Cyclistic Bike-Share Case Study

Oluwaseun Otoki

2022-05-08

The following analysis is based on the Google Data Analytics Capstone Case Study “How Does a Bike-Share Navigate Speedy Success?”

## Background

Cyclistic is bike-sharing company with a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Cyclistic has 3 flexible pricing options: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are casual riders. Customers who purchase annual memberships are Cyclistic members.

## Scenario

You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company’s future success depends on maximizing the number of annual memberships. Therefore, your team wants 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, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

### Business Task

How do annual members and casual riders use Cyclistic bikes differently?

## Preperation

I am using Cyclistic’s historical trip [dataset](https://divvy-tripdata.s3.amazonaws.com/index.html) to analyze 12 months (April 2021-March 2022), prepared by [Motivate International Inc](https://ride.divvybikes.com/data-license-agreement). R will be used to for data cleaning and visualizations.

**Step 1: Install packages**

library(tidyverse)  
library(ggplot2)  
library(lubridate)  
library(dplyr)  
library(janitor)  
library(skimr)

**Step 2: Import data sets**

trips\_2104 <- read.csv('202104-divvy-tripdata.csv')  
trips\_2105 <- read.csv('202105-divvy-tripdata.csv')  
trips\_2106 <- read.csv('202106-divvy-tripdata.csv')  
trips\_2107 <- read.csv('202107-divvy-tripdata.csv')  
trips\_2108 <- read.csv('202108-divvy-tripdata.csv')  
trips\_2109 <- read.csv('202109-divvy-tripdata.csv')  
trips\_2110 <- read.csv('202110-divvy-tripdata.csv')  
trips\_2111 <- read.csv('202111-divvy-tripdata.csv')  
trips\_2112 <- read.csv('202112-divvy-tripdata.csv')  
trips\_2201 <- read.csv('202201-divvy-tripdata.csv')  
trips\_2202 <- read.csv('202202-divvy-tripdata.csv')  
trips\_2203 <- read.csv('202203-divvy-tripdata.csv')

Compare column names of each file. **Columns must be in the same order for data merging**. Result “0 rows” means there are no discrepancies.

compare\_df\_cols(trips\_2104, trips\_2105, trips\_2106, trips\_2107, trips\_2108, trips\_2109, trips\_2110, trips\_2111, trips\_2112, trips\_2201, trips\_2202, trips\_2203, return = "mismatch")

Inspect the data frames for any inconsistencies

str(trips\_2104)

## 'data.frame': 337230 obs. of 13 variables:  
## $ ride\_id : chr "6C992BD37A98A63F" "1E0145613A209000" "E498E15508A80BAD" "1887262AD101C604" ...  
## $ rideable\_type : chr "classic\_bike" "docked\_bike" "docked\_bike" "classic\_bike" ...  
## $ started\_at : chr "2021-04-12 18:25:36" "2021-04-27 17:27:11" "2021-04-03 12:42:45" "2021-04-17 09:17:42" ...  
## $ ended\_at : chr "2021-04-12 18:56:55" "2021-04-27 18:31:29" "2021-04-07 11:40:24" "2021-04-17 09:42:48" ...  
## $ start\_station\_name: chr "State St & Pearson St" "Dorchester Ave & 49th St" "Loomis Blvd & 84th St" "Honore St & Division St" ...  
## $ start\_station\_id : chr "TA1307000061" "KA1503000069" "20121" "TA1305000034" ...  
## $ end\_station\_name : chr "Southport Ave & Waveland Ave" "Dorchester Ave & 49th St" "Loomis Blvd & 84th St" "Southport Ave & Waveland Ave" ...  
## $ end\_station\_id : chr "13235" "KA1503000069" "20121" "13235" ...  
## $ start\_lat : num 41.9 41.8 41.7 41.9 41.7 ...  
## $ start\_lng : num -87.6 -87.6 -87.7 -87.7 -87.7 ...  
## $ end\_lat : num 41.9 41.8 41.7 41.9 41.7 ...  
## $ end\_lng : num -87.7 -87.6 -87.7 -87.7 -87.7 ...  
## $ member\_casual : chr "member" "casual" "casual" "member" ...

