Bridj Customer Data Analysis Report

Niranjan Adhikari

April 28, 2018

Introduction

This manuscript presents the analytical report of the dataset *'bridj_bookings.csv'* from Bridj trip bookings for the month of March 2018. The aim of this report is to analyze the given data using statistical software and ultimately provide the pieces of information of the number of customers successfully picked by Bridj over the time, passenger distribution over time and identify the location of the customer. Bridj always optimizes the service delivered by the dynamic rooting, stopping and cloud clustering of the passenger. The remainder of the report is organized as follows. The next session briefly describes the physical meaning of the dataset and the following session demonstrates the methodology, necessary libraries, and finally data manipulation and visualization.

Information of Dataset

The dataset contains the 13 variables and 3426 number of observations. The meaning of each variable are presented below.

- BOOKING_ID : Id of the booking. Values of -1 identifies customers who searched for a trip but did not actually make a booking.
- PASSENGER_ID : Id of the passenger booking the trip. Values of -1 identifies passenger that have since deregistered from the app.
- NUMBER_OF_PASSENGERS : Number of passengers in the booking.
- BOOKING_TIME : Date and time at which the booking occurred.
- SUCCESSFUL_PICK_UP : Boolean entry to identify bookings where the customer was picked up by the bus.
- SCHEDULED_PICK_UP_DATETIME : Date and time at which the pick was scheduled.
- PICK_UP_LOCATION_ID : Name of the pick up stop.
- PICK_UP_LATITUDE : Latitude of the pick up stop requested by the customer.
- PICK_UP_LONGITUDE : Longitude of the pick up stop requested by the customer.
- SCHEDULED_DROP_OFF_DATETIME : Date and time at which the drop off was scheduled.
- DROP_OFF_LOCATION_ID : Name of the drop off stop.
- DROP_OFF_LATITUDE : Latitude of the drop off stop requested by the customer.
- DROP_OFF_LONGITUDE : Longitude of the drop off stop requested by the customer

Methodology and Steps

The dataset has been analyzed using statistical package R and R studio. The details of the machine are presented below, and R is free and compatible with windows and mac.

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          4.0
## year           2017
## month          04
## day            21
## svn rev        72570
## language       R
## version.string R version 3.4.0 (2017-04-21)
## nickname       You Stupid Darkness
```

The necessary steps used are summarized below

1. Load the required libraries into R

2. Upload the dataset into R and observe the dimension and string of each variable.

3. Do data wrangling, data quality check and finally tidy up the data using tidyverse package

4. Get insights from the data using statistical formula and suitable visualization

Required Libraries

This report uses following libraries in R.

```
library(tidyverse)     ## data wrangling
library(lubridate)     ## date and time
library(stringr)       ## string manupulation
library(ggplot2)       ## data visualisation
library(scales)        ## axis lable manupulation
library(ggmap)         ## map visualisation
```

Data Manupulation and Visualization

```
bridj <- read_csv('bridj_bookings.csv')
glimpse(bridj)
```

```
## Observations: 3,426
## Variables: 13
## $ BOOKING_ID                  <int> -1, 896, -1, 911, 626, -1, -1, -1,...
## $ PASSENGER_ID                <int> -1, 1576, 1544, 1609, 137, 1279, 9...
## $ NUMBER_OF_PASSENGERS        <int> NA, 2, NA, 3, 1, NA, NA, NA, 1, NA...
## $ BOOKING_TIME                <chr> "2018-03-20 15:52", "2018-03-30 17...
## $ SUCCESSFUL_PICK_UP          <chr> NA, "NO", NA, "YES", "YES", NA, NA...
## $ SCHEDULED_PICK_UP_DATETIME  <chr> NA, "2018-03-30 17:33", NA, "2018-...
## $ PICK_UP_LOCATION_ID         <chr> NA, "Wet n' Wild Parramatta Depart...
## $ PICK_UP_LATITUDE            <dbl> 42.35518, -33.80697, -33.81901, -3...
## $ PICK_UP_LONGITUDE           <dbl> -71.16293, 150.90862, 151.00177, 1...
## $ SCHEDULED_DROP_OFF_DATETIME <chr> NA, "2018-03-30 18:03", NA, "2018-...
## $ DROP_OFF_LOCATION_ID        <chr> NA, "Parramatta Interchange Arrive...
## $ DROP_OFF_LATITUDE           <dbl> -71.16293, -33.81759, 151.00177, -...
## $ DROP_OFF_LONGITUDE          <dbl> 42.35518, 151.00486, -33.81901, 15...
```

We have 13 variables and 3426 number of observations in the Bridj customer booking dataset. It is clearly observed that data is not tidy to do the further analysis so we have to clean the data first. Since R reads the time variable as a character string, so it needs to be converted into date-time format using ymd_hm() function from lubricate library.

