Cyclistic Analysis

Michael Zegas

2023-03-13

Cyclistic case study

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

The company wants to increase the annual memberships as they make up most of the total profit compared to casual bike rentals. In order to do that, the analyst and the marketing team must get insights on how casual riders and annual members differ and why would casual riders would get an annual membership.

Ask phase

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

Dataset source:

The data has been made available by Motivate International Inc under this licence https://ride.divvybikes.com/data-license-agreement and they are available at: https://divvy-tripdata.s3.amazonaws.com/index.html. The analysis is based on a 12 month period (02/2022 - 01/2023).

Loading libraries

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(tidyr)
library(sf)
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(readr)
library(tibble)
library(geosphere)
library(oce)
```

```
## Loading required package: gsw
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(forcats)
library(scales)
```

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

Importing datasets and assigning to variables

```
#Importing
trips_02_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202202-divvy-tripdata/202202-div
```

```
## Rows: 115609 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_03_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202203-divvy-tripdata/202203-div
```

```
## Rows: 284042 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_04_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202204-divvy-tripdata/202204-div
```

```
## Rows: 371249 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_05_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202205-divvy-tripdata/202205-div
```

```
## Rows: 634858 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_06_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202206-divvy-tripdata/202206-div
```

```
## Rows: 769204 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_07_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202207-divvy-tripdata/202207-div
```

```
## Rows: 823488 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_08_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202208-divvy-tripdata/202208-div
```

```
## Rows: 785932 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_09_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202209-divvy-tripdata/202209-div
```

```
## Rows: 701339 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_10_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202210-divvy-tripdata/202210-div
```

```
## Rows: 558685 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_11_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202211-divvy-tripdata/202211-div
```

```
## Rows: 337735 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_12_2022.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202212-divvy-tripdata/202212-div
```

```
## Rows: 181806 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
trips_01_2023.csv <- read_csv("C:/Users/User/Desktop/cyclistic/Updated/202301-divvy-tripdata.csv")
```

```
## Rows: 190301 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Merging all data to one
trips_22_23.csv <- rbind(trips_02_2022.csv, trips_03_2022.csv, trips_04_2022.csv, trips_05_2022.csv, tri

head(trips_22_23.csv)
```

```
## # A tibble: 6 x 13
##   ride_id        ridea~1 started_at          ended_at            start~2 start~3
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 E1E065E7ED285~ classi~ 2022-02-19 18:08:41 2022-02-19 18:23:56 State ~ TA1305~
## 2 1602DCDC5B30F~ classi~ 2022-02-20 17:41:30 2022-02-20 17:45:56 Halste~ TA1309~
## 3 BE7DD2AF4B55C~ classi~ 2022-02-25 18:55:56 2022-02-25 19:09:34 State ~ TA1305~
## 4 A1789BDF84441~ classi~ 2022-02-14 11:57:03 2022-02-14 12:04:00 Southp~ 13235
## 5 07DE78092C62F~ classi~ 2022-02-16 05:36:06 2022-02-16 05:39:00 State ~ TA1305~
## 6 9A2F204F04AB7~ classi~ 2022-02-07 09:51:57 2022-02-07 10:07:53 St. Cl~ 13016
## # ... with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names 1: rideable_type,
## #   2: start_station_name, 3: start_station_id
```

Process phase

```
#Creating a new column with the mean ride duration
trips_22_23.csv$duration <- difftime(trips_22_23.csv$ended_at, trips_22_23.csv$started_at, units = "mins

#Clean data by removing n/a and assigning to new variable
trips_clean <- drop_na(trips_22_23.csv)

#Cleaning ride duration
trips_clean$duration <- round(trips_clean$duration, digits = 1)

#Separate datetime column in columns: date, day, year, day_of_week, month name
trips_clean$date <- as.Date(trips_clean$started_at)
trips_clean$day <- format(as.Date( trips_clean$date), "%d")
trips_clean$year <- format(as.Date(trips_clean$date), "%Y")
trips_clean$day_of_week <- format(as.Date(trips_clean$date), "%A")
trips_clean$month_name <- format(as.Date(trips_clean$date), "%b")

#Changing the language from default language to English
lookup_table <- c("Δ   " = "Monday",
                  "Τ  " = "Tuesday",
                  "Τ   " = "Wednesday",
                  "Π   " = "Thursday",
                  "Π    " = "Friday",
                  "Σ   " = "Saturday",
                  "Κ   " = "Sunday")

trips_clean$day_of_week <- lookup_table[trips_clean$day_of_week]


lookup_table2 <- c("Ι " = "January",
                  "Φ " = "February",
                  "Μ " = "March",
                  "Α " = "April",
                  "Μ " = "May",
                  "Ι  " = "June",
                  "Ι  " = "July",
                  "Α " = "August",
```

