bikesharing

Maciej

2023-08-17

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.

Three questions will guide the future marketing program:

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?

Bike-share project

The objectives of this project is:

- Grab data from given source : Link for files
- Consolidate/aggregate data of bike trips from bike-sharing company
- Make analysis due to business task.
- Share insights and recommendations

Installing required packages

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(ggplot2)
library(dplyr)
library(here)
```

```
## here() starts at C:/Users/krzyw/OneDrive/Dokumenty/r-projects/bike-share-v1
```

```
library(aws.s3)
```

Setting global language

```
Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

Setting enviroment for aws s3

```
Sys.setenv("AWS_DEFAULT_REGION" = "eu-north-1")
```

Checking connection to bucket

```
bucket_exists(
  bucket = "s3://analitycal-projects-data/",
  region = "eu-north-1"
)
```

```
## [1] TRUE
## attr(,"x-amz-id-2")
## [1] "sR+HTCt0gTt8+Dnb5DHN1CFfz9coHG6NJdp6kO4ltgJaoK7+19eWCR9vEvCOw9FKl9Tqu/np22Y="
## attr(,"x-amz-request-id")
## [1] "D949WFXA4PFXFX42"
## attr(,"date")
## [1] "Mon, 11 Sep 2023 14:57:29 GMT"
## attr(,"x-amz-bucket-region")
## [1] "eu-north-1"
## attr(,"x-amz-access-point-alias")
## [1] "false"
## attr(,"content-type")
## [1] "application/xml"
## attr(,"server")
## [1] "AmazonS3"
```

Listing bucket contents and saving file names

```
bucket_objects <- get_bucket("analitycal-projects-data")
first_12_objects <- head(bucket_objects,12)
names_of_s3_files <- sapply(first_12_objects, function(obj) obj$Key)
```

Downloading csv files from bucket

There is 12 CSV files to download in cyclist_data/csv_files folder. Downloaded files are stored at project directory

```
subfolder = "cyclist_data/csv_files/"
files_date <- c("2022_08", "2022_09", "2022_10", "2022_11", "2022_12", "2023_01", "2023_02", "2023_03", "2023_04", "2023_05", "2023_06", "2023_07
")
file_ending = "_trip_data.csv"



for (name_of_s3_file in names_of_s3_files) {

  save_object(
    object = name_of_s3_file,
    bucket = "s3://analitycal-projects-data/",
    region = "eu-north-1",
    file = name_of_s3_file
  )

}
```

Setting relative path to csv files

```
main_path <- here("cyclist_data", "csv_files")
print(main_path)
```

```
## [1] "C:/Users/krzyw/OneDrive/Dokumenty/r-projects/bike-share-v1/cyclist_data/csv_files"
```

Reading CSV files and saving to list

Reading already downloaded files that are stored at cyclist_data/csv_files

```
list_of_month_trips <- list()
i <- 1

for (name_of_s3_file in names_of_s3_files) {
  file_path <- name_of_s3_file
  data <- read.csv(file_path)
  list_of_month_trips[[i]] <- data
  i <- i + 1
}
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
## : Koniec pliku wewnątrz cudzysłowia

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
## : Koniec pliku wewnątrz cudzysłowia

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
## : Koniec pliku wewnątrz cudzysłowia
```

Dropping irrelevant columns

Columns associated with place of trips are irrelevant because business tasks are related to when and how questions but not where. For example information about the most frequent start/end station will not help to find patterns that can help to find a way to attract customers to buy a membership. Eventually it can be useful for placing advertisments in these places.

```
relevant_columns <- c("ride_id","started_at","ended_at","member_casual")
list_length = length(list_of_month_trips)
for(i in 1:list_length)
{
  list_of_month_trips[[i]] <- subset(list_of_month_trips[[i]],select=relevant_columns)

}
```

Consolidating the 12 datasets into big one df

```
all_trips <- bind_rows(list_of_month_trips)
```

