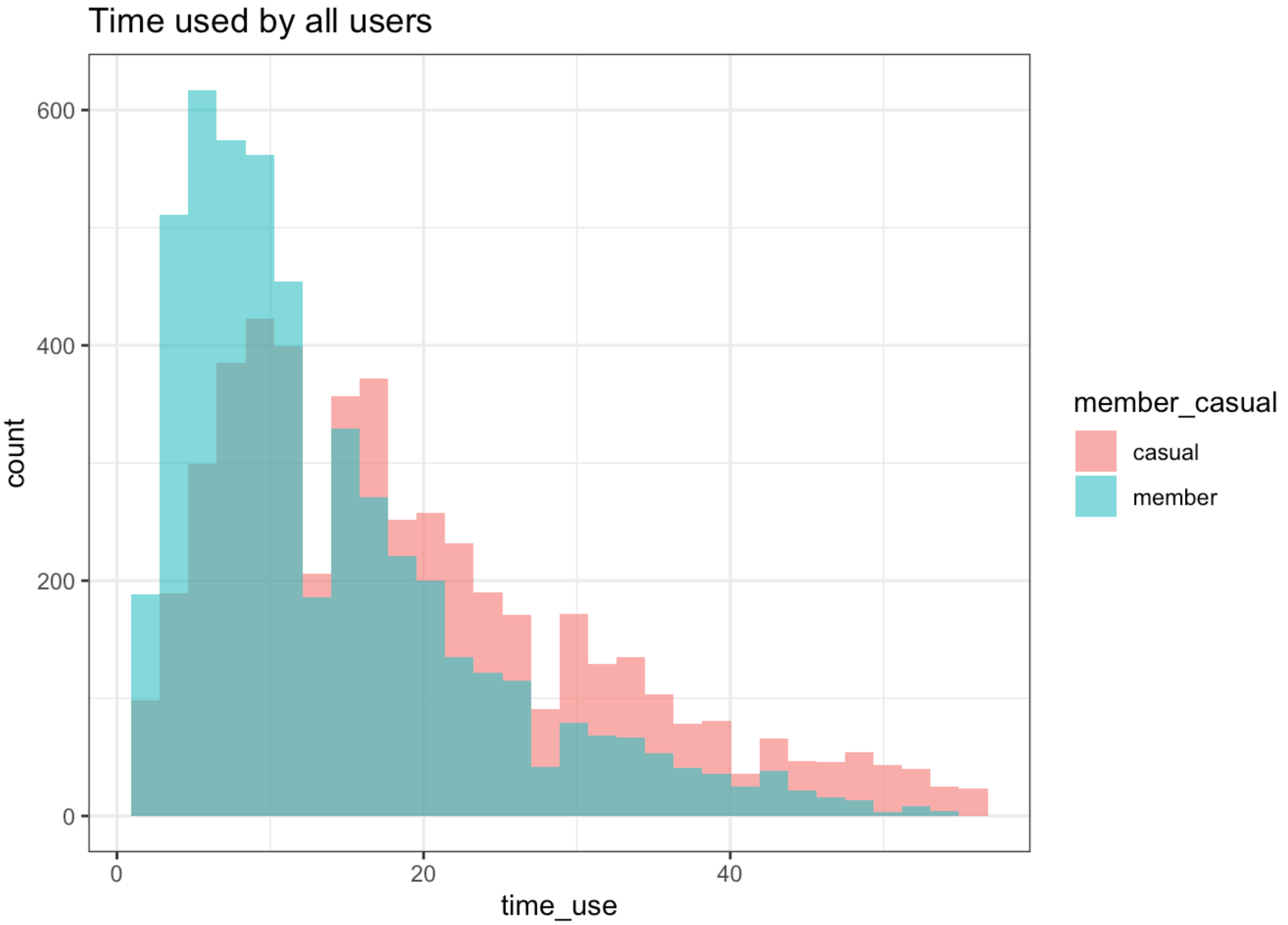
```
##   status        x
## 1 casual 12.28221
## 2 member  9.92009
```

```
aggregate(total_samp$time_use, by = list(status = total_samp$member_casual), FUN = su
mmary)
```

```
##   status  x.Min. x.1st Qu. x.Median  x.Mean x.3rd Qu.  x.Max.
## 1 casual  1.0000   10.0000  16.0000 19.2654   26.0000 55.0000
## 2 member  1.0000    6.0000  11.0000 13.5388   18.0000 54.0000
```

