Assingment\_1\_Answers

Ritesh Ingale

December 24, 2015

This document contains the summary statistic and analysis on Taxi Service Dataset. Loading required libraries

suppressMessages(library(dplyr))  
library(chron)

Step 1 : Acquire the Dataset

tsd <- read.csv("TS\_Dataset.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

Step 2 : Refine the Dataset Correction of spelling mistakes

tsd <- tsd %>% rename(ID = ï..ID)

Selecting the important variable for statistic

tsd <- tsd %>% select(1:3, dtArrive, dtBegin, Sum, ServiceID, 17:20, DistanceTotal, RouteLength)

Filter out the unwanted obeservations

tsd <- tsd %>% filter(CarID != "NULL", dtArrive != "NULL", dtBegin != "NULL", AddrFromGpsY != "NULL", AddrToGpsX != "NULL", RouteLength != "NULL")

Convert dtArrive and dtBegin into HH:mm:ss format

tsd$dtArrive <- gsub(".\\d$", "", tsd$dtArrive)  
tsd$dtBegin <- gsub(".\\d$", "", tsd$dtBegin)

The oders in dataset are captured from 12:00 to 12:59 (1 hour)

tsd$dtArrive <- paste0("12:", tsd$dtArrive)  
tsd$dtBegin <- paste0("12:", tsd$dtBegin)

calculate difference in seconds between dtBegin and dtArrive and store the result in separate variable as "time\_to\_reach"

tsd$dtArrive <- chron(times. = tsd$dtArrive)  
tsd$dtBegin <- chron(times. = tsd$dtBegin)  
tsd$time\_to\_reach <- tsd$dtBegin - tsd$dtArrive   
ch <- times(tsd$time\_to\_reach)  
tsd$time\_to\_reach <- 60 \* 60 \* hours(ch) + minutes(ch) \*60 + seconds(ch)  
tsd$time\_to\_reach <- round(tsd$time\_to\_reach, digits = 0)

Q1 Unique number of clients/users?

unique\_user\_cnt <- n\_distinct(tsd$Client)  
unique\_user\_cnt

## [1] 1469

Q2: Top 10 clients/users? 2.1 Highest revenue Generator

top\_10\_user\_by\_revenue <- tsd %>% group\_by(Client) %>% summarise(Revenue = sum(Sum)) %>% arrange(desc(Revenue)) %>% head(10)  
top\_10\_user\_by\_revenue

## Source: local data frame [10 x 2]  
##   
## Client Revenue  
## (dbl) (int)  
## 1 1.03e+48 20082  
## 2 1.43e+48 19867  
## 3 1.21e+48 19569  
## 4 1.06e+48 18983  
## 5 1.32e+48 18675  
## 6 1.04e+48 18442  
## 7 1.25e+48 18381  
## 8 1.35e+48 17937  
## 9 1.08e+48 17889  
## 10 1.10e+48 17884

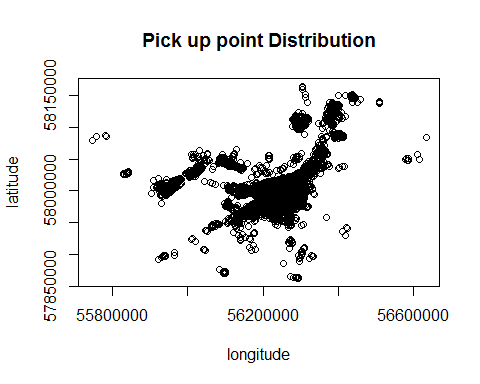
2.2 Regular service consumers

top\_10\_user\_by\_regularity <- tsd %>% group\_by(Client) %>% summarise(freq = n()) %>% arrange(desc(freq)) %>% head(10)  
top\_10\_user\_by\_regularity