```
bridj$BOOKING_TIME <- ymd_hm(bridj$BOOKING_TIME)
bridj$SCHEDULED_PICK_UP_DATETIME <- ymd_hm(bridj$SCHEDULED_PICK_UP_DATETIME)
bridj$SCHEDULED_DROP_OFF_DATETIME <- ymd_hm(bridj$SCHEDULED_DROP_OFF_DATETIME)
```

Now, lets create corresponding tables each for searcher and actual booking customers which are bridj_searcher and bridj_booking respectively.

```
# create searcher vector which returns corresponding rows number
searcher <- which(bridj$BOOKING_ID == -1)
## table for searching customer
bridj_searcher <- bridj[searcher,]
# glimpse(bridj_searcher) ## 2222 customer just searches the bridj

## table for actual booking made
bridj_booking <- bridj[-searcher,]
# glimpse(bridj_booking) ## 1204 customer made booking
missing <- which(is.na(bridj_booking$BOOKING_ID)) ## Checking if any booking id missing
bridj_booking <- bridj_booking[order(bridj_booking$BOOKING_ID),]
glimpse(bridj_booking)
```

```
## Observations: 1,204
## Variables: 13
## $ BOOKING_ID                  <int> 623, 623, 623, 623, 624, 624, 624,...
## $ PASSENGER_ID                <int> 519, 519, 519, 519, 252, 252, 252,...
## $ NUMBER_OF_PASSENGERS        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ BOOKING_TIME                <dttm> 2018-02-28 13:23:00, 2018-02-28 1...
## $ SUCCESSFUL_PICK_UP          <chr> "YES", "YES", "YES", "YES", "YES",...
## $ SCHEDULED_PICK_UP_DATETIME  <dttm> 2018-03-01 07:45:00, 2018-03-01 0...
## $ PICK_UP_LOCATION_ID         <chr> "Prairiewood T-Way Station - To Pa...
## $ PICK_UP_LATITUDE            <dbl> -33.85872, -33.85872, -33.85872, -...
## $ PICK_UP_LONGITUDE           <dbl> 150.8969, 150.8969, 150.8969, 150....
## $ SCHEDULED_DROP_OFF_DATETIME <dttm> 2018-03-01 08:04:00, 2018-03-01 0...
## $ DROP_OFF_LOCATION_ID        <chr> "33 Bentley Street", "33 Bentley S...
## $ DROP_OFF_LATITUDE           <dbl> -33.84535, -33.84535, -33.84535, -...
## $ DROP_OFF_LONGITUDE          <dbl> 150.8835, 150.8835, 150.8835, 150....
```

Here, data duplicate was observed, which means data redundancy. Then further searching and analyzing of original *CSV* file confirmed the presence of this issue. Let's get rid of duplicate first.

```
dupes <- which(duplicated(bridj_booking))
# print(dupes)
bridj_actual_booking <- bridj_booking[-dupes,]
```

To perform the further analysis, the cleaned bridj_booking table splits into two corresponding tables for the customers who used Bridj services and customer who missed the Bridj services respectively.

