Cyclistic Case study

Assem M. A. Fadl

5/10/2021

Setting Up my Environment

Installing Packages

- install.packages(“tidyverse”)
- install.packages(“lubridate”)
- install.packages(“ggplot2”)

Loading Packages

- library(tidyverse)
- library(lubridate)
- library(ggplot2)

PREPARING DATA

Uploading Csv files and Assigning Names to them

- q2_2019 <- read_csv(“Divvy_Trips_2019_Q2.csv”)
- q3_2019 <- read_csv(“Divvy_Trips_2019_Q3.csv”)
- q4_2019 <- read_csv(“Divvy_Trips_2019_Q4.csv”)
- q1_2020 <- read_csv(“Divvy_Trips_2020_Q1.csv”)

PROCESSING DATA

COMBINING DATA INTO A SINGLE FILE

Comparing column names for each of the files

- colnames(q2_2019)
- colnames(q3_2019)
- colnames(q4_2019)
- colnames(q1_2020)

Renaming columns to make them consisent with q1_2020

- (q4_2019 <- rename(q4_2019, ride_id=trip_id, rideable_type=bikeid, started_at=start_time, ended_at=end_time, start_station_name=from_station_name, start_station_id=from_station_id, end_station_name=to_station_name, end_station_id=to_station_id, member_casual=usertype))

- (q3_2019 <- rename(q3_2019, ride_id=trip_id, rideable_type=bikeid, started_at=start_time, ended_at=end_time, start_station_name=from_station_name, start_station_id=from_station_id, end_station_name=to_station_name, end_station_id=to_station_id, member_casual=usertype))

- (q2_2019 <- rename(q2_2019, ride_id="01 - Rental Details Rental ID", rideable_type="01 - Rental Details Bike ID", started_at="01 - Rental Details Local Start Time", ended_at="01 - Rental Details Local End Time", start_station_name="03 - Rental Start Station Name", start_station_id="03 - Rental Start Station ID", end_station_name="02 - Rental End Station Name", end_station_id="02 - Rental End Station ID", member_casual="User Type"))

Inspecting the data frames

- str(q1_2020)
- str(q4_2019)
- str(q3_2019)
- str(q2_2019)

Aligning Data types together correctly

- q4_2019 <- mutate(q4_2019, ride_id = as.character(ride_id) ,rideable_type = as.character(rideable_type))
- q3_2019 <- mutate(q3_2019, ride_id = as.character(ride_id) ,rideable_type = as.character(rideable_type))
- q2_2019 <- mutate(q2_2019, ride_id = as.character(ride_id) ,rideable_type = as.character(rideable_type))

Stack individual quarter's data frames into one big data frame

```
all_trips <- bind_rows(q2_2019, q3_2019, q4_2019, q1_2020)
```

Remove lat, long, birthyear, and gender fields as this data was dropped beginning in 2020

```
all_trips <- all_trips %>%
select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, "01 - Rental Details Duration In Seconds Uncapped", "05 - Member Details Member Birthday Year", "Member Gender", "tripduration"))
```

Inspecting the new table that has been created

- colnames(all_trips)
- nrow(all_trips)
- dim(all_trips)

- head(all_trips)
- str(all_trips)
- summary(all_trips)

There are a few problems we will need to fix:

(1) In the “member_casual” column, there are two names for members (“member” and “Subscriber”) and two names for casual riders (“Customer” and “casual”). We will need to consolidate that from four to two labels.

(2) The data can only be aggregated at the ride-level, which is too granular. We will want to add some additional columns of data – such as day, month, year – that provide additional opportunities to aggregate the data.

(3) We will want to add a calculated field for length of ride since the 2020Q1 data did not have the “tripduration” column. We will add “ride_length” to the entire dataframe for consistency.

(4) There are some rides where tripduration shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. We will want to delete these rides.

Reassign to the desired values (we will go with the current 2020 labels)

- all_trips <- all_trips %>% mutate(member_casual = recode(member_casual ,“Subscriber” = “member” ,“Customer” = “casual”))

Adding columns that list the date, month, day, and year of each ride

- all_trips$date < -as.Date(all_{t}rips$started_at) #The default format is yyyy-mm-dd
- all_trips$month < -format(as.Date(all_{t}rips$date), “%m”)
- all_trips$day < -format(as.Date(all_{t}rips$date), “%d”)
- all_trips$year < -format(as.Date(all_{t}rips$date), “%Y”)
- all_trips$day_{o}f_{w}eek < -format(as.Date(all_{t}rips$date), “%A”)

Adding a “ride_length” calculation to all_trips (in seconds)

all_trips$ride_{l}ength < -difftime(all_{t}rips$ended_at,all_trips$started_at)

Converting “ride_length” from Factor to numeric so we can run calculations on the data

is.factor(all_trips$ride_{l}ength)all_{t}rips$ride_length <-
as.numeric(as.character(all_trips$ride_{l}ength))is.numeric(all_{t}rips$ride_length)