## Source: local data frame [10 x 2]  
##   
## Client freq  
## (dbl) (int)  
## 1 1.03e+48 112  
## 2 1.21e+48 110  
## 3 1.43e+48 110  
## 4 1.10e+48 106  
## 5 1.22e+48 105  
## 6 1.06e+48 104  
## 7 1.04e+48 102  
## 8 1.05e+48 101  
## 9 1.07e+48 101  
## 10 1.08e+48 101

Q3: How are starting points distributed? Is there a specific region in which taxi company is seeing very high demand?

plot(x = tsd$AddrFromGpsX, y = tsd$AddrFromGpsY, type = "p", main = "Pick up point Distribution", xlab = "longitude", ylab = "latitude")



top\_region <- tsd %>% group\_by(AddrFromGpsX, AddrFromGpsY) %>% summarise(popularity = n()) %>% arrange(desc(popularity)) %>% head(1)  
top\_region

## Source: local data frame [1 x 3]  
## Groups: AddrFromGpsX [1]  
##   
## AddrFromGpsX AddrFromGpsY popularity  
## (chr) (chr) (int)  
## 1 55745744 58079410 1

Q4: What is the typical cart size/ cost per ride?

cost\_per\_ride <- mean(tsd$Sum)  
cost\_per\_ride

## [1] 179.1818

Q5: Find out the regular customer?

regular\_customer <- tsd %>% group\_by(Client) %>% summarise(rides\_cnt = n()) %>% arrange(desc(rides\_cnt)) %>% head(10)   
regular\_customer

## Source: local data frame [10 x 2]  
##   
## Client rides\_cnt  
## (dbl) (int)  
## 1 1.03e+48 112  
## 2 1.21e+48 110  
## 3 1.43e+48 110  
## 4 1.10e+48 106  
## 5 1.22e+48 105  
## 6 1.06e+48 104  
## 7 1.04e+48 102  
## 8 1.05e+48 101  
## 9 1.07e+48 101  
## 10 1.08e+48 101

Q6: Which distance regular customers travels? (large distance or short distance)

tsd\_without\_zero\_dist <- tsd %>% filter(DistanceTotal != 0)  
regular\_customer\_type <- tsd\_without\_zero\_dist %>% group\_by(Client) %>% summarise\_each(funs(mean, median), DistanceTotal)   
regular\_customer\_type <- dplyr::left\_join(regular\_customer, regular\_customer\_type, by = "Client")  
regular\_customer\_type

## Source: local data frame [10 x 4]  
##   
## Client rides\_cnt mean median  
## (dbl) (int) (dbl) (dbl)  
## 1 1.03e+48 112 12.045455 5  
## 2 1.21e+48 110 7.705882 5  
## 3 1.43e+48 110 7.647887 6  
## 4 1.10e+48 106 5.087719 5  
## 5 1.22e+48 105 4.854839 3  
## 6 1.06e+48 104 8.557143 4  
## 7 1.04e+48 102 6.880597 5  
## 8 1.05e+48 101 7.158730 4  
## 9 1.07e+48 101 6.781818 5  
## 10 1.08e+48 101 9.582090 4

Q7: Find most crowded pick up points?

Top\_pickups\_points <- tsd %>% group\_by(AddrFromGpsX, AddrFromGpsY) %>% summarise(customer\_cnt = n()) %>% arrange(desc(customer\_cnt)) %>% head(5)  
Top\_pickups\_points

## Source: local data frame [5 x 3]  
## Groups: AddrFromGpsX [5]  
##   
## AddrFromGpsX AddrFromGpsY customer\_cnt  
## (chr) (chr) (int)  
## 1 55745744 58079410 1  
## 2 55758071 58085994 1  
## 3 55782240 58086743 1  
## 4 55783360 58085509 1  
## 5 55827972 58027319 1

Q8: What is the average cost paid by any customer?

avg\_ride\_cost\_customer <- tsd %>% group\_by(Client) %>% summarise(typical\_cost = mean(Sum)) %>% arrange(desc(typical\_cost)) %>% head(10)  
avg\_ride\_cost\_customer

