Cyclistic bike-share analysis

Pranay Reddy

2024-08-23

INTRODUCTION

Cyclistic, a bike-share company based in Chicago, has rapidly grown since its launch in 2016. The company now operates over 5,800 bicycles across 692 stations, offering a variety of bike options, including traditional bikes, reclining bikes, hand tricycles, and cargo bikes. The objective of this analysis is to explore how annual members and casual riders use Cyclistic bikes differently. By understanding these differences, the marketing team can design targeted strategies to encourage casual riders to become annual members. The findings from this analysis will support Cyclistic's efforts to increase the number of annual memberships. This report give detail analysis of difference in usage pattern between members and casual riders.

Business Task

The primary goal of analysis is to understand how the customers uses bikes differently and understand for developing marketing strategies.

Ask

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

Prepare

Data Sources

Cyclistic's historical bike trip data for the past 12 months, provided by Motivate International Inc.The data is publicly available under a license that prohibits the use of personally identifiable information.

Data Processing

The data is provied in 12 different csv files so first we read the files

```
setwd("C:/Users/Pranay/Desktop/data")
d1<-read.csv("202301-divvy-tripdata.csv")
d2<-read.csv("202302-divvy-tripdata.csv")
d3<-read.csv("202303-divvy-tripdata.csv")
```

```
d4<-read.csv("202304-divvy-tripdata.csv")
d5<-read.csv("202305-divvy-tripdata.csv")
d6<-read.csv("202306-divvy-tripdata.csv")
d7<-read.csv("202307-divvy-tripdata.csv")
d8<-read.csv("202308-divvy-tripdata.csv")
d9<-read.csv("202309-divvy-tripdata.csv")
d10<-read.csv("202310-divvy-tripdata.csv")
d11<-read.csv("202311-divvy-tripdata.csv")
d12<-read.csv("202312-divvy-tripdata.csv")
```

merge the files

```
tripdata_2023<- rbind(d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12)
```

Load the required packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
library(lubridate)
library(dplyr)
library(modeest)
library(tidyr)
```

overview of data

```r
str(tripdata_2023)
```

```
## 'data.frame':    5719877 obs. of  13 variables:
##  $ ride_id           : chr  "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C90792D034FED9
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2023-01-21 20:05:42" "2023-01-10 15:37:36" "2023-01-02 07:51:57" "2023-
##  $ ended_at          : chr  "2023-01-21 20:16:33" "2023-01-10 15:46:05" "2023-01-02 08:05:11" "2023-
##  $ start_station_name: chr  "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western Ave & Lun
##  $ start_station_id  : chr  "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr  "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli Produce - E
##  $ end_station_id    : chr  "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num  41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num  -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "casual" "member" ...
```

Check for duplicates

```r
df_unique <- tripdata_2023 %>%
  distinct(ride_id, .keep_all = TRUE)

# Print the number of rows before and after removing duplicates
original_row_count <- nrow(df)
new_row_count <- nrow(df_unique)
duplicates_removed <- original_row_count - new_row_count

print(paste("Number of duplicates removed:", duplicates_removed))
```

```
## [1] "Number of duplicates removed: "
```

Divide started at and ended at to start date start time and end date end time simplifying calculations. creating a another data frame to take only required data

```r
## split the started_at and ended_at
tripdata_2023 <- tripdata_2023 %>%
  mutate(
    started_date = as.Date(started_at),
    started_time = format(as.POSIXct(started_at), "%H:%M:%S"),
    ended_date = as.Date(ended_at),
    ended_time = format(as.POSIXct(ended_at), "%H:%M:%S")
  )
## creating a another data frame to take only required data
```

```r
tripdata_2023_v2 <- tripdata_2023[, c("ride_id", "rideable_type","start_station_name","started_date", "
##replacing ride_id with numbers in sequence and converting ride_id to character data type
tripdata_2023_v2$ride_id <- seq(1, nrow(tripdata_2023_v2))
tripdata_2023_v2$ride_id <- as.character(tripdata_2023_v2$ride_id)
```

calculating ride duration and assigning week and month name

```r
##calculating ride duration and assigning week and month name
tripdata_2023_v2 <- tripdata_2023_v2 %>%
  mutate(
    # Calculate ride duration in minutes
    ride_duration_minutes = as.numeric(difftime(
      ymd_hms(paste(ended_date, ended_time, sep=" ")),
      ymd_hms(paste(started_date, started_time, sep=" ")),
      units = "mins"
    )),

    # Extract week name and month name
    week_name = wday(ymd_hms(paste(started_date, started_time, sep=" ")), label = TRUE, abbr = FALSE),
    month_name = month(ymd_hms(paste(started_date, started_time, sep=" ")), label = TRUE, abbr = FALSE)
  )
# filter to remove negative values
tripdata_2023_v2.1<- tripdata_2023_v2 %>%
  filter(ride_duration_minutes>=0)
```

Analyze

calculating total rides