**Step 3: Data Merging**

Due to the consistency in the data, we are now able to merge our data into a single DataFrame.

city\_trips <- bind\_rows(trips\_2104, trips\_2105, trips\_2106, trips\_2107, trips\_2108, trips\_2109, trips\_2110, trips\_2111, trips\_2112, trips\_2201, trips\_2202, trips\_2203)

Drop the following columns: start\_lat, start\_lng, emd\_lat, end\_lng. Data contains station locations already. Coordinates are not needed.

city\_trips <- city\_trips %>%   
 select(-c(start\_lat, start\_lng, end\_lat, end\_lng))

**Step 4: Adding new variables / Preparing for analysis**

1. Add the date, month, year, day, and the day of the week.

city\_trips$date <- as.Date(city\_trips$started\_at) #Format is yyyy-mm-dd  
city\_trips$month <- format(as.Date(city\_trips$date), "%m")  
city\_trips$day <- format(as.Date(city\_trips$date), "%d")  
city\_trips$year <- format(as.Date(city\_trips$date), "%Y")  
city\_trips$day\_of\_week <- format(as.Date(city\_trips$date), "%A")

1. Add ride the ride length.

city\_trips$ride\_length <- difftime(city\_trips$ended\_at,city\_trips$started\_at)

## Analysis

I will be calculating the mean, median, max, and min values to identify trends and differences in behavior between the different customers.

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual, FUN = mean)

## city\_trips$member\_casual city\_trips$ride\_length  
## 1 casual 1904.4376 secs  
## 2 member 802.1597 secs

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual, FUN = median)

## city\_trips$member\_casual city\_trips$ride\_length  
## 1 casual 946 secs  
## 2 member 562 secs

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual, FUN = max)

## city\_trips$member\_casual city\_trips$ride\_length  
## 1 casual 3356649 secs  
## 2 member 89998 secs

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual, FUN = min)

## city\_trips$member\_casual city\_trips$ride\_length  
## 1 casual -7082 secs  
## 2 member -6845 secs

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual + city\_trips$day\_of\_week, FUN = mean)

## city\_trips$member\_casual city\_trips$day\_of\_week city\_trips$ride\_length  
## 1 casual Friday 1806.1713 secs  
## 2 member Friday 788.3662 secs  
## 3 casual Monday 1888.9178 secs  
## 4 member Monday 778.1024 secs  
## 5 casual Saturday 2056.9014 secs  
## 6 member Saturday 899.5926 secs  
## 7 casual Sunday 2244.4810 secs  
## 8 member Sunday 920.5760 secs  
## 9 casual Thursday 1672.9012 secs  
## 10 member Thursday 754.2513 secs  
## 11 casual Tuesday 1646.0722 secs  
## 12 member Tuesday 751.3066 secs  
## 13 casual Wednesday 1665.9345 secs  
## 14 member Wednesday 755.3065 secs

**Organize the days of the week in order**

city\_trips$day\_of\_week <- ordered(city\_trips$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

**View the average ride time for each day by members v casual riders**

aggregate(city\_trips$ride\_length ~ city\_trips$member\_casual + city\_trips$day\_of\_week, FUN = mean)

## city\_trips$member\_casual city\_trips$day\_of\_week city\_trips$ride\_length  
## 1 casual Sunday 2244.4810 secs  
## 2 member Sunday 920.5760 secs  
## 3 casual Monday 1888.9178 secs  
## 4 member Monday 778.1024 secs  
## 5 casual Tuesday 1646.0722 secs  
## 6 member Tuesday 751.3066 secs  
## 7 casual Wednesday 1665.9345 secs  
## 8 member Wednesday 755.3065 secs  
## 9 casual Thursday 1672.9012 secs  
## 10 member Thursday 754.2513 secs  
## 11 casual Friday 1806.1713 secs  
## 12 member Friday 788.3662 secs  
## 13 casual Saturday 2056.9014 secs  
## 14 member Saturday 899.5926 secs

**View the number of rides and average duration per day by customer type**

city\_trips %>%   
 mutate(weekday = wday(started\_at, label = TRUE)) %>%   
 group\_by(member\_casual, weekday) %>%   
 summarise(number\_of\_rides = n()   
 ,average\_duration = mean(ride\_length/60)) %>%   
 arrange(member\_casual, weekday)