```
successful <- which(bridj_actual_booking$SUCCESSFUL_PICK_UP == "YES")
# print(successful)

## table for successfully picked up customer
bridj_successful <- bridj_actual_booking[successful,]
bridj_successful <- bridj_successful[order(bridj_successful$BOOKING_ID),]

# glimpse(bridj_successful)
which(is.na(bridj_successful))
```

```
## integer(0)
```

```
a <- nrow(bridj_successful)
a
```

```
## [1] 266
```

```
## Table for unsuccessful customer
bridj_unsuccessful <- bridj_actual_booking[-successful,]
# glimpse(bridj_unsuccessful)
which(is.na(bridj_unsuccessful))
```

```
## integer(0)
```

```
b<- nrow(bridj_unsuccessful)
b
```

```
## [1] 33
```

```
## let's calculate the rate of successful picked up
successrate <- a/(a+b)
round(successrate, 2)
```

```
## [1] 0.89
```

Solution of QN 1

With the above operations, it is confirmed that we got rid of the duplicate issue and no missing values in the dataset. Since our dataset contains only March 2018 transactions so we can safely say that 266 number of customers were successfully picked up by Bridj while only 33 pick up were unsuccessful. This results in approximately 89% success rate of Bridj for March 2018.

Solution of QN 3

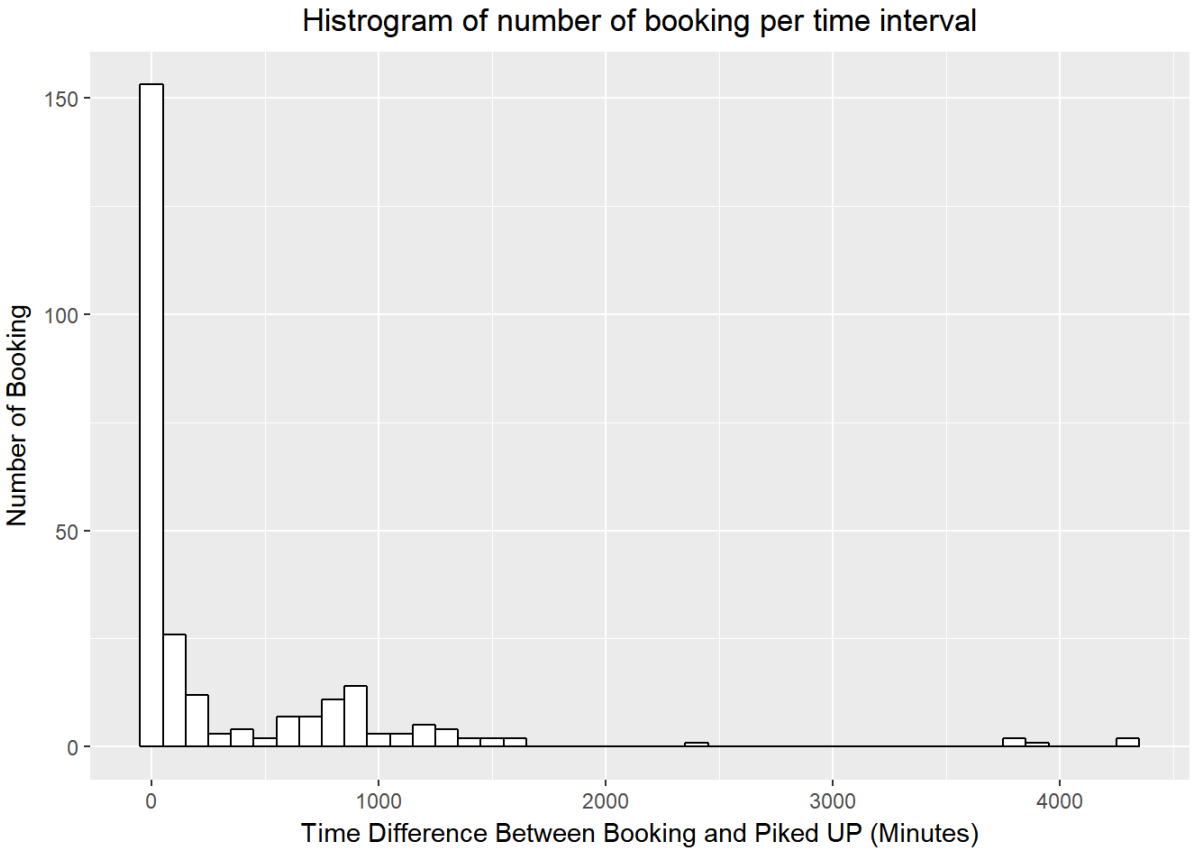
The following code of chunks returns the time differece in minutes between booking time and scheduled picking up time from the locations.