```r
total_ride<-tripdata_2023_v2.1 %>%
  group_by(member_casual) %>%
  summarise(total_rides=n())
```

calculating mean rides

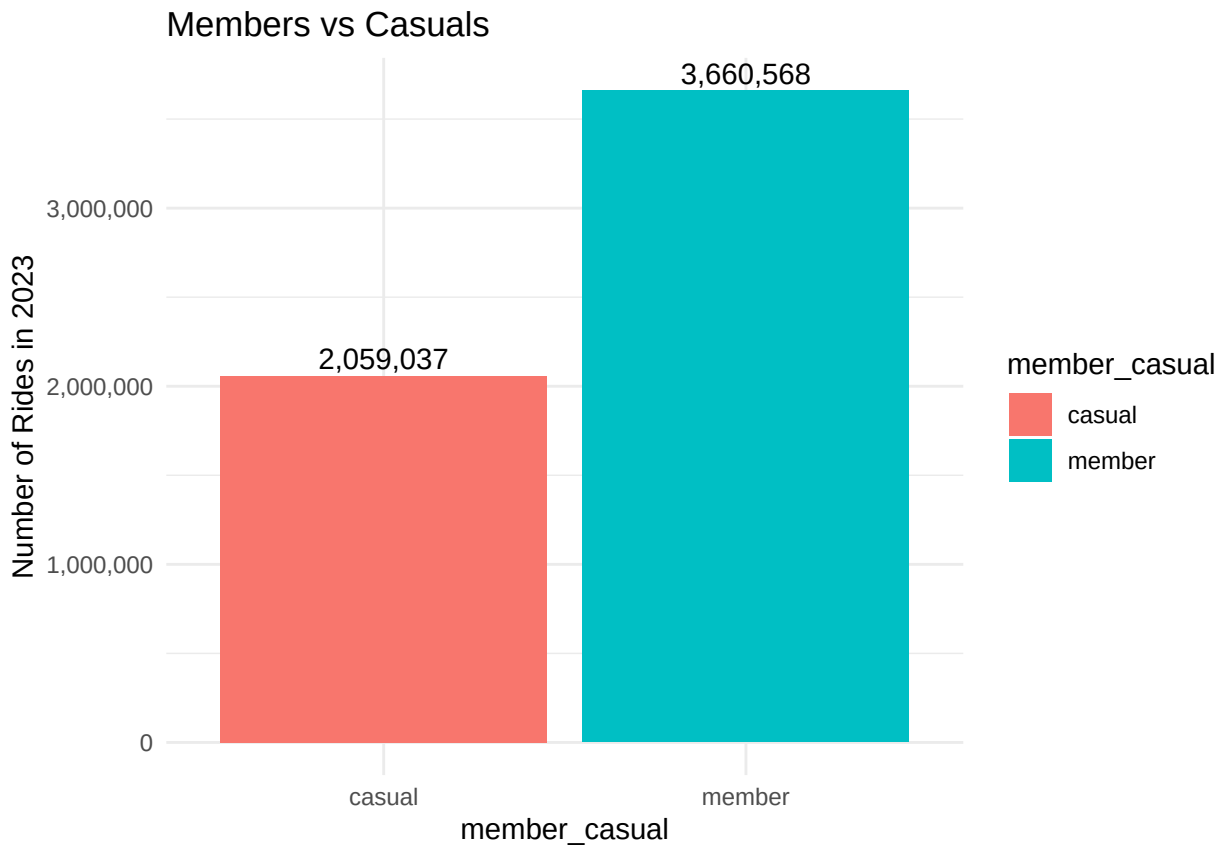
```r
avg_ride<-tripdata_2023_v2.1 %>%
  group_by(member_casual) %>%
  summarise(avg_length=mean(ride_duration_minutes,na.rm = TRUE))
```

Central Tendency of data

```r
total_rides_cal <-tripdata_2023_v2.1 %>%
  group_by(member_casual) %>%
  summarise(
    mean_rides=mean(ride_duration_minutes,na.rm = TRUE),
    median_rides=median(ride_duration_minutes,na.rm = TRUE),
    mode_rides=mfv(ride_duration_minutes,na_rm = TRUE),
    sd_rides=sd(ride_duration_minutes,na.rm = TRUE),
    max_rides=max(ride_duration_minutes,na.rm = TRUE),
    min_rides=min(ride_duration_minutes)
  )
```

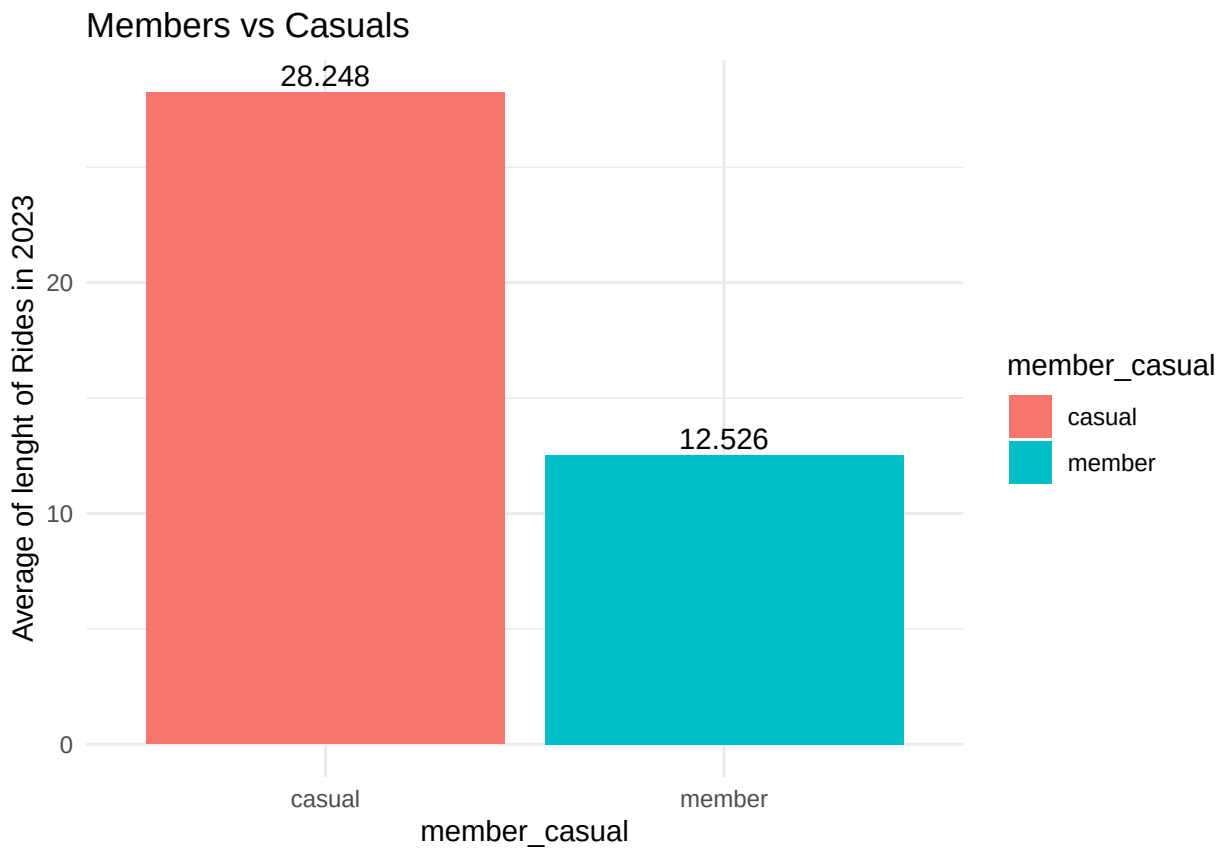
plot for rides for members and casuals