## `summarise()` has grouped output by 'member\_casual'. You can override using the  
## `.groups` argument.

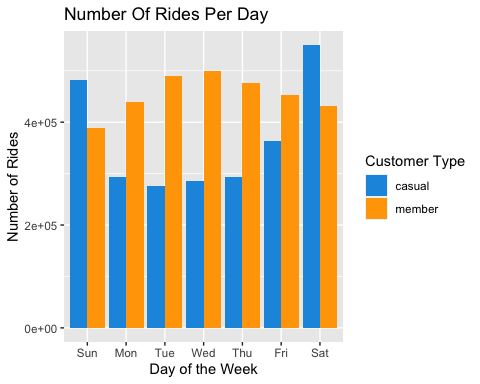
## # A tibble: 14 × 4  
## # Groups: member\_casual [2]  
## member\_casual weekday number\_of\_rides average\_duration  
## <chr> <ord> <int> <drtn>   
## 1 casual Sun 482840 37.40802 secs   
## 2 casual Mon 292996 31.48196 secs   
## 3 casual Tue 276375 27.43454 secs   
## 4 casual Wed 286402 27.76557 secs   
## 5 casual Thu 293632 27.88169 secs   
## 6 casual Fri 364282 30.10285 secs   
## 7 casual Sat 550015 34.28169 secs   
## 8 member Sun 387742 15.34293 secs   
## 9 member Mon 439435 12.96837 secs   
## 10 member Tue 490099 12.52178 secs   
## 11 member Wed 499936 12.58844 secs   
## 12 member Thu 475334 12.57086 secs   
## 13 member Fri 453113 13.13944 secs   
## 14 member Sat 431331 14.99321 secs

## 

## Visualizations

#### Number of rides per day by customer type

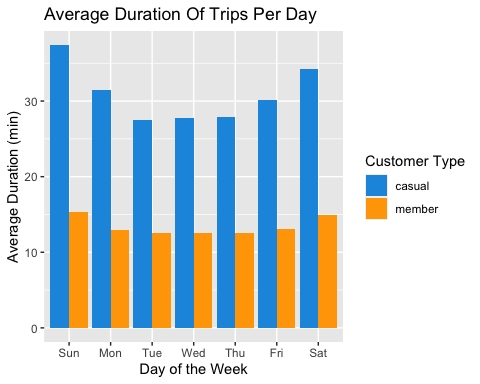
city\_trips %>%   
 mutate(weekday = wday(started\_at, label = TRUE)) %>%   
 group\_by(member\_casual, weekday) %>%   
 summarise(number\_of\_rides = n()  
 ,average\_duration = mean(ride\_length)) %>%   
 arrange(member\_casual, weekday) %>%   
 ggplot(aes(x = weekday, y = number\_of\_rides, fill = member\_casual)) +  
 geom\_col(position = "dodge") + scale\_fill\_manual(values = c("casual" = "#1b98e0", "member" = "orange")) + labs(title= "Number Of Rides Per Day",   
 x = "Day of the Week", y = "Number of Rides",   
 fill = "Customer Type")



#### 

#### Average trip duration per day by customer type

city\_trips %>%   
 mutate(weekday = wday(started\_at, label = TRUE)) %>%   
 group\_by(member\_casual, weekday) %>%   
 summarise(number\_of\_rides = n()  
 ,average\_duration = mean(ride\_length/60)) %>%   
 arrange(member\_casual, weekday) %>%   
 ggplot(aes(x = weekday, y = average\_duration, fill = member\_casual)) +  
 geom\_col(position = "dodge") + scale\_fill\_manual(values = c("casual" = "#1b98e0", "member" = "orange"))+  
 labs(title= "Average Duration Of Trips Per Day",  
 x = "Day of the Week", y = "Average Duration (min)",   
 fill = "Customer Type")



## Conclusions

1. Casual riders average more than two times the rate of members in trip duration.
2. Members take more rides than casual customers except on weekends.

## Recommendations

1. Since casual riders average longer trips, increase their trip rate, which will incentivize them to switch to a member with lower trip rates.
2. Develop member incentives such as store promotions or weekend promotions where the casual riders usually peak.