Divvy Trips Exercise

2022-12-29

## Divvy Trips Project

This analysis helps to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, I am going to [download](https://divvy-tripdata.s3.amazonaws.com/index.html), clean and prepare data for analysis.

# Download libraries and collect data

library(tidyverse)  
library(lubridate)  
library(ggplot2) library(dplyr) 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”)

# Prepare data to analysis

colnames(q3\_2019) colnames(q4\_2019) colnames(q2\_2019) colnames(q1\_2020)

# Rename columns to make them consistent

(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”))

# Inspect the dataframes and look for incongruencies

str(q1\_2020) str(q4\_2019) str(q3\_2019) str(q2\_2019)

# Convert ride\_id and rideable\_type to character so that they can stack 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

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”))

# Clean up and add data to prepare for analysis

colnames(all\_trips)  
nrow(all\_trips)  
dim(all\_trips)  
head(all\_trips)  
str(all\_trips)  
summary(all\_trips)

table(all\_trips$member\_casual)

all\_trips <- all\_trips %>% mutate(member\_casual = recode(member\_casual ,“Subscriber” = “member” ,“Customer” = “casual”))

table(all\_trips$member\_casual)

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

all\_tripsstarted\_at) all\_tripsdate), “%m”) all\_tripsdate), “%d”) all\_tripsdate), “%Y”) all\_tripsdate), “%A”) all\_tripsdate, label=TRUE) table(all\_trips$day\_of\_week)

all\_tripsended\_at,all\_trips$started\_at)

str(all\_trips)

is.factor(all\_tripsride\_length <- as.numeric(as.character(all\_tripsride\_length)

# Remove “bad” data

all\_trips\_v2 <- all\_trips[!(all\_trips$start\_station\_name == "HQ QR" | all\_trips$ride\_length<0),] all\_trips\_v2date), “%A”) all\_trips\_v2$day\_of\_week

# Conduct descriptive analysis

mean(all\_trips\_v2ride\_length) max(all\_trips\_v2ride\_length)

summary(all\_trips\_v2$ride\_length)

# Compare members and casual users

aggregate(all\_trips\_v2member\_casual, FUN = mean) aggregate(all\_trips\_v2member\_casual, FUN = median) aggregate(all\_trips\_v2member\_casual, FUN = max) aggregate(all\_trips\_v2member\_casual, FUN = min)

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

all\_trips\_v2day\_of\_week, levels=c(“воскресенье”, “понедельник”, “вторник”, “среда”, “четверг”, “пятница”, “суббота”))

# Now, let’s run the average ride time by each day for members vs casual users

aggregate(all\_trips\_v2member\_casual + all\_trips\_v2$day\_of\_week, FUN = mean)

# analyze 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)) %>%  
arrange(member\_casual, weekday)

# Let’s 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”)

# Let’s create 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”)

# Export summary file for further analysis

counts <- aggregate(all\_trips\_v2member\_casual + all\_trips\_v2$day\_of\_week, FUN = mean) write.csv(counts, file = “clean data.csv”)