```
ggplot(data = total_ride) +
  geom_bar(mapping = aes(x = member_casual, y = total_rides, fill = member_casual), stat = "identity") +
  geom_text(aes(x = member_casual, y = total_rides, label = scales::comma(total_rides)),
            vjust = -0.3, size = 4) +  # Adjust position and size of labels
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Members vs Casuals",
    y = "Number of Rides in 2023"
  ) +
  theme_minimal()
```



Plot for average rides for members and casuals

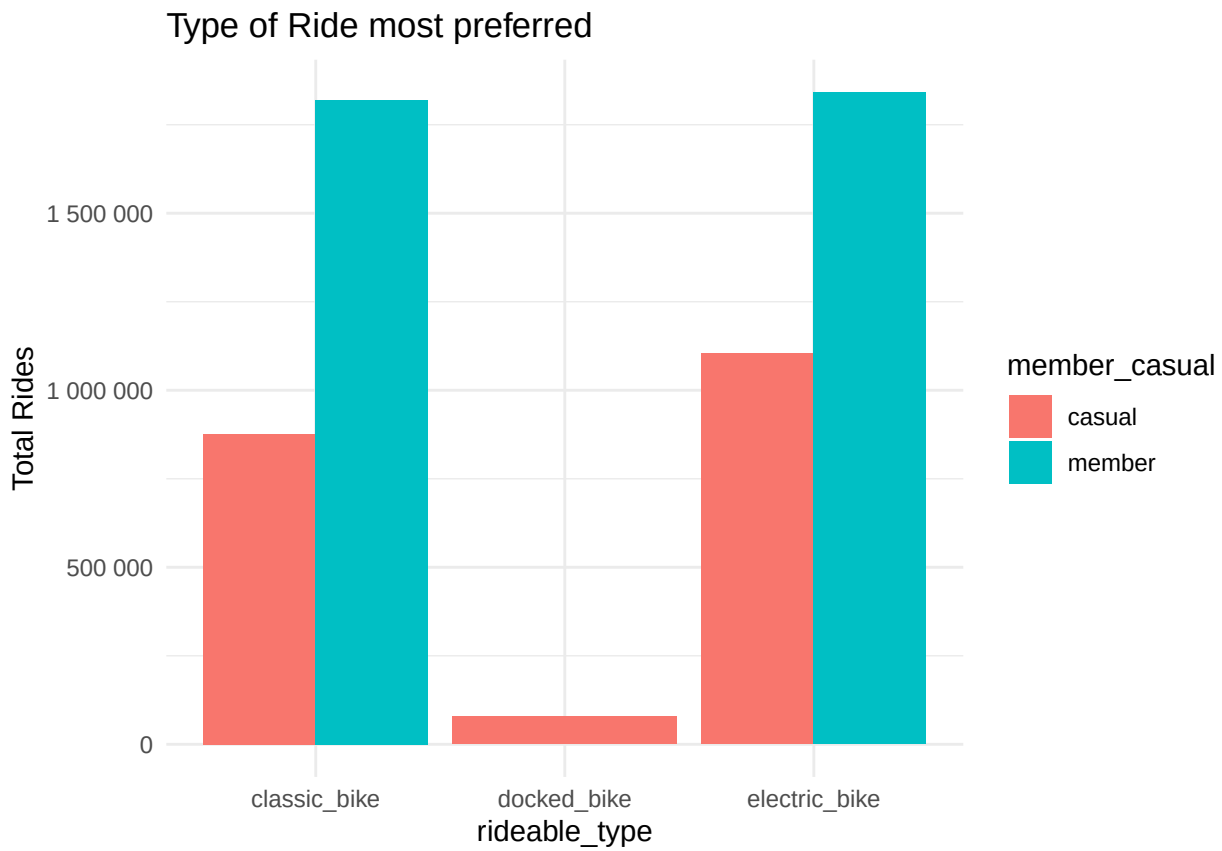
```
ggplot(data = avg_ride) +
  geom_bar(mapping = aes(x = member_casual, y = avg_length, fill = member_casual), stat = "identity") +
  geom_text(aes(x = member_casual, y = avg_length, label = scales::number(avg_length,accuracy = 0.001)),
            vjust = -0.3, size = 4) +  # Adjust position and size of labels