Remove “bad” data & creating a new version of the dataframe (v2)

all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" |
all_trips$ride_length<0),]

CONDUCT DESCRIPTIVE ANALYSIS

Descriptive analysis on ride_length (all figures in seconds)

summary(all_trips_v2$ride_length)

Compare members and casual users

- aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}2$member_casual, FUN = mean)

- aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}$2member_casual, FUN = median)
- aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}$2member_casual, FUN = max)
- aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}$2member_casual, FUN = min)

See the average ride time by each day for members vs casual users

aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}$2member_casual + all_trips_v2$day_of_week, FUN = mean)

Ordering the days of the week

all_trips_v2$day_{o}f_{w}eek < -ordered(all_{t}rips_{v}$2day_of_week, levels=c(“Sunday”, “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”))

Running the average ride time by each day for members vs casual users

aggregate(all_trips_v2$ride_{l}ength\ all_{t}rips_{v}$2member_casual + all_trips_v2$day_of_week, FUN = mean)

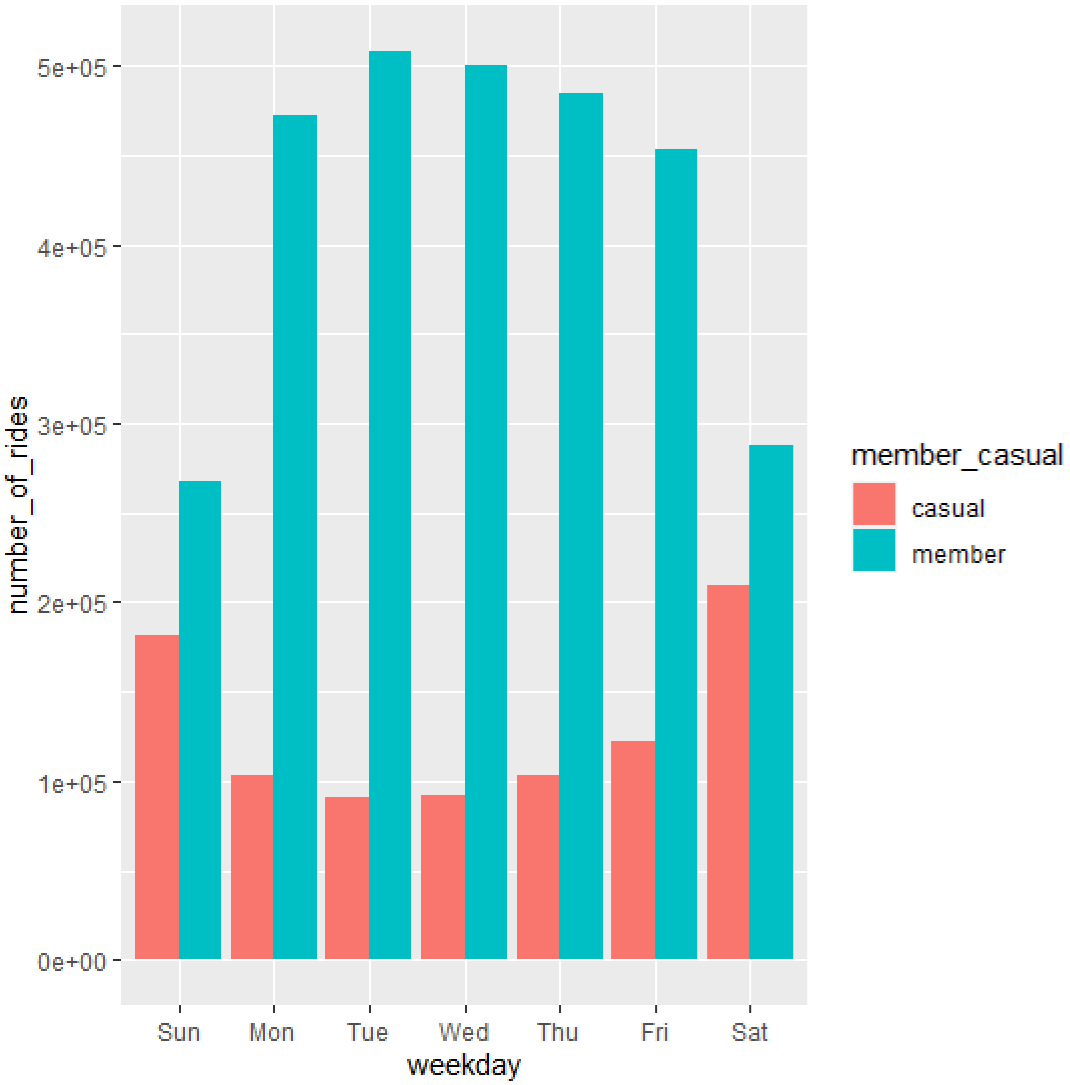
Analyzing ridership data by type and weekday

all_trips_v2 %>% mutate(weekday = wday(started_at, label = TRUE)) %>%
group_by(member_casual, weekday) %>%
summarise(number_of_rides = n() ,average_duration = mean(ride_length))