```
bridj_successful$interval <- difftime(bridj_successful$SCHEDULED_PICK_UP_DATETIME, bridj_successful$BOOKING
_TIME, units = "mins")
summary(as.numeric(bridj_successful$interval))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -40.0    10.0    29.0   336.1   514.5  4311.0
```

The following histogram visually presents the diffrences of booking and pick up time distribution.

```
ggplot(data=bridj_successful, aes(interval)) +
geom_histogram(binwidth=100,
               colour="black", fill="white") +
  ggtitle("Histrogram of number of booking per time interval")+
  ylab("Number of Booking")+ xlab("Time Difference Between Booking and Piked UP (Minutes)")+
  theme(plot.title = element_text(hjust = 0.5))
```



From the above bar, it is observed that majority of customers prefer to book on time or within a few hours while some of them booked in more advance.

```
abnormal <- which(bridj_successful$interval < 0)
print(bridj_successful[abnormal,c(1,4,6,14)])
```

```
## # A tibble: 13 x 4
##    BOOKING_ID BOOKING_TIME        SCHEDULED_PICK_UP_DATETIME interval
##         <int> <dttm>              <dttm>                     <time>
##  1        628 2018-03-01 16:48:00 2018-03-01 16:47:00        -1
##  2        709 2018-03-16 17:31:00 2018-03-16 17:30:00        -1
##  3        723 2018-03-17 14:25:00 2018-03-17 14:12:00        -13
##  4        759 2018-03-20 16:13:00 2018-03-20 15:33:00        -40
##  5        771 2018-03-21 18:38:00 2018-03-21 18:34:00        -4
##  6        782 2018-03-23 17:31:00 2018-03-23 17:30:00        -1
##  7        804 2018-03-25 09:31:00 2018-03-25 09:30:00        -1
##  8        821 2018-03-26 17:32:00 2018-03-26 17:30:00        -2
##  9        848 2018-03-29 17:33:00 2018-03-29 17:30:00        -3
## 10        856 2018-03-30 09:43:00 2018-03-30 09:35:00        -8
## 11        857 2018-03-30 09:43:00 2018-03-30 09:35:00        -8
## 12        871 2018-03-30 11:34:00 2018-03-30 11:32:00        -2
## 13        922 2018-03-31 10:09:00 2018-03-31 10:08:00        -1
```

Here I have found some anomaly (negative difference time) in the dataset that few customers booking time returns after picked up. I got suspicious of two reasons that first was due to the software app delayed to update information in the database and second is just wrong information noted in the dataset, this can be treated as a present of outliers. I know that how to deal with outliers.

Solution of QN 2

Since booking time, scheduled pickup and scheduled drop off columns are in the date-time format, so to do the further analysis the strings character of the variables have to be separated into two corresponding columns. The newly created columns separately retain the information of date and time respectively.

```
bridj_successful <- separate(bridj_successful, BOOKING_TIME, c("BOOKING_DAY", "BOOKING_TIME"), sep=" ", rep
lace = TRUE)
bridj_successful <- separate(bridj_successful, SCHEDULED_PICK_UP_DATETIME, c("PICKUP_DAY", "PICKUP_TIME"),
 sep=" ", replace = TRUE)
bridj_successful <- separate(bridj_successful, SCHEDULED_DROP_OFF_DATETIME, c("DROPOFF_DAY", "DROPOFF_TIME"
), sep=" ", replace = TRUE)


bridj_successful %>%
  group_by(PICKUP_DAY) %>%
  summarise(NoPassenger= sum(NUMBER_OF_PASSENGERS)) %>%
  filter(NoPassenger== max(NoPassenger))
```

```
## # A tibble: 1 x 2
##   PICKUP_DAY NoPassenger
##   <chr>            <int>
## 1 2018-03-31          78
```