  scale_y_continuous(labels = scales::number) +
  labs(
    title = "Members vs Casuals",
    y = "Average of lenght of Rides in 2023"
  ) +
  theme_minimal()
```



Type of ride used for members and casuals

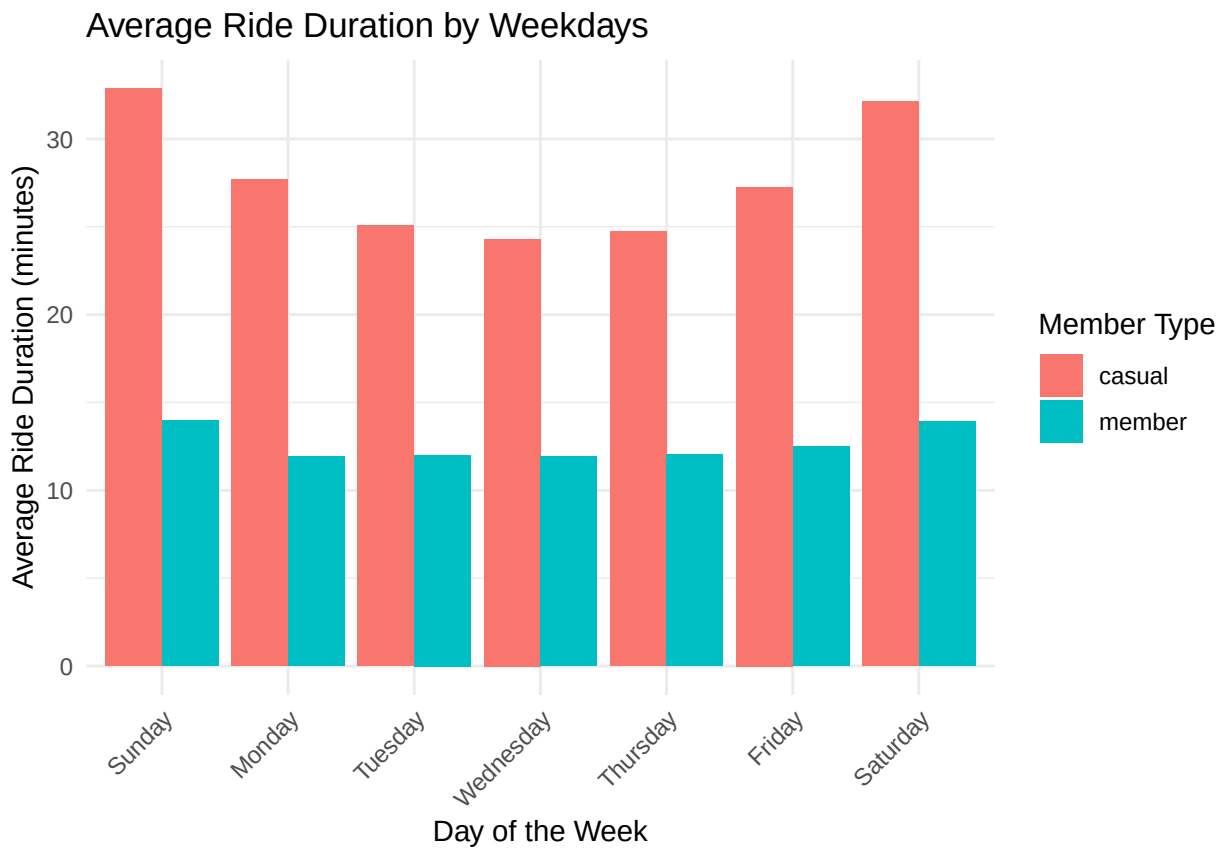
```
tripdata_2023_v2.1 %>%
  group_by(rideable_type, member_casual) %>%
  summarise(total_rides = n(), avg_duration = mean(ride_duration_minutes, na.rm = TRUE)) %>%
  ggplot(mapping = aes(x = rideable_type, y = total_rides, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Type of Ride most preferred",
    y = "Total Rides"
  ) +
  scale_y_continuous(labels = scales::number) +
  theme_minimal()
```

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



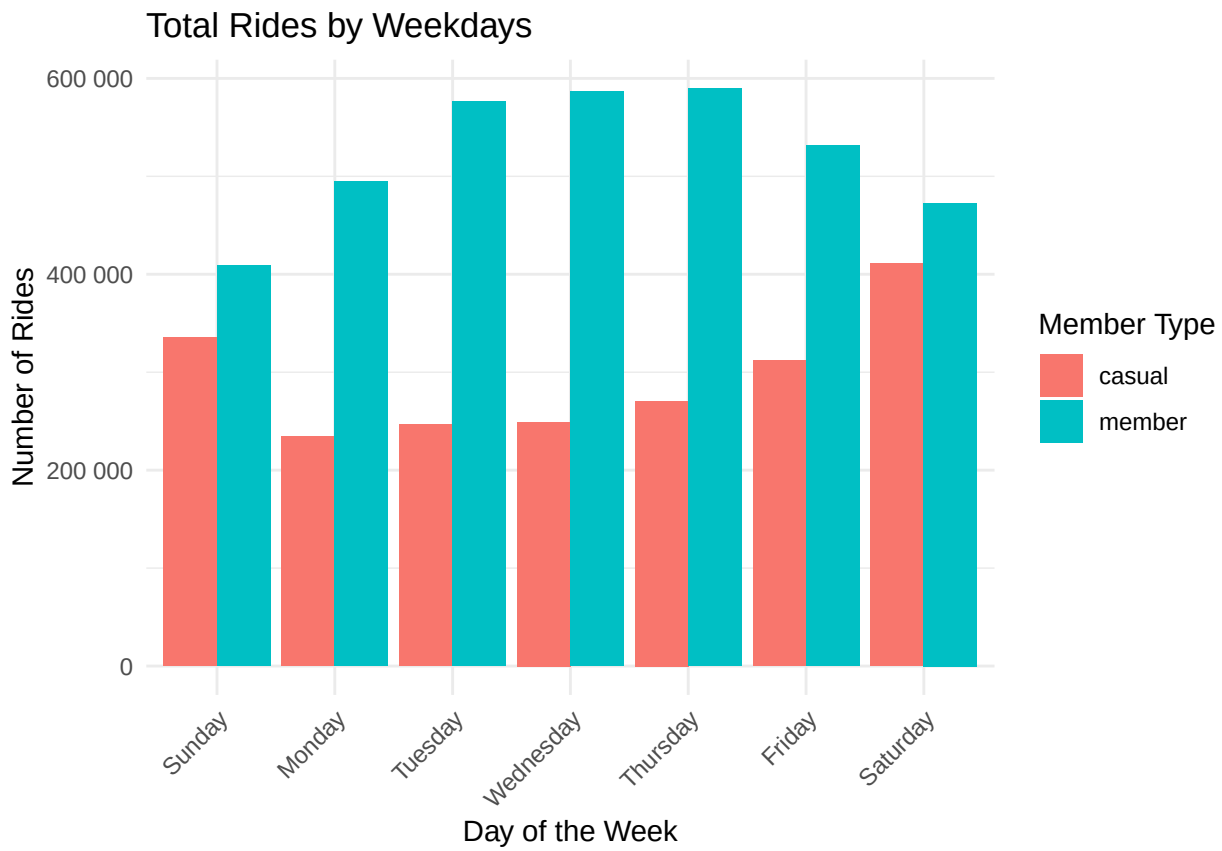
Average ride duration for Members and casuals for weekdays