```
                  "Σ " = "September",
                  "Ο " = "October",
                  "Ν " = "November",
                  "Δ " = "December")

trips_clean$month_name <- lookup_table2[trips_clean$month_name]

#Right order of weekdays
weekday_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
```

Analyze phase

```
#Finding the mean ride duration for all rides and then for each user type
avg_dur_total <- mean(trips_clean$duration)
options(scipen = 999)
avg_dur_total
```

```
## Time difference of 16.96543 mins
```

```
avg_dur_per_user <- trips_clean %>%
  group_by(member_casual) %>%
  summarise(avg_duration = mean(duration))

avg_dur_per_user
```

```
## # A tibble: 2 x 2
##   member_casual avg_duration
##   <chr>         <drtn>
## 1 casual        23.81604 mins
## 2 member        12.39751 mins
```

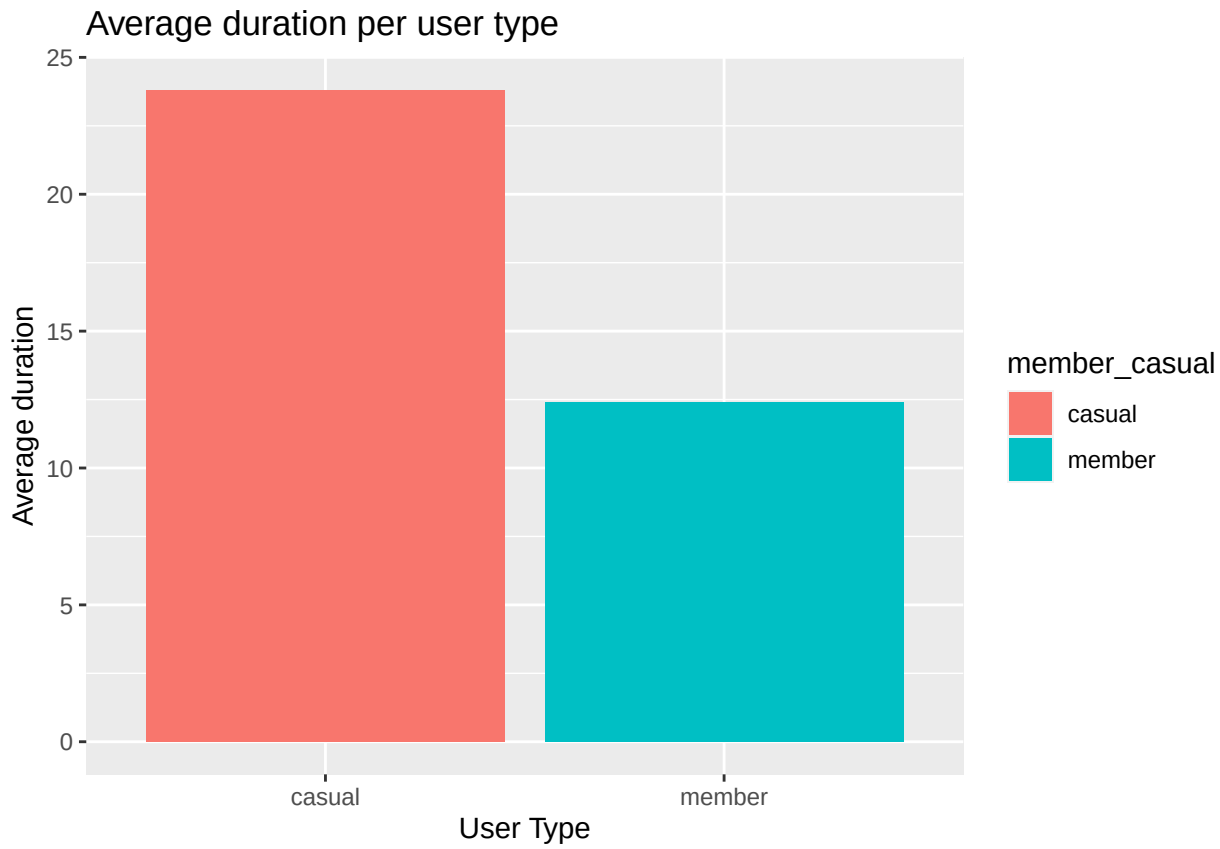
```
#Making a distance traveled column

trips_clean$ride_distance <- distGeo(matrix(c(trips_clean$start_lng, trips_clean$start_lat), ncol = 2),
trips_clean$ride_distance <- trips_clean$ride_distance/1000
trips_clean$ride_distance <- round(trips_clean$ride_distance, digits = 1)