```
## Summary of passengers who used Bridj services
bridj_successful %>%
  group_by(PICKUP_DAY) %>%
  summarise(NoPassenger= sum(NUMBER_OF_PASSENGERS)) %>%
  summary(NoPassenger)
```

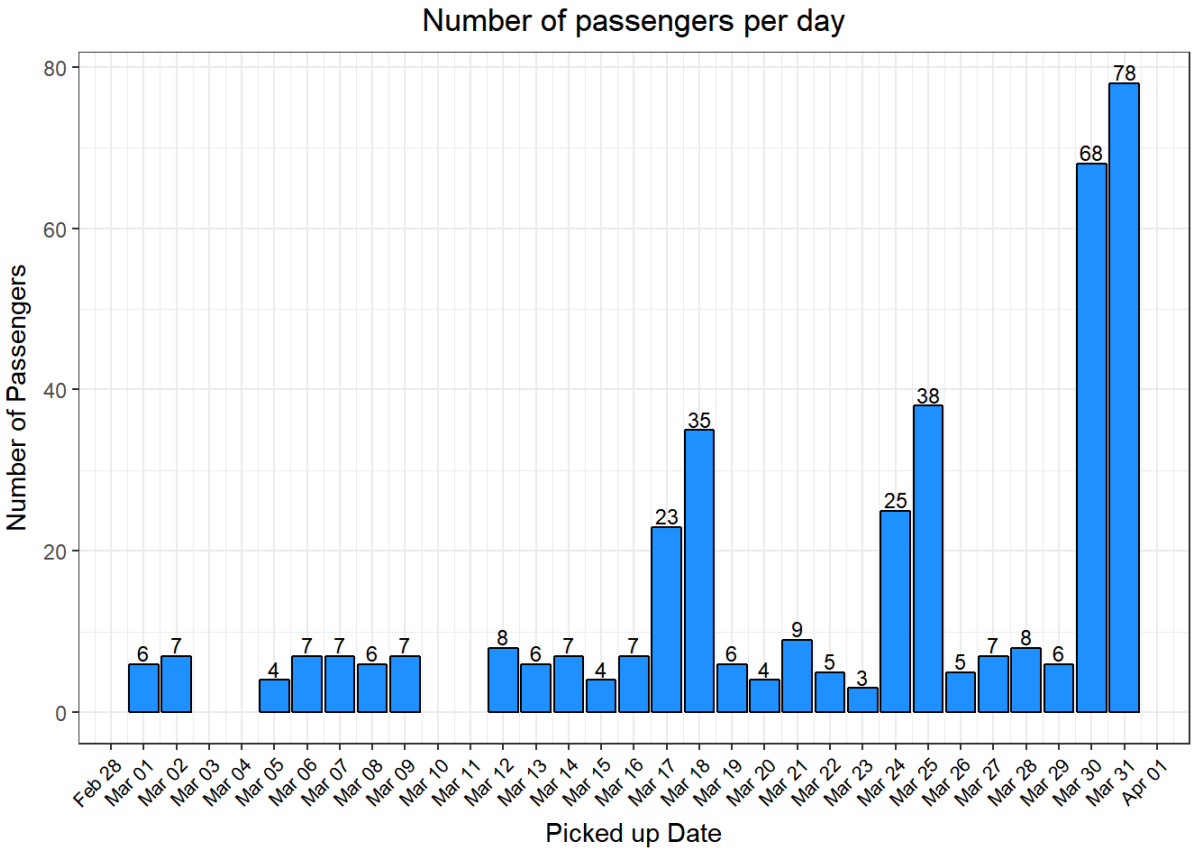
```
##   PICKUP_DAY         NoPassenger
##  Length:27          Min.   : 3.00
##  Class :character   1st Qu.: 6.00
##  Mode  :character   Median : 7.00
##                     Mean   :14.67
##                     3rd Qu.: 8.50
##                     Max.   :78.00
```

On March 31, 2018, the maximum number of passengers used the services which are 78. It is also observed that mean number of the passengers for March is 14. More insights of the passenger distribution over the time is visualized below.