```
tripdata_2023_v2.1 %>%
  group_by(member_casual, week_name) %>%
  summarise(total_rides = n(), avg_duration = mean(ride_duration_minutes, na.rm = TRUE), .groups = 'dro
  arrange(member_casual, week_name) %>%
  ggplot(aes(x = week_name, y = avg_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(
    title = "Average Ride Duration by Weekdays",
    x = "Day of the Week",
    y = "Average Ride Duration (minutes)",
    fill = "Member Type"
  ) +
  theme_minimal()+theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Total rides by weekdays

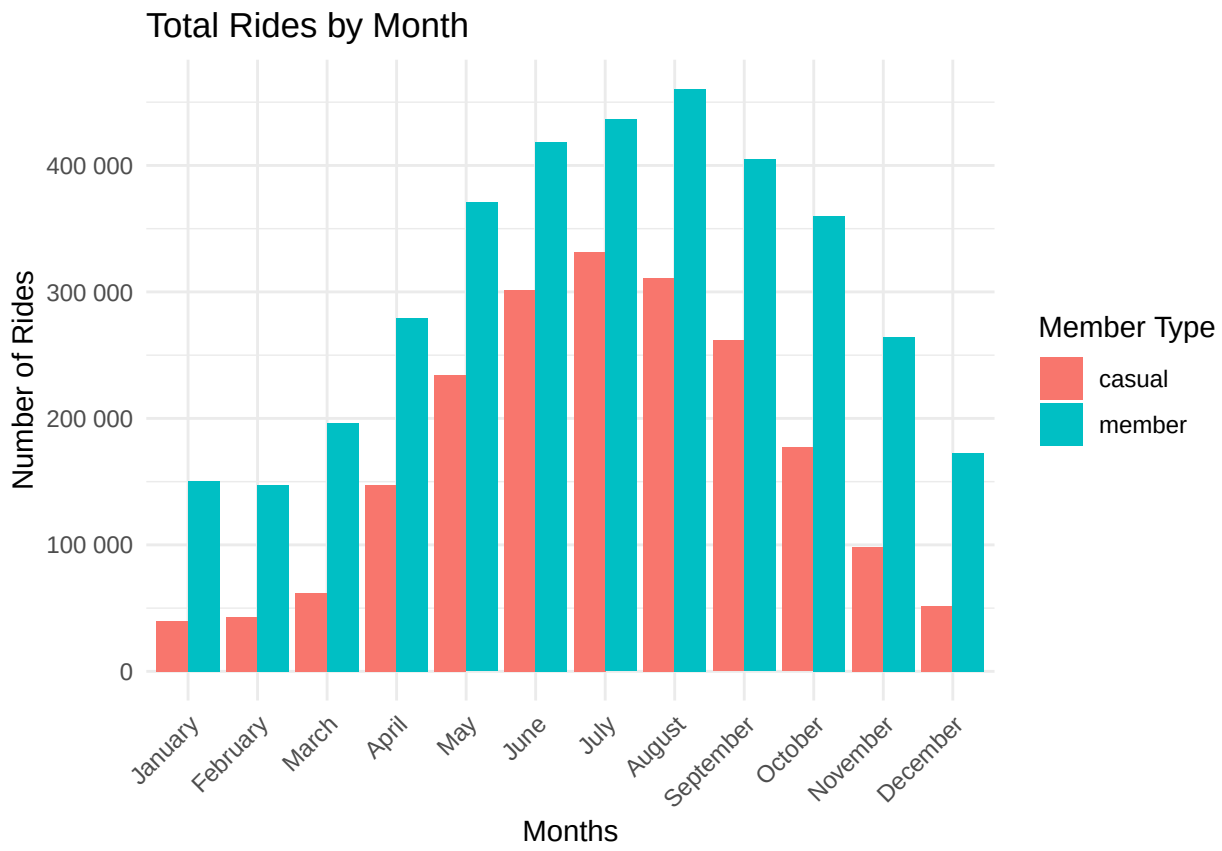
```
tripdata_2023_v2.1 %>%
  group_by(member_casual, week_name) %>%
  summarise(total_rides = n(), avg_duration = mean(ride_duration_minutes, na.rm = TRUE), .groups = 'drop
  arrange(member_casual, week_name) %>%
  ggplot(aes(x = week_name, y = total_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(
    title = "Total Rides by Weekdays",
    x = "Day of the Week",
    y = "Number of Rides",
    fill = "Member Type"
  ) +scale_y_continuous(labels = scales::number)+
  theme_minimal()+theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Total rides in 2023 by months

