Cyclist_Data_Analysis

Yash

2023-12-20

Cyclist Data Analysis

Data Preparation and Cleaning

```
# Load packages and set working directory, use install.packages if not installed

library(tidyverse);
library(lubridate);
library(ggplot2);
library(readr);
library(dplyr);
library(anytime);
#setwd("C:/Users/yashs/OneDrive/Documents/Datasets/Cyclist_Case_Study/csvs");
```

```
# Read data

q1_q2_2014 <- read_csv("./csvs/Divvy_Trips_2014_Q1Q2.csv");
q1_2015 <- read_csv("./csvs/Divvy_Trips_2015-Q1.csv");
q2_2015 <- read_csv("./csvs/Divvy_Trips_2015-Q2.csv");
q4_2015 <- read_csv("./csvs/Divvy_Trips_2015_Q4.csv");
q3_2016 <- read_csv("./csvs/Divvy_Trips_2016_Q3.csv");
q4_2016 <- read_csv("./csvs/Divvy_Trips_2016_Q4.csv");
q1_2017 <- read_csv("./csvs/Divvy_Trips_2017_Q1.csv");
q2_2017 <- read_csv("./csvs/Divvy_Trips_2017_Q2.csv");
q3_2017 <- read_csv("./csvs/Divvy_Trips_2017_Q3.csv");
q4_2017 <- read_csv("./csvs/Divvy_Trips_2017_Q4.csv");
q1_2018 <- read_csv("./csvs/Divvy_Trips_2018_Q1.csv");
q2_2018 <- read_csv("./csvs/Divvy_Trips_2018_Q2.csv");
q3_2018 <- read_csv("./csvs/Divvy_Trips_2018_Q3.csv");
q4_2018 <- read_csv("./csvs/Divvy_Trips_2018_Q4.csv");
q1_2019 <- read_csv("./csvs/Divvy_Trips_2019_Q1.csv");
q2_2019 <- read_csv("./csvs/Divvy_Trips_2019_Q2.csv");
q3_2019 <- read_csv("./csvs/Divvy_Trips_2019_Q3.csv");
q4_2019 <- read_csv("./csvs/Divvy_Trips_2019_Q4.csv");
q1_2020 <- read_csv("./csvs/Divvy_Trips_2020_Q1.csv");
```

```
# rename columns so they match for data merging

q1_2018 <- rename(q1_2018
                   ,trip_id = "01 - Rental Details Rental ID"
                   ,start_time = "01 - Rental Details Local Start Time"
                   ,end_time = "01 - Rental Details Local End Time"
                   ,bikeid = "01 - Rental Details Bike ID"
                   ,tripduration="01 - Rental Details Duration In Seconds Uncapped"
                   ,from_station_id = "03 - Rental Start Station ID"
                   ,from_station_name = "03 - Rental Start Station Name"
                   ,to_station_id = "02 - Rental End Station ID"
                   ,to_station_name = "02 - Rental End Station Name"
                   ,usertype = "User Type"
                   ,gender="Member Gender"
                   ,birthyear="05 - Member Details Member Birthday Year");
q2_2019 <- rename(q2_2019
                  ,trip_id = "01 - Rental Details Rental ID"
                  ,start_time = "01 - Rental Details Local Start Time"
                  ,end_time = "01 - Rental Details Local End Time"
                  ,bikeid = "01 - Rental Details Bike ID"
                  ,tripduration="01 - Rental Details Duration In Seconds Uncapped"
                  ,from_station_id = "03 - Rental Start Station ID"
                  ,from_station_name = "03 - Rental Start Station Name"
                  ,to_station_id = "02 - Rental End Station ID"
                  ,to_station_name = "02 - Rental End Station Name"
                  ,usertype = "User Type"
                  ,gender="Member Gender"
                  ,birthyear="05 - Member Details Member Birthday Year");
q1_2020 <- rename(q1_2020
                  ,trip_id = "ride_id"
                  ,start_time = "started_at"
                  ,end_time = "ended_at"
                  ,from_station_id = "start_station_id"
                  ,from_station_name = "start_station_name"
                  ,to_station_id = "end_station_id"
                  ,to_station_name = "end_station_name"
                  ,usertype = "member_casual");
q1_q2_2014 <- rename(q1_q2_2014
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
q1_2015 <- rename(q1_2015
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
q2_2015 <- rename(q2_2015
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
q4_2015 <- rename(q4_2015
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
q3_2016 <- rename(q3_2016
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
q4_2016 <- rename(q4_2016
                  ,start_time = "starttime"
                  ,end_time = "stoptime");
```

```
# add missing columns to 2020 data for merging

q1_2020$tripduration <- NA;
q1_2020$bikeid <- NA;
q1_2020$gender <- NA;
q1_2020$birthyear <- NA;
q1_2020_modified <- select(q1_2020, -start_lat, -start_lng, -end_lat, -end_lng, -rideable_type);
```

```
# merge data

full_table <- rbind(q1_q2_2014,
                    q1_2015,
                    q2_2015,
                    q4_2015,
                    q3_2016,
                    q4_2016,
                    q1_2017,
                    q2_2017,
                    q3_2017,
                    q4_2017,
                    q1_2018,
                    q2_2018,
                    q3_2018,
                    q4_2018,
                    q1_2019,
                    q2_2019,
                    q3_2019,
                    q4_2019,
                    q1_2020_modified);
```

```
# check for duplicate trip ids
ids <- data.frame(table(full_table$trip_id));
duplicate_ids <- ids[which(ids$Freq > 1),];
print(nrow(duplicate_ids) == 0); # prints FALSE, meaning there are duplicates
```

```
# get records with duplicate ids
duplicate_trips <- full_table[full_table$trip_id %in% duplicate_ids$Var1,]
# 120 records. There should be one record for each duplicate id, meaning 60 should be deleted
```

```
# number of rows should only go down by 60
num_rows_before_duplicate_clean <- nrow(full_table); # 16435933
full_table_duplicate_removed <- distinct(full_table, full_table$trip_id, .keep_all=TRUE);
num_rows_after_duplicate_clean <-  nrow(full_table_duplicate_removed); #16435873 = 16435933-60
```

```
# store copy of clean table
full_table_clean <- full_table_duplicate_removed
```

```
# clean usertype column so it only contains values for Subscriber and Customer
# see all different user type labels
usertypes <- table(full_table_clean$usertype); # casual, Customer, member, Subscriber, dependent
#remove dependent (not relevant to business task, 121 rows)
full_table_clean <- full_table_clean[which(full_table_clean$usertype !="Dependent"),];
```

```
num_rows <- nrow(full_table_clean); #16435752 = 16435873 - 121

# change casual to Customer and member to Subscriber
full_table_clean <- full_table_clean %>%
    mutate(usertype = recode(usertype,
      "casual" = "Customer",
      "member" = "Subscriber"
    ));

# verify that only Customers and Subscribers remain
clean_usertypes <- table(full_table_clean$usertype);
```