Summary

```
summary(all_trips)
```

```
##    ride_id           started_at          ended_at         member_casual
##  Length:5115139     Length:5115139     Length:5115139     Length:5115139
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
head(all_trips)
```

```
##            ride_id          started_at            ended_at member_casual
## 1 550CF7EFEAE0C618 2022-08-07 21:34:15 2022-08-07 21:41:46       casual
## 2 DAD198F405F9C5F5 2022-08-08 14:39:21 2022-08-08 14:53:23       casual
## 3 E6F2BC47B65CB7FD 2022-08-08 15:29:50 2022-08-08 15:40:34       casual
## 4 F597830181C2E13C 2022-08-08 02:43:50 2022-08-08 02:58:53       casual
## 5 0CE689BB4E313E8D 2022-08-07 20:24:06 2022-08-07 20:29:58       casual
## 6 BFA7E7CC69860C20 2022-08-08 13:06:08 2022-08-08 13:19:09       casual
```

Adding date columns

```
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
```

Changing column types

Making changes to column type started_at and ended_at from char to posixct due to latter computation - ride_length

```
all_trips$started_at <-as.POSIXct(all_trips$started_at,
                                  format = "%Y-%m-%d %H:%M:%S")
all_trips$ended_at <-as.POSIXct(all_trips$ended_at,
                                format = "%Y-%m-%d %H:%M:%S")
```

Adding ride_length column

Adding ride_length column that compute the difference in seconds between start and end of the trip

```
all_trips$ride_length <- as.numeric(all_trips$ended_at - all_trips$started_at)
```

Filtering and making new subset of records

Deleting rows that have ride length less than 0 - these rows are irrelevant from analysis point of view and rows that have empty string in member_casual

```
all_trips_v2 <- all_trips %>%
  filter(ride_length > 0, member_casual %in% c("member", "casual"))
```

Comparing members and casual riders

Aggregating groups the unique values in member_casual column and compute given statistical value for each group

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1560.7605
## 2                     member                 731.4244
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      710
## 2                     member                      507
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  2486835
## 2                     member                    93597
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        1
## 2                     member                        1
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = summary)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length.Min.
## 1                     casual                        1.0000
## 2                     member                        1.0000
##   all_trips_v2$ride_length.1st Qu. all_trips_v2$ride_length.Median
## 1                         400.0000                        710.0000
## 2                         294.0000                        507.0000
##   all_trips_v2$ride_length.Mean all_trips_v2$ride_length.3rd Qu.
## 1                     1560.7605                        1323.0000
## 2                      731.4244                         872.0000
##   all_trips_v2$ride_length.Max.
## 1                    2486835.0000
## 2                      93597.0000
```

Defining the order of caterogical values

Making order of day_of_week column and their values without changing the factual order of records

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"
))
```

Aggregating the average ride time

Aggregating by each day and each member type in ordered manner

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                  Sunday                1850.8723
## 2                      member                  Sunday                 809.5448
## 3                      casual                  Monday                1478.7873
## 4                      member                  Monday                 699.9498
## 5                      casual                 Tuesday                1388.5087
## 6                      member                 Tuesday                 702.2417
## 7                      casual               Wednesday                1309.2345
## 8                      member               Wednesday                 696.8366
## 9                      casual                Thursday                1338.6006
## 10                     member                Thursday                 703.2275
## 11                     casual                  Friday                1530.0299
## 12                     member                  Friday                 727.6204
## 13                     casual                Saturday                1813.5846
## 14                     member                Saturday                 818.6968
```

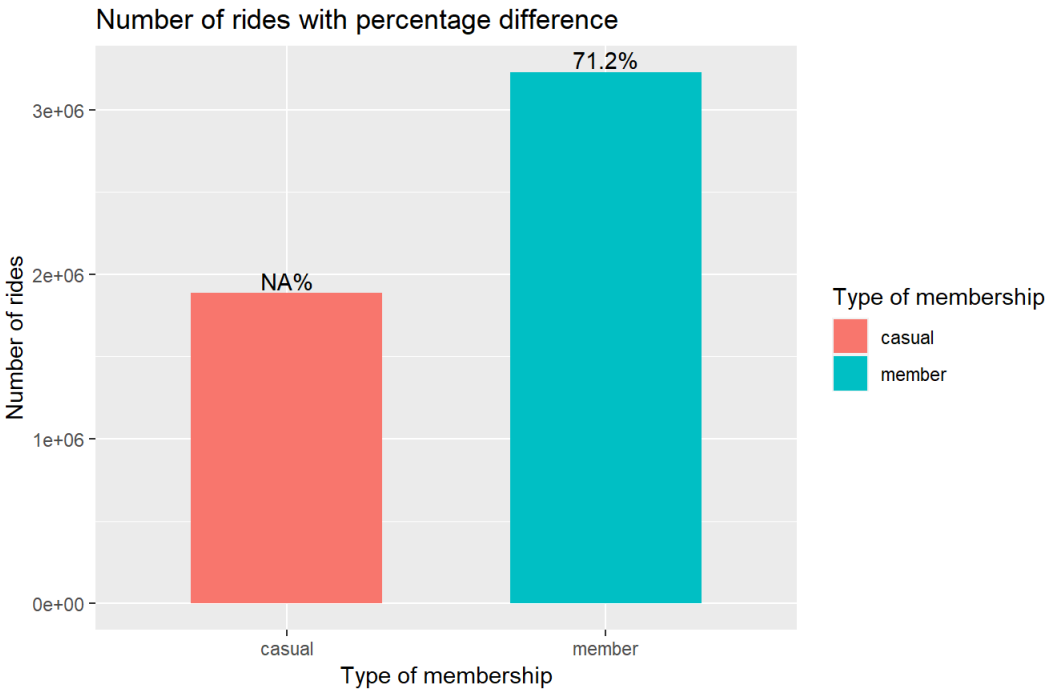
Plot 1

Comparing the number of rides by each type of membership