```
tripdata_2023_v2.1 %>%
  group_by(member_casual,month_name) %>%
  summarise(total_rides=n(),avg_duration=mean(ride_duration_minutes,na.rm = TRUE))%>%
  arrange(member_casual,month_name)%>%
  ggplot(aes(x = month_name, y = total_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(
    title = "Total Rides by Month",
    x = "Months",
    y = "Number of Rides",
    fill = "Member Type"
  ) +scale_y_continuous(labels = scales::number)+
  theme_minimal()+theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

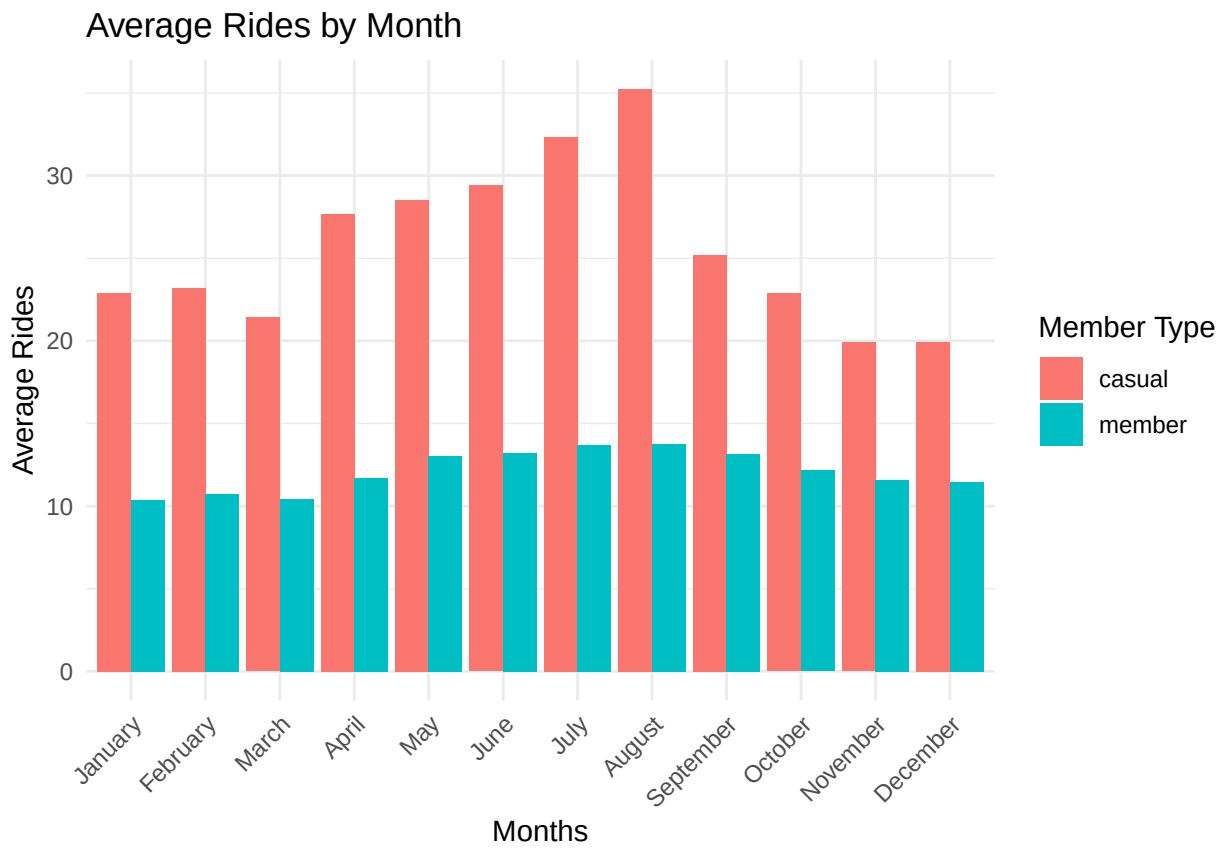
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