Dates

```
#create copy for testing
full_table_date <- full_table_clean;

# parse date column into positCX object so we can easily extract month, day, year
full_table_date$start_time_date <- parse_date_time(full_table_date$start_time, orders = c("%m/%d/%Y %H:
full_table_date$end_time_date <- parse_date_time(full_table_date$end_time, orders = c("%m/%d/%Y %H:%M:%

# order by start_time
full_table_date <- full_table_date[order(full_table_date$start_time_date),];

#remove records where date failed to parse
full_table_good_dates <-  full_table_date;
full_table_good_dates <- drop_na(full_table_good_dates, start_time_date, end_time_date);
```

```
# recalculate durations

full_table_good_dates$tripduration <- with(full_table_good_dates, as.numeric(difftime(end_time_date, st

#remove trips where duration <0
full_table_good_dates <- full_table_good_dates[which(full_table_good_dates$tripduration > 0),]

# extract month, day, weekday, year from date and add columns for each field
full_table_good_dates$date <- as.Date(full_table_good_dates$start_time_date)
full_table_good_dates$month <- format(as.Date(full_table_good_dates$date), "%m")
full_table_good_dates$day <- format(as.Date(full_table_good_dates$date), "%d")
full_table_good_dates$year <- format(as.Date(full_table_good_dates$date), "%Y")
full_table_good_dates$day_of_week <- format(as.Date(full_table_good_dates$date), "%A")
```

```
# remove entries from particular station per case study spec
full_table_good_dates <- full_table_good_dates[(full_table_good_dates$from_station_name != "HQ QR"),];
```