## Source: local data frame [10 x 2]  
##   
## Client typical\_cost  
## (dbl) (dbl)  
## 1 1.32e+45 940.0000  
## 2 3.91e+46 500.0000  
## 3 2.79e+46 470.0000  
## 4 8.94e+46 470.0000  
## 5 3.34e+45 410.0000  
## 6 3.33e+46 410.0000  
## 7 3.70e+46 408.0000  
## 8 4.67e+47 405.0000  
## 9 9.93e+45 400.0000  
## 10 1.78e+47 393.3333

Q9: Are regular customers tends to choose specific car if yes then what average distance they travels using that car?

regular\_customer\_details <- dplyr::left\_join(regular\_customer, tsd, by = "Client") %>% select(Client, CarID, DistanceTotal)  
regular\_customer\_car <- regular\_customer\_details %>%   
 group\_by(Client, CarID) %>%   
 summarise(freq\_car\_used = n()) %>%   
 arrange(desc(freq\_car\_used)) %>%   
 top\_n(1) %>%   
 filter(freq\_car\_used > 2) %>%   
 dplyr::left\_join(., regular\_customer\_details, by = c("Client", "CarID")) %>%   
 select(Client, CarID, DistanceTotal) %>%  
 group\_by(Client, CarID) %>%   
 summarise(avg\_dist = mean(DistanceTotal)) %>%   
 arrange(desc(avg\_dist))

## Selecting by freq\_car\_used

regular\_customer\_car

## Source: local data frame [11 x 3]  
## Groups: Client [7]  
##   
## Client CarID avg\_dist  
## (dbl) (chr) (dbl)  
## 1 1.04e+48 26081 9.333333  
## 2 1.05e+48 38375 8.000000  
## 3 1.06e+48 1746 2.666667  
## 4 1.06e+48 2126 1.000000  
## 5 1.06e+48 61100 0.000000  
## 6 1.07e+48 38197 8.500000  
## 7 1.08e+48 61549 6.333333  
## 8 1.21e+48 1222 5.000000  
## 9 1.21e+48 38392 3.666667  
## 10 1.43e+48 13530 2.333333  
## 11 1.43e+48 42511 1.000000

Q10: Which type of car is more popular among the customers?

car\_popularity <- tsd %>% group\_by(CarID) %>% summarise(Total\_Revenue = sum(Sum)) %>% arrange(desc(Total\_Revenue)) %>% head(10)  
car\_popularity

## Source: local data frame [10 x 2]  
##   
## CarID Total\_Revenue  
## (chr) (int)  
## 1 38168 16701  
## 2 27919 16414  
## 3 24643 16271  
## 4 38350 15845  
## 5 1125 15386  
## 6 27069 14682  
## 7 38296 14439  
## 8 38197 14061  
## 9 43000 13952  
## 10 24581 13898

Q11: What is the average time car driver takes to reach at suggested pickup points after customer placed the order?

avg\_time\_by\_driver <- tsd %>% group\_by(CarID) %>% summarise(avg\_time = mean(time\_to\_reach)) %>% arrange(avg\_time) %>% head(10)

Q12: From which pick up points company gets customer who travels large distance?

large\_dist\_points <- tsd %>% group\_by(AddrFromGpsX, AddrFromGpsY) %>% summarise(large\_dist = max(DistanceTotal))  
large\_dist\_points <- large\_dist\_points[order(large\_dist\_points$large\_dist, decreasing = TRUE),]  
head(large\_dist\_points, 10)

## Source: local data frame [10 x 3]  
## Groups: AddrFromGpsX [10]  
##   
## AddrFromGpsX AddrFromGpsY large\_dist  
## (chr) (chr) (int)  
## 1 56389329 58125817 384  
## 2 56168371 57962325 329  
## 3 56190663 57984646 288  
## 4 56295001 58021289 266  
## 5 56197572 57965839 265  
## 6 56176802 57975821 250  
## 7 56288600 58111786 249  
## 8 56299979 57981554 246  
## 9 56305364 58025150 244  
## 10 56210925 57976645 215