```
no_passenger <- bridj_successful %>%
  group_by(PICKUP_DAY) %>%
  summarise(NoPassenger= sum(NUMBER_OF_PASSENGERS))

no_passenger$PICKUP_DAY <- ymd(no_passenger$PICKUP_DAY)

## Visual display of distribution of passenger used Bridge services
ggplot(data=no_passenger, aes(x = PICKUP_DAY, y=NoPassenger)) +
  geom_bar(stat = "identity",color="black", fill = "dodgerblue")+
  ggtitle("Daily visitors Volume")+
  theme_bw() +
  scale_x_date(breaks = date_breaks("1 day"),  date_labels = "%b %d") +
  theme(axis.text.x=element_text(angle=45,hjust=1, size = 8, color = "black"),
        plot.title = element_text(hjust = 0.5)) +
  labs(title = "Number of passengers per day",
       y = "Number of Passengers",
       x = "Picked up Date") +
  geom_text(aes(label=NoPassenger), vjust = -0.2,size = 3)
```

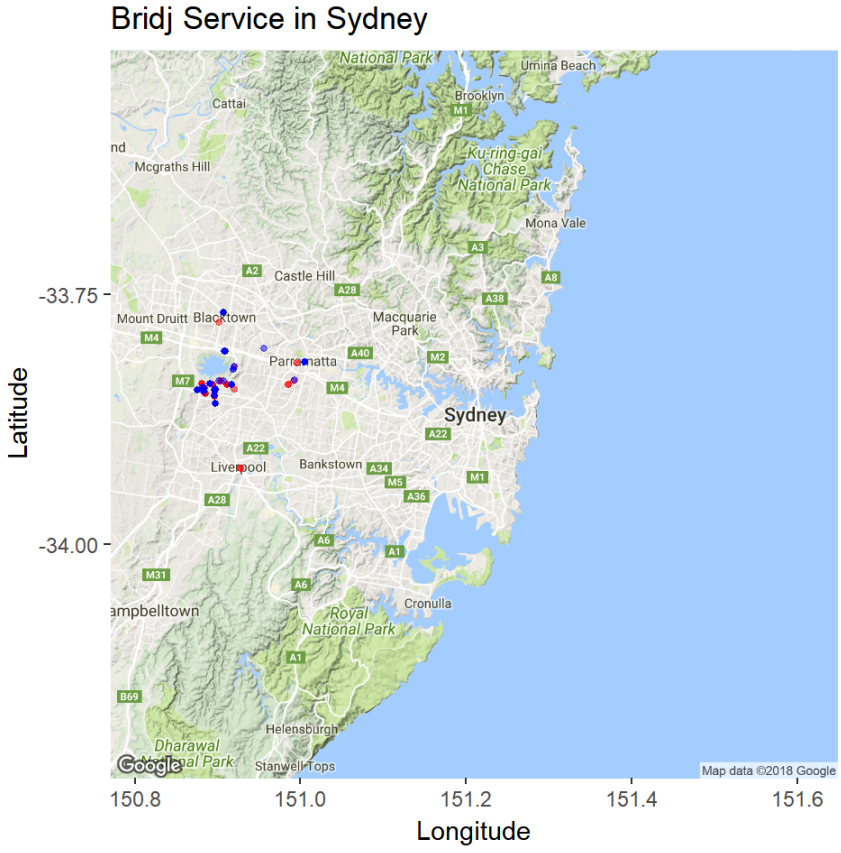


The bar plot shows the passenger distribution for different pickup days in March. The x-axis represents the date whereas y-axis represents the number of the passengers for the particular days. The actual number of the passengers for a corresponding day is printed just above the bar. The fluctuation of the passengers over the time is observed while the overall trend is increasing.

Solution of QN 4

The following visulization presents the pick up and drop off locations of passengers. The x-axis represents the longitudional co-ordinates while y-axis represents the lattitude co-ordinates.Where red color indicates the pickup location while that of blue represents the drop-off.