Analysis

```
# get summary statistics of trip duration
tripduration_summary <- summary(full_table_good_dates$tripduration);
```

```r
# create logical order for day of week column
full_table_good_dates$day_of_week <- ordered(full_table_good_dates$day_of_week, levels=c("Sunday", "Mon
```

```r
#get average trip duration by usertype
mean_length_by_member <- aggregate(full_table_good_dates$tripduration ~ full_table_good_dates$usertype,
colnames(mean_length_by_member) <- c("usertype", "mean_trip_duration");
```

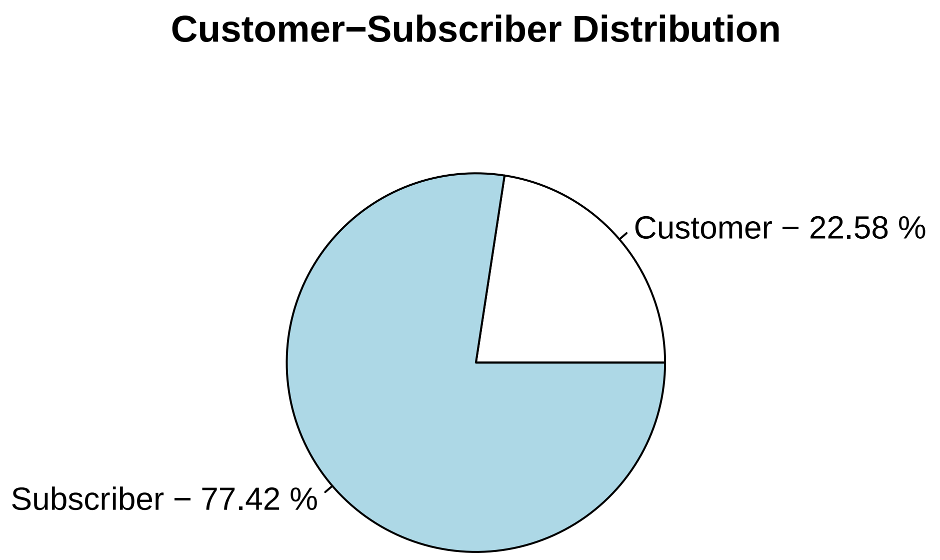
```r
#get mean lengths by member type and day
mean_length_by_member_day <- aggregate(full_table_good_dates$tripduration ~ full_table_good_dates$usert
colnames(mean_length_by_member_day) <- c("usertype", "day_of_week", "mean_trip_duration");
```

```r
# group by usertype and weekday, and generate average trip duration for each group
full_table_groups <- full_table_good_dates;
full_table_groups = full_table_groups %>% group_by(usertype, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration=mean(tripduration)) %>%
  arrange(usertype, day_of_week);
```