```
data <- all_trips_v2 %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ungroup() %>%
  mutate(percent_change = ((number_of_rides - lag(number_of_rides)) / lag(number_of_rides)) * 100)

p <- ggplot(data, aes(x = member_casual, y = number_of_rides, fill = member_casual)) +
  geom_col(width=0.6) +
  geom_text(aes(label = paste0(round(percent_change, 1), "%")),
     vjust = -0.2) +
  labs(title = "Number of rides with percentage difference",
       y = "Number of rides",
       x = "Type of membership",
       caption ="Regular users use bikes more (amount of rentals) by 71.2 on average.",
       fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))

print(p)
```



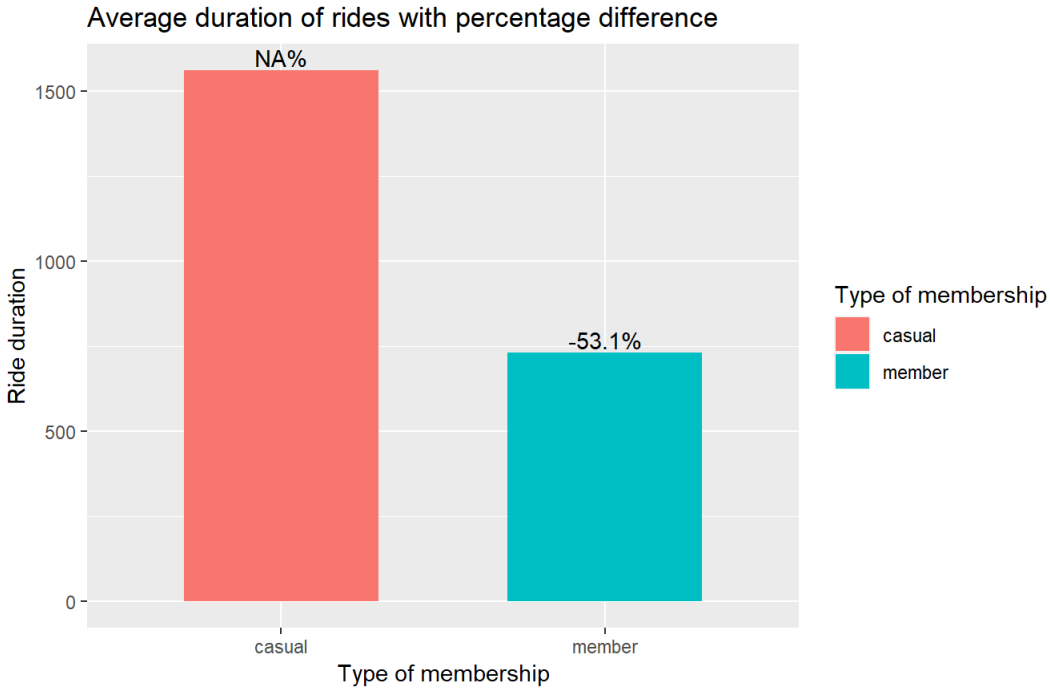
Regular users use bikes more (amount of rentals) by 71.2 on average.

Plot 2

Comparing the average duration of rides by each type of membership