Average rides by months in 2023

```
tripdata_2023_v2.1 %>%
  group_by(member_casual,month_name) %>%
  summarise(total_rides=n(),avg_duration=mean(ride_duration_minutes,na.rm = TRUE))%>%
  arrange(member_casual,month_name)%>%
  ggplot(aes(x = month_name, y = avg_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(
    title = "Average Rides by Month",
    x = "Months",
    y = "Average Rides",
    fill = "Member Type"
  ) +scale_y_continuous(labels = scales::number)+
  theme_minimal()+theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

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



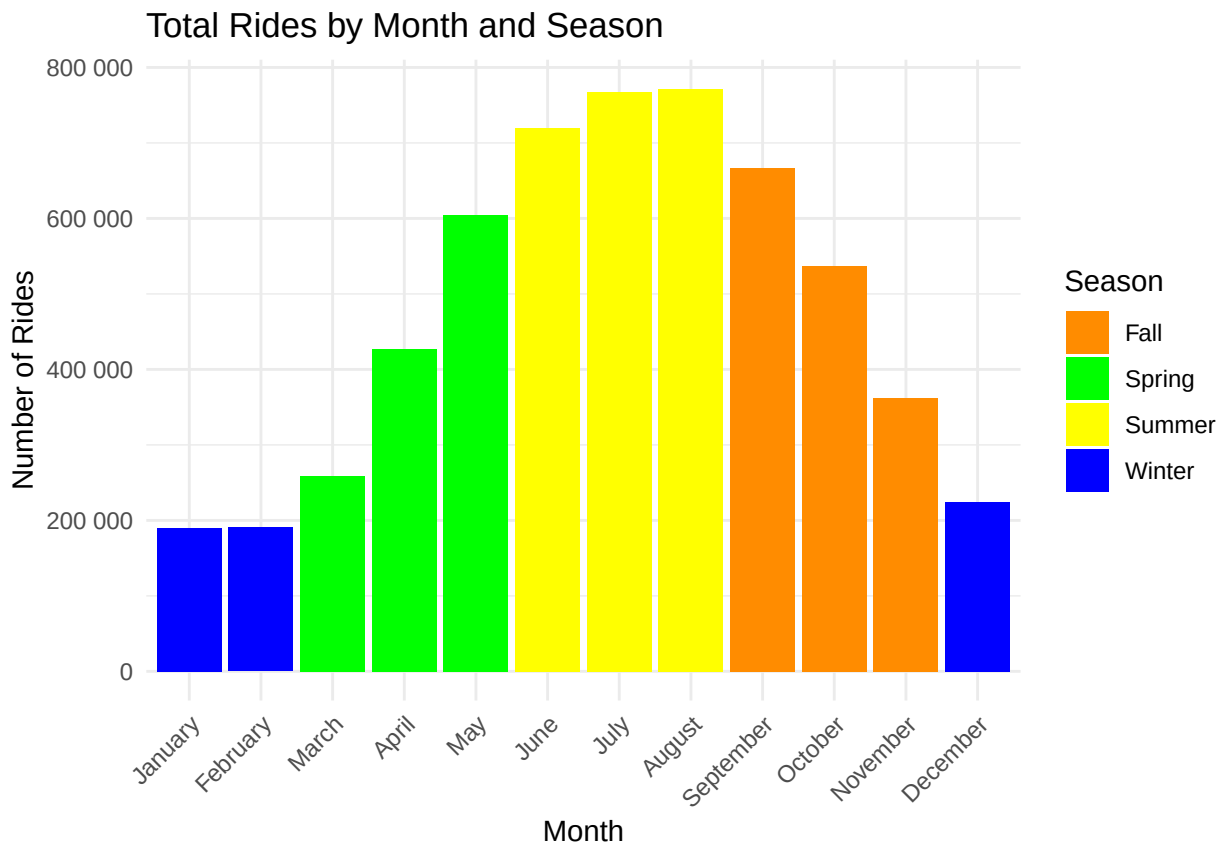
Dividing months by seasons to get total rides by month and season

```
tripdata_2023_v2.1 %>%
  group_by(month_name) %>%
  summarise(total_rides=n(),avg_duration=mean(ride_duration_minutes,na.rm = TRUE))%>%
  arrange(month_name)
```

```
## # A tibble: 12 x 3
##    month_name total_rides avg_duration
##    <ord>            <int>        <dbl>
##  1 January         190301         13.0
##  2 February        190444         13.5
##  3 March           258678         13.1
##  4 April           426586         17.2
##  5 May             604817         19.0
##  6 June            719611         20.0
##  7 July            767620         21.7
##  8 August          771633         22.4
##  9 September       666321         17.9
## 10 October         537077         15.7
## 11 November        362454         13.8
## 12 December        224063         13.4
```

```
get_season <- function(month) {
  case_when(
```

```
    month %in% c("December", "January", "February") ~ "Winter",
    month %in% c("March", "April", "May") ~ "Spring",
    month %in% c("June", "July", "August") ~ "Summer",
    month %in% c("September", "October", "November") ~ "Fall"
  )
}
# Plot total rides by month and season
tripdata_2023_v2.1 %>%
  group_by(month_name) %>%
  summarise(
    total_rides = n(),
    avg_duration = mean(ride_duration_minutes, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(season = get_season(month_name)) %>%
  arrange(month_name) %>%
  ggplot(aes(x = month_name, y = total_rides, fill = season)) +
  geom_col(position = "dodge") +
  labs(
    title = "Total Rides by Month and Season",
    x = "Month",
    y = "Number of Rides",
    fill = "Season"
  ) +
  scale_y_continuous(labels = scales::number) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Winter" = "blue", "Spring" = "green", "Summer" = "yellow", "Fall" = "da
```



Popular starting stations

```
tripdata_2023_v2.1 %>%
  group_by(start_station_name) %>%
  summarise(no_of_starts = n()) %>%
  filter(start_station_name != "") %>%
  arrange(desc(no_of_starts))
```

```
## # A tibble: 1,592 x 2
##    start_station_name                  no_of_starts
##    <chr>                                      <int>
##  1 Streeter Dr & Grand Ave                    63249
##  2 DuSable Lake Shore Dr & Monroe St          40288
##  3 Michigan Ave & Oak St                      37383
##  4 DuSable Lake Shore Dr & North Blvd         35966
##  5 Clark St & Elm St                          35805
##  6 Kingsbury St & Kinzie St                   34965
##  7 Wells St & Concord Ln                      33588
##  8 Clinton St & Washington Blvd               32715
##  9 Wells St & Elm St                          30407
## 10 Millennium Park                            30154
## # i 1,582 more rows
```

Popular final stations for all users by end station

```r
tripdata_2023_v2.1 %>%
  group_by(end_station_name) %>%
  summarise(no_of_ends = n()) %>%
  filter(end_station_name != "") %>%
  arrange(desc(no_of_ends))
```