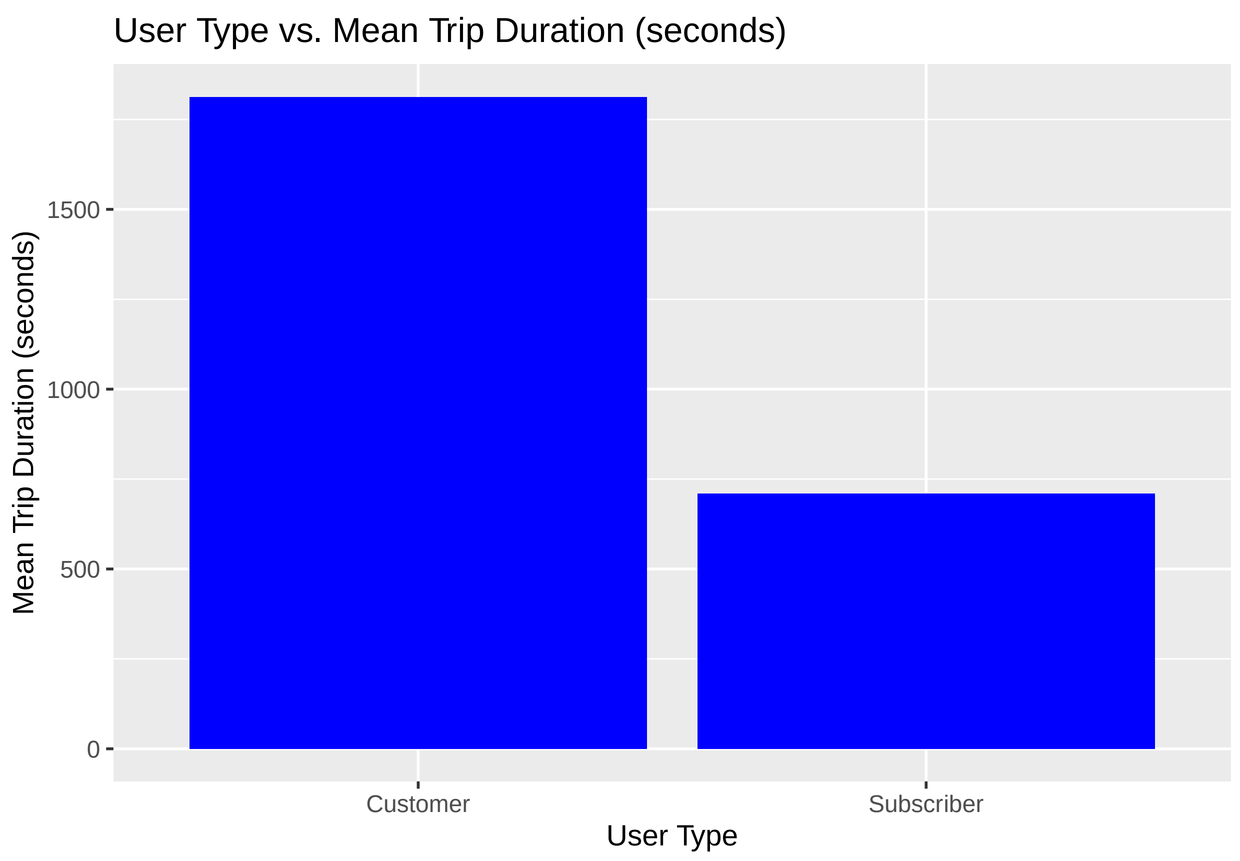
Plot Data

```r
# Organize data by subscribers and regular customers, and label data for plotting
num_usertypes <- full_table_clean %>% group_by(usertype) %>%
  summarize(num_users = n());
num_usertypes = num_usertypes %>% mutate(user_percentage=round((num_usertypes$num_users/sum(num_usertyp
lbls <- paste(num_usertypes$usertype, num_usertypes$user_percentage, sep=" - ");
lbls <- paste(lbls, "%", sep=" ");
```