```
data <- all_trips_v2 %>%
  group_by(member_casual) %>%
  summarise(average_duration = mean(ride_length)) %>%
  ungroup() %>%
  mutate(percent_change = ((average_duration - lag(average_duration)) / lag(average_duration)) * 100)

p <- ggplot(data, aes(x = member_casual, y = average_duration, fill = member_casual)) +
  geom_col(width=0.6) +
  geom_text(aes(label = paste0(round(percent_change, 1), "%")),
            vjust = -0.2) +
  labs(title = "Average duration of rides with percentage difference",
       y = "Ride duration",
       x = "Type of membership",
       caption ="On the other hand, ordinary users ride longer on average by 53.1% more",
       fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))
print(p)
```



On the other hand, ordinary users ride longer on average by 53.1% more

Creating and saving comparision of number of rides and average duration grouped by type of member and weekday

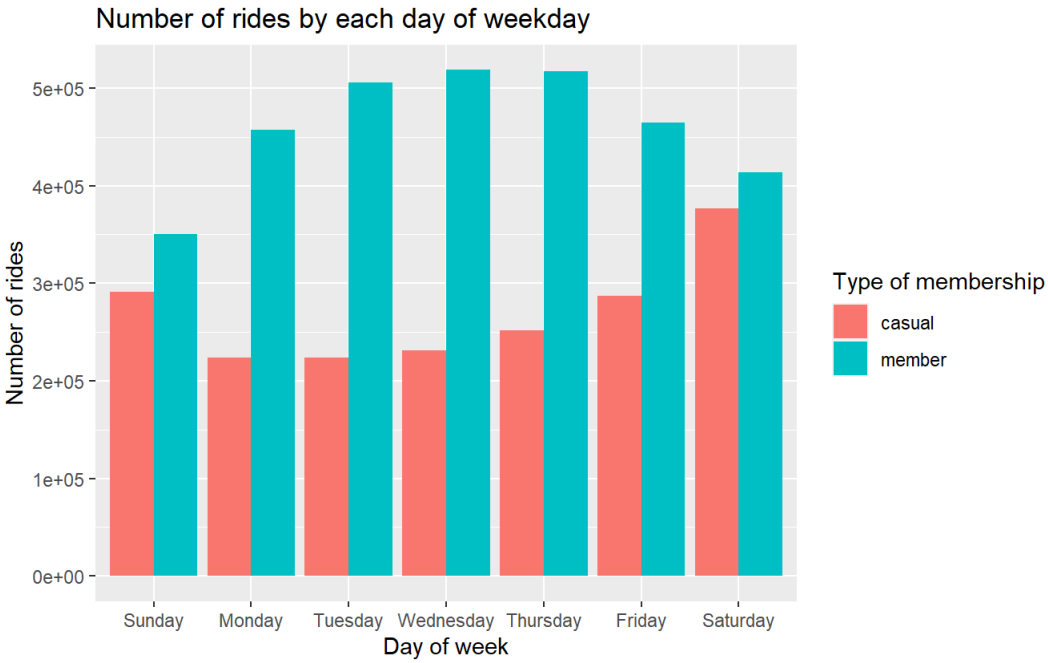
```
rides_and_mean_by_day_and_member <- all_trips_v2 %>%
  mutate(weekday = day_of_week) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
  ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

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

Plot 3

Comparing number of rides within each day of week by member type

```
rides_and_mean_by_day_and_member  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of rides by each day of weekday",
       y = "Number of rides",
       x = "Day of week",caption ="Casual users use bike sharing most often on weekends.\nRegular use most often at the week.",
       fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))
```