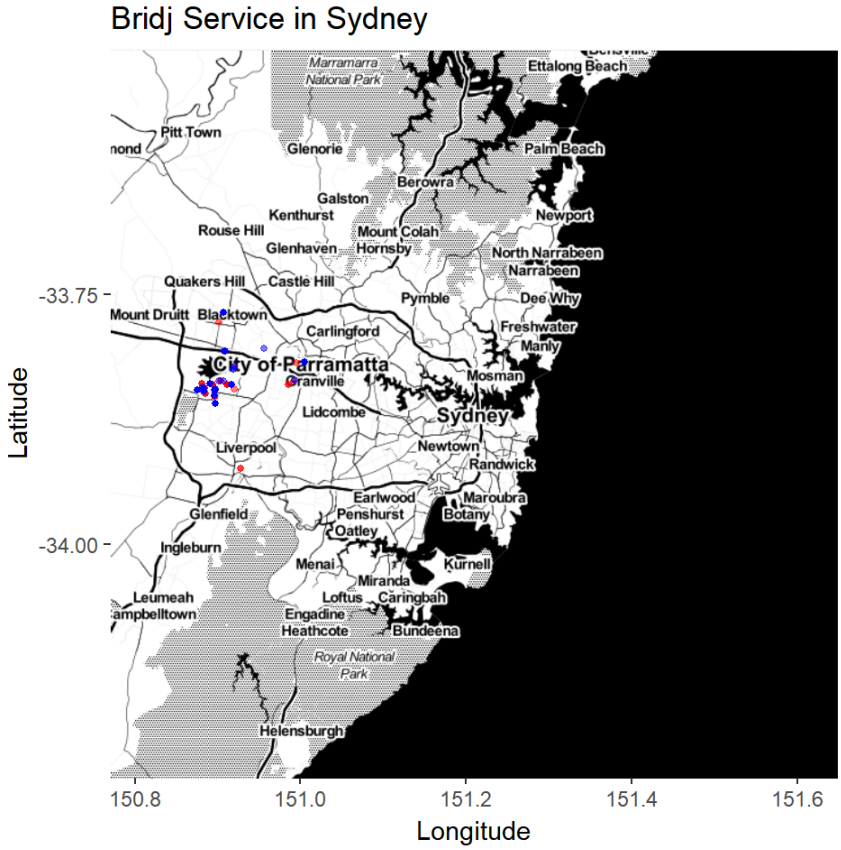
```
Sydney <- get_map("Sydney,Australia", zoom = 10)
# ggmap(Sydney)
p  <-  ggmap(Sydney)
p1 <- p + geom_point(data=bridj_successful, aes(x=PICK_UP_LONGITUDE,    y=PICK_UP_LATITUDE), size=1, colour
 = "red", alpha = .5)
p1 + geom_point(data=bridj_successful, aes(x=DROP_OFF_LONGITUDE , y=DROP_OFF_LATITUDE),
                size=1, colour = "blue", alpha = .5) +
            labs(title = "Bridj Service in Sydney ",
            x = "Longitude", y = "Latitude")
```



```
Syd <- get_map("Sydney, Australia", zoom = 10, source = "stamen", maptype = "toner")
# ggmap(Syd)

pp  <- ggmap(Syd)
pp1 <- pp + geom_point(data=bridj_successful, aes(x=PICK_UP_LONGITUDE, y=PICK_UP_LATITUDE), size=1, colour
 = "red", alpha = .5)

pp1 + geom_point(data=bridj_successful, aes(x=DROP_OFF_LONGITUDE , y=DROP_OFF_LATITUDE),
                size=1, colour = "blue", alpha = .5) +
  labs(title = "Bridj Service in Sydney ",
       x = "Longitude", y = "Latitude")
```



From the above map visualization it is observed that the majority of Bridj customers are located in the City of Parramatta followed by Granville while few are from the Liverpool and Blacktown.

Conclusion

The customer data of Bridj booking has been analyzed using the statistical package R, where we observed that the end of the month has the maximum number of the passenger while at the beginning of the month was recorded as a lower. From the map visualization, we can reveal that the all the customer are located at the periphery of the City of Parramatta.