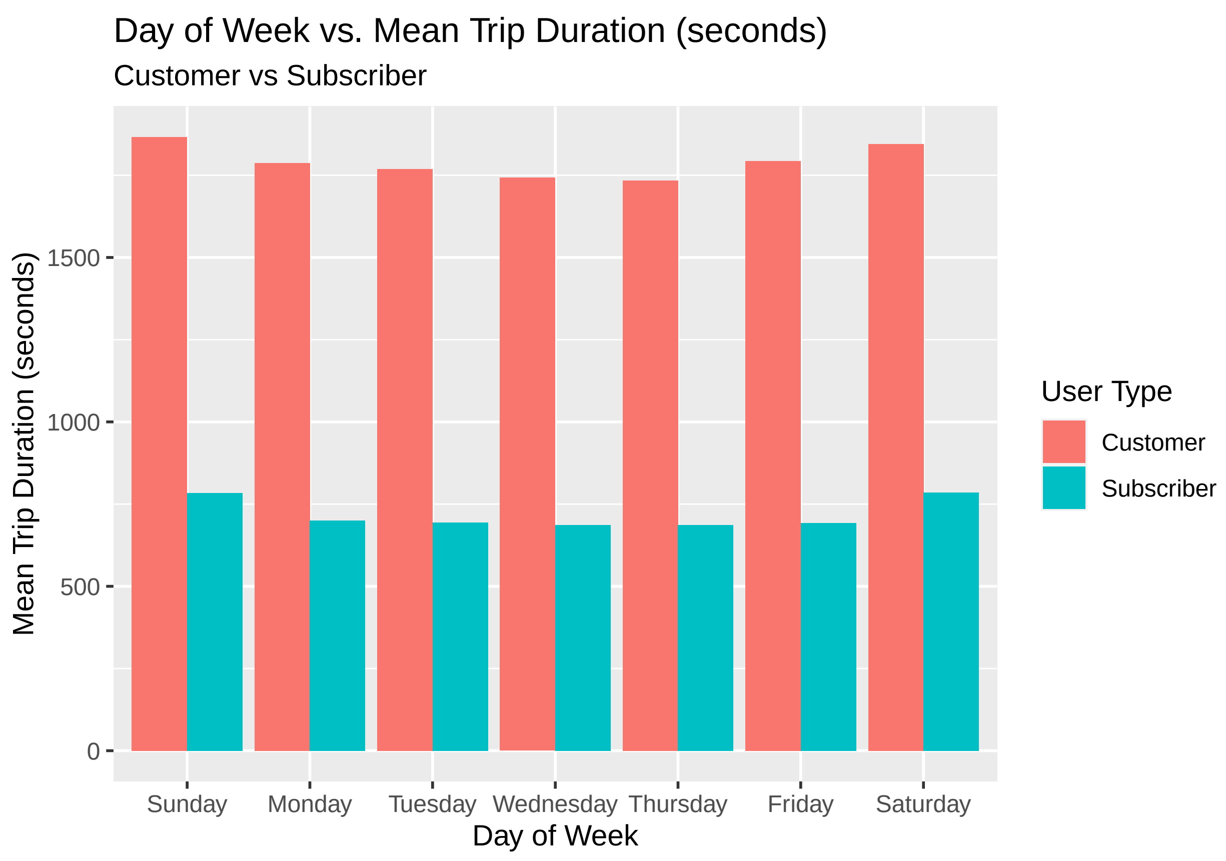
```r
pie(num_usertypes$user_percentage, labels=lbls, main="Customer-Subscriber Distribution");
```



```
ggplot(mean_length_by_member) + geom_col(aes(x=usertype,y=mean_trip_duration), fill="blue") +
  ggtitle("User Type vs. Mean Trip Duration (seconds)") +
  labs(x="User Type", y="Mean Trip Duration (seconds)");
```



```r
ggplot(mean_length_by_member_day) + geom_col(aes(x=day_of_week, y=mean_trip_duration, fill=usertype), p
  ggtitle("Day of Week vs. Mean Trip Duration (seconds)", subtitle="Customer vs Subscriber") +
  labs(x="Day of Week", y="Mean Trip Duration (seconds)", fill="User Type");
```


Day of Week vs. Mean Trip Duration (seconds)
Customer vs Subscriber
Mean Trip Duration (seconds)
0
500
1000
1500
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of Week
User Type
Customer
Subscriber

“‘