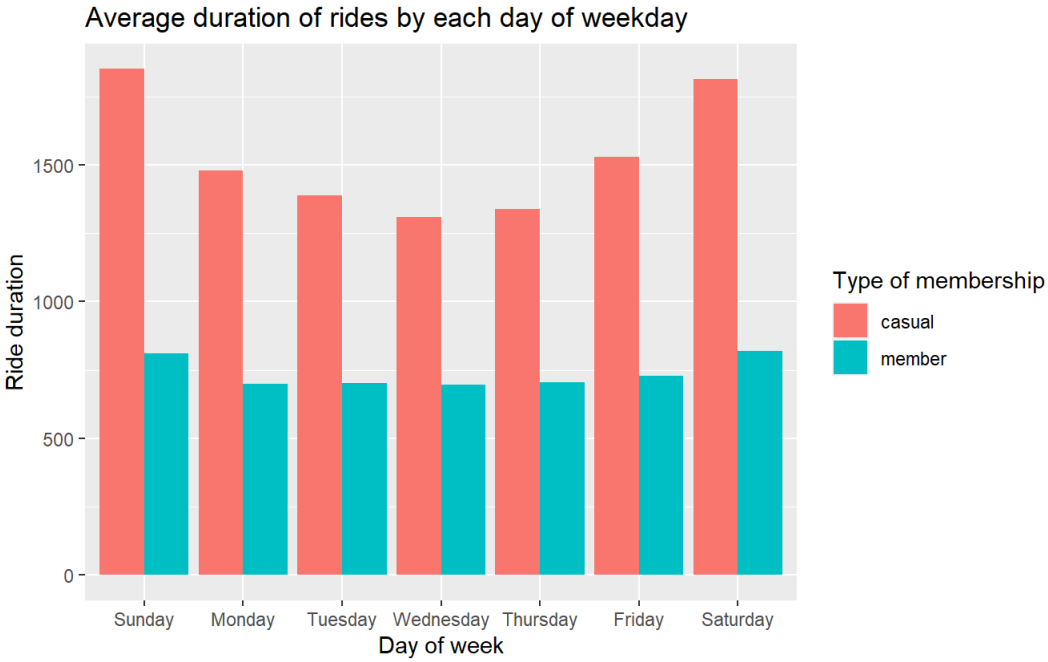


Casual users use bike sharing most often on weekends.
Regular use most often at the week.

Plot 4

Comparing mean of rides within each day of week by member type

```
rides_and_mean_by_day_and_member %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Average duration of rides by each day of weekday",
       y = "Ride duration",
       x = "Day of week",
       caption = "Casual users use bike longer on weekends than on weekdays.\nRegular users and their length of time are stable",
       fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))
```



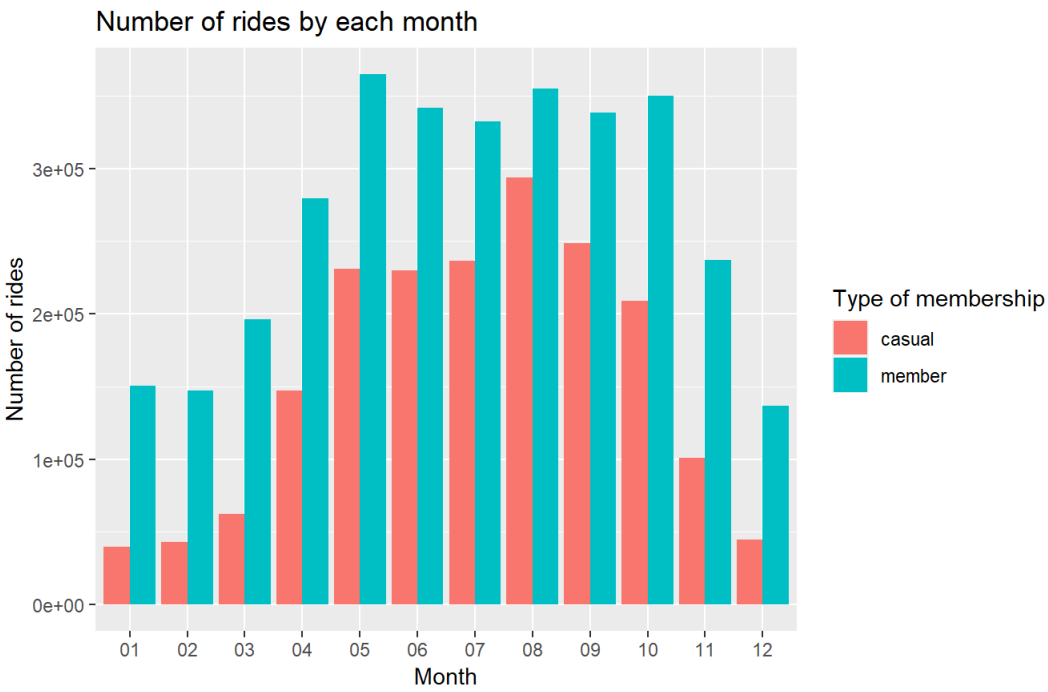
Casual users use bike longer on weekends than on weekdays.
Regular users and their length of time are stable