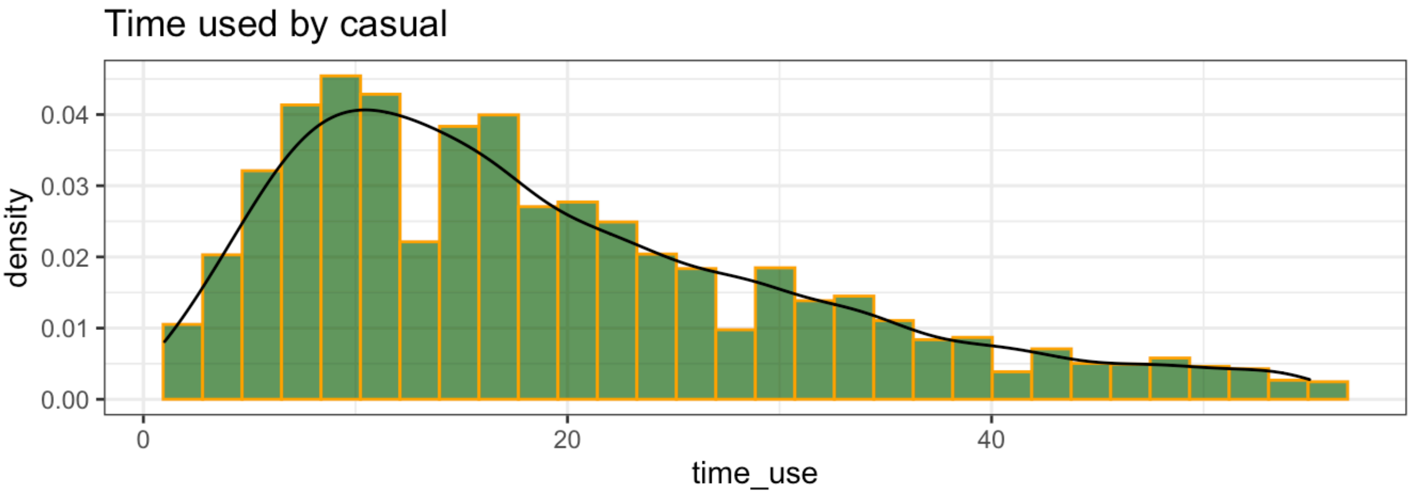
```
# basic visualizations
### total visu
count_plot <- ggplot(data = total_samp, aes(x = time_use, fill = member_casual))
count_plot <- count_plot +
  geom_histogram(alpha = 0.6, position = 'identity') +
  theme_bw() +
  labs(title = 'Time used by all users')

count_plot
```

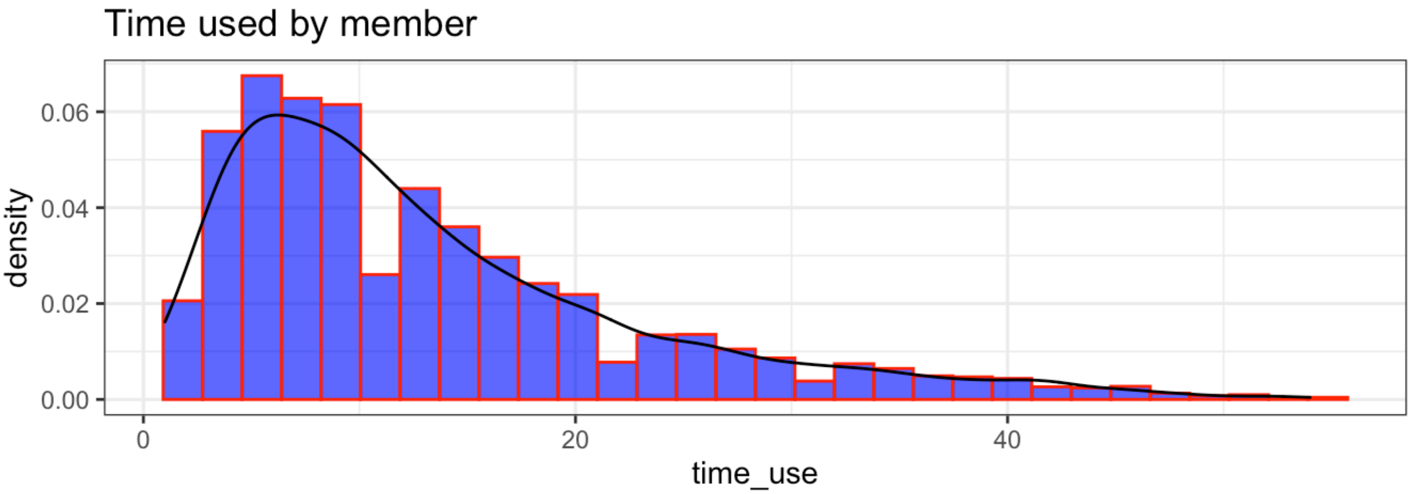
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
### seperate
casual_plot <- ggplot(data = casual_rider, aes(x = time_use))
casual_plot <- casual_plot +
  geom_histogram(aes(y = ..density..), fill = 'darkgreen', color = 'orange', alpha =
0.7) +
  theme_bw() +
  geom_density(alpha = 0.1, color = 'black') +
  labs(title = 'Time used by casual')

member_plot <- ggplot(data = member_rider, aes(x = time_use))
member_plot <- member_plot +
  geom_histogram(aes(y = ..density..), fill = 'blue', color = 'red', alpha = 0.7) +
  theme_bw() +
  geom_density(alpha = 0.1, color = 'black') +
  labs(title = 'Time used by member')

ggarrange(casual_plot, member_plot, ncol = 1, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





```
## basic visualizations end
##############
# t-test
# our guess is members should ride longer than casuals
# first, do the t-test to see if the members ride longer than casuals
t.test(member_rider$time_use, casual_rider$time_use, alt = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  member_rider$time_use and casual_rider$time_use
## t = -25.648, df = 9574.2, p-value = 1
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -6.093892       Inf
## sample estimates:
## mean of x mean of y
##   13.5388   19.2654
```

```
# p-value is 1 which means members' using is defintely not longer than casuals'
# so, next we do t-test to see if the members ride shorter than casuals
t.test(member_rider$time_use, casual_rider$time_use, alt = 'less')
```

```
## 
##  Welch Two Sample t-test
## 
## data:  member_rider$time_use and casual_rider$time_use
## t = -25.648, df = 9574.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -5.359308
## sample estimates:
## mean of x mean of y 
##   13.5388   19.2654
```

```
# p-value is almost 0 which means the member actually ride shorter than casuals significantly.
```