```
## # A tibble: 1,597 x 2
##    end_station_name                   no_of_ends
##    <chr>                                   <int>
##  1 Streeter Dr & Grand Ave                 64197
##  2 DuSable Lake Shore Dr & North Blvd      39299
##  3 DuSable Lake Shore Dr & Monroe St       38022
##  4 Michigan Ave & Oak St                   37994
##  5 Clark St & Elm St                       34962
##  6 Kingsbury St & Kinzie St                34253
##  7 Wells St & Concord Ln                   34172
##  8 Clinton St & Washington Blvd            33394
##  9 Millennium Park                         31049
## 10 Theater on the Lake                     30596
## # i 1,587 more rows
```

popular starting stations for members

```r
tripdata_2023_v2.1 %>%
  filter(member_casual == 'member') %>%
  group_by(start_station_name) %>%
  summarise(no_of_starts = n()) %>%
  filter(start_station_name != "") %>%
  arrange(desc(no_of_starts))
```

```
## # A tibble: 1,455 x 2
##    start_station_name           no_of_starts
##    <chr>                               <int>
##  1 Clinton St & Washington Blvd        26216
##  2 Kingsbury St & Kinzie St            26171
##  3 Clark St & Elm St                   25001
##  4 Wells St & Concord Ln               21418
##  5 Clinton St & Madison St             20596
##  6 Wells St & Elm St                   20400
##  7 University Ave & 57th St            20038
##  8 Broadway & Barry Ave                18959
##  9 Loomis St & Lexington St            18900
## 10 State St & Chicago Ave              18484
## # i 1,445 more rows
```

Popular final stations for members by end station

```r
tripdata_2023_v2.1 %>%
  filter(member_casual == 'member') %>%
  group_by(end_station_name) %>%
  summarise(no_of_ends = n()) %>%
  filter(end_station_name != "") %>%
  arrange(desc(no_of_ends))
```

```
## # A tibble: 1,455 x 2
##    end_station_name             no_of_ends
##    <chr>                             <int>
##  1 Clinton St & Washington Blvd      27445
##  2 Kingsbury St & Kinzie St          26366
##  3 Clark St & Elm St                 24858
##  4 Wells St & Concord Ln             22248
##  5 Clinton St & Madison St           22095
##  6 Wells St & Elm St                 20227
##  7 University Ave & 57th St          20217
##  8 Broadway & Barry Ave              19393
##  9 State St & Chicago Ave            19027
## 10 Loomis St & Lexington St          18591
## # i 1,445 more rows
```

Popular starting stations for casual

```
tripdata_2023_v2.1 %>%
  filter(member_casual == 'casual') %>%
  group_by(start_station_name) %>%
  summarise(no_of_starts = n()) %>%
  filter(start_station_name != "") %>%
  arrange(desc(no_of_starts))
```

```
## # A tibble: 1,549 x 2
##    start_station_name                 no_of_starts
##    <chr>                                     <int>
##  1 Streeter Dr & Grand Ave                   46030
##  2 DuSable Lake Shore Dr & Monroe St         30487
##  3 Michigan Ave & Oak St                     22664
##  4 DuSable Lake Shore Dr & North Blvd        20338
##  5 Millennium Park                           20226
##  6 Shedd Aquarium                            17781
##  7 Theater on the Lake                       16359
##  8 Dusable Harbor                            15490
##  9 Wells St & Concord Ln                     12170
## 10 Montrose Harbor                           11987
## # i 1,539 more rows
```

Popular final stations for casual users by end station

```
tripdata_2023_v2.1 %>%
  filter(member_casual == 'casual') %>%
  group_by(end_station_name) %>%
  summarise(no_of_ends = n()) %>%
  filter(end_station_name != "") %>%
  arrange(desc(no_of_ends))
```

```
## # A tibble: 1,543 x 2
##    end_station_name                 no_of_ends
##    <chr>                                 <int>
##  1 Streeter Dr & Grand Ave               49310
```

```
##  2 DuSable Lake Shore Dr & Monroe St       27539
##  3 Michigan Ave & Oak St                   23688
##  4 DuSable Lake Shore Dr & North Blvd      23255
##  5 Millennium Park                         22219
##  6 Theater on the Lake                     17572
##  7 Shedd Aquarium                          15652
##  8 Dusable Harbor                          13558
##  9 Wells St & Concord Ln                   11924
## 10 Montrose Harbor                         11640
## # i 1,533 more rows
```

Key findings

Casual riders takes longer rides compared to annual members.

Annual members use bikes during weekdays, while casual riders use bikes more on weekends.

Classic bikes are the most preferred bike for both members and casuals.

Electric bikes are more popular among casual riders.

Summer months have more rides compared to other seasons and Winter have least number of rides.

Most popular starting stations are Streeter Dr & Grand Ave and DuSable Lake Shore Dr & Monroe St

Most popular ending stations are Streeter Dr & Grand Ave and DuSable Lake Shore Dr & North Blvd

Most popular starting stations for casuals are Streeter Dr & Grand Ave,DuSable Lake Shore Dr & Monroe St and Michigan Ave & Oak St

Most popular ending stations for casuals are Streeter Dr & Grand Ave,DuSable Lake Shore Dr & Monroe St and Michigan Ave & Oak St

Recommendations

1. Advertise about annual membership and other new membership at popular locations of casual riders.
2. Introduce special packages like weekend specific planes to encourage for taking a membership.
3. Focus on converting casual riders to annual members by offering weekend-specific discounts on annual memberships.
4. Develop a loyalty program that rewards casual riders with points on frequency of use, Where points can be redeemed for membership discounts or other befits.