Plot 5

Comparing number of rides within each month by member type

```
all_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
  ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
    labs(title = "Number of rides by each month",
        y = "Number of rides",
        x = "Month",
        caption = "In both cases, it can be seen that the number of trips increases as the temperature increases",
        fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))
```

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



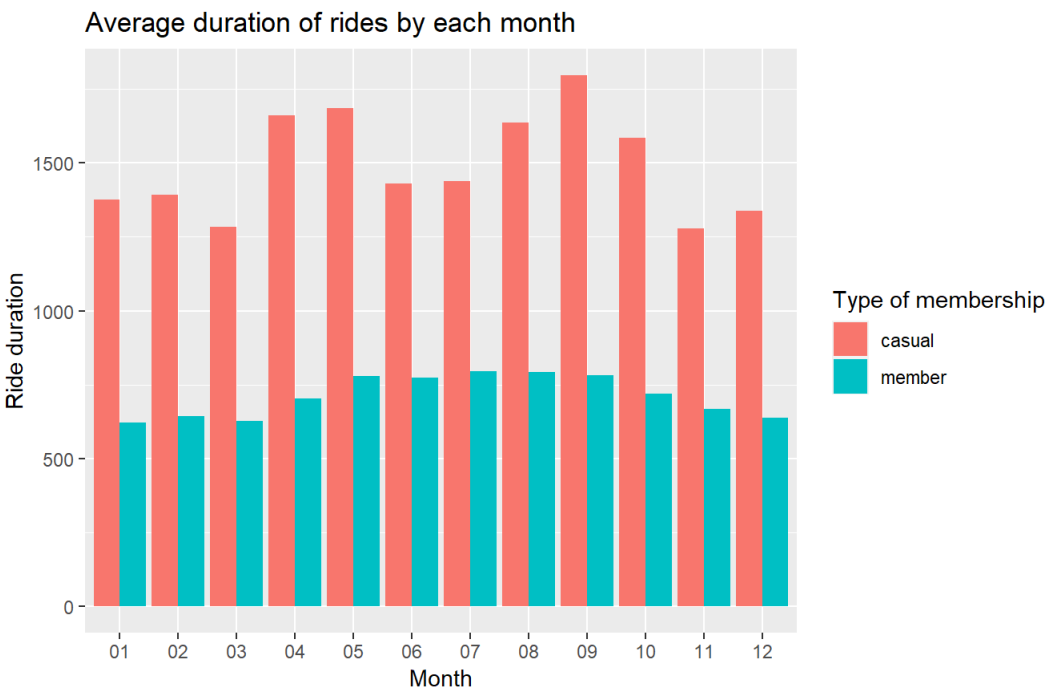
In both cases, it can be seen that the number of trips increases as the temperature incr

Plot 6

Comparing number of rides within each month by member type

```
all_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
  ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
    labs(title = "Average duration of rides by each month",
        y = "Ride duration",
        x = "Month",
        caption = "In both cases it can be seen that the duration of trips is bigger when it is warm",
        fill="Type of membership") +
  theme(plot.caption = element_text(margin=margin(t=20), size=12,hjust=0))
```

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



In both cases it can be seen that the duration of trips is bigger when it is warm

Questions and answers

Why would casual riders buy Cyclistic annual memberships ?

- To save on rides because they use them longer than the users who have annual memberships
- It may also be about convenience of use
- They may be encouraged by advertising campaigns

How can Cyclistic use digital media to influence casual riders to become members?

- The idea is to pay attention to the length of the rides and the high cost associated with it - on weekends, casual users use bike-sharing longer and more frequent. It is worth to pay attention to possibility of saving when they buy annual membership. They could take advantage by longer trips and less cost associated with this longer trips. The point is to make them more flexible on weekends and charge them less costs.
- It can be seen thar regular customers use it most often during the week. It is possible they use this solution to get to work. Maybe some of casual users also commute by bike-sharing. The idea is to encourage casual users to start commuting by bike-sharing or use it more frequent. This could led to change of thinking of casual riders to buy annual membership.
- Finally, recommendation that is based on increasing investment outlay before warm months and keep promoting during these months