Visualization

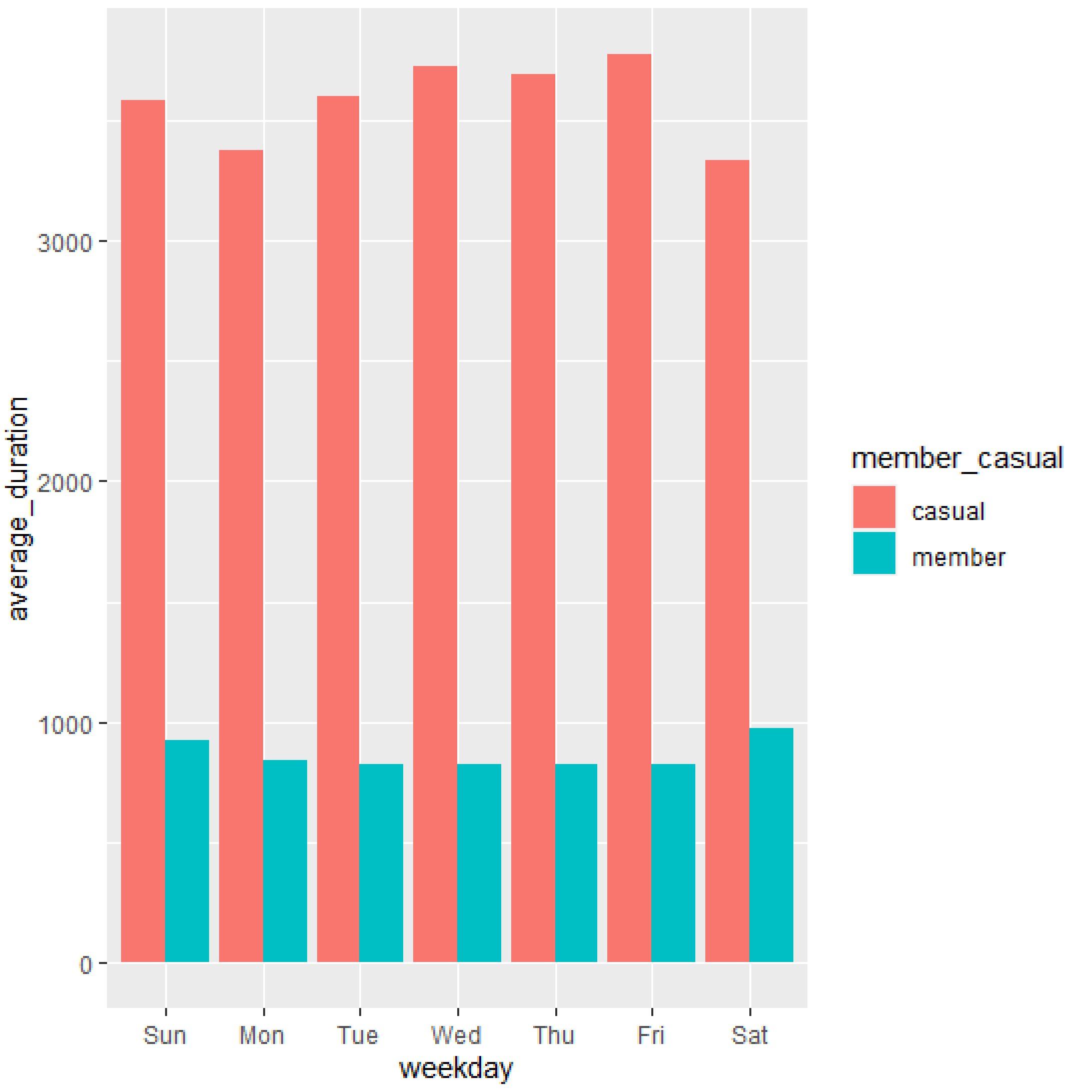
Visualize the number of rides by rider type

all_trips_v2 %>% mutate(weekday = wday(started_at, label = TRUE)) %>% group_by(member_casual, weekday) %>% summarise(number_of_rides = n() ,average_duration = mean(ride_length)) %>% arrange(member_casual, weekday) %>% ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) + geom_col(position = “dodge”)



A visualization for average duration

```
all_trips_v2 %>% mutate(weekday = wday(started_at, label = TRUE)) %>%
group_by(member_casual, weekday) %>% summarise(number_of_rides = n()
,average_duration = mean(ride_length)) %>% arrange(member_casual, weekday) %>%
ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) + geom_col(position
= “dodge”)
```


average_duration
3000
2000
1000
0
Sun
Mon
Tue
Wed
Thu
Fri
Sat
weekday
member_casual
casual
member