#Finding the mean ride distance for all users and for each user type
avg_distance_total <- mean(trips_clean$ride_distance)

avg_distance_per_user <- trips_clean %>%
  group_by(member_casual) %>%
  summarise(avg_dist = mean(ride_distance))

avg_distance_per_user
```

```
## # A tibble: 2 x 2
##   member_casual avg_dist
##   <chr>            <dbl>
## 1 casual            2.15
## 2 member            2.06
```

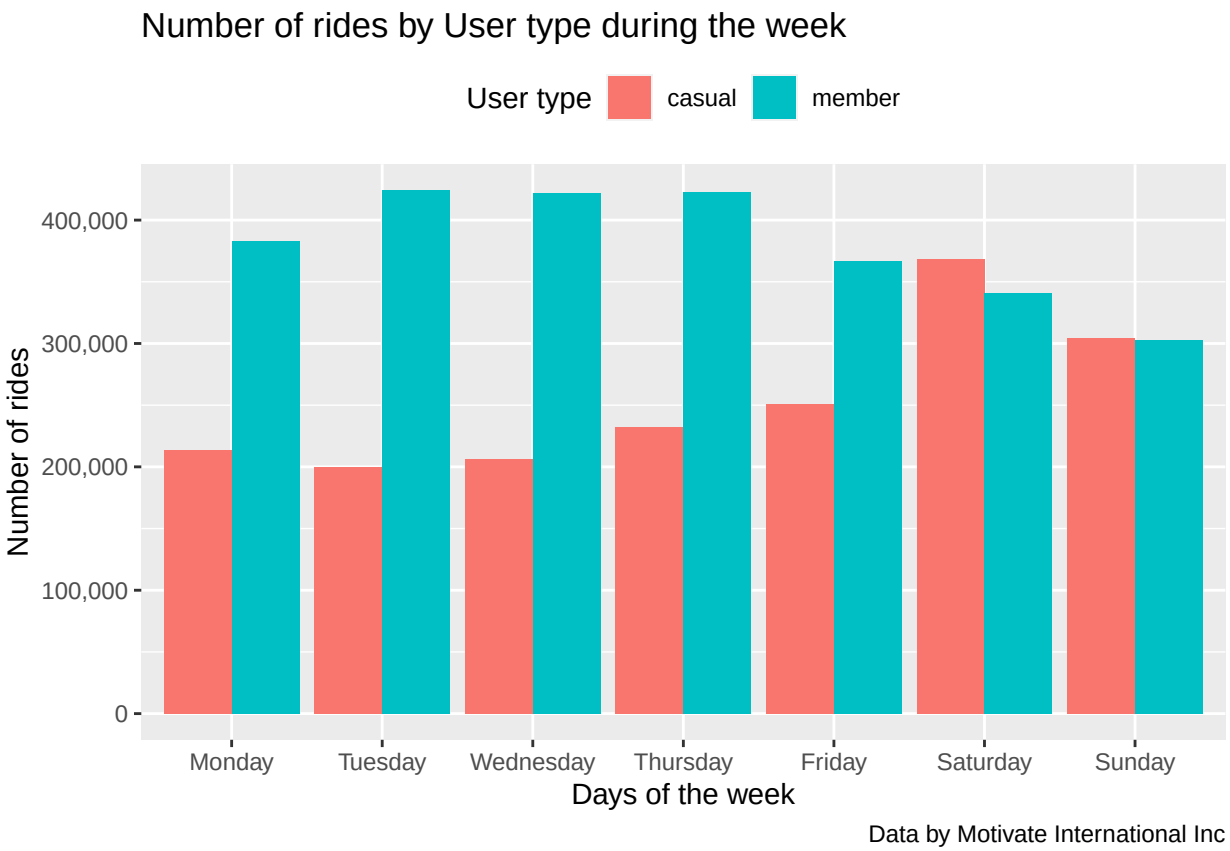
Visualizing data

```
#Duration graph
ggplot(avg_dur_per_user, aes(x = member_casual, y = avg_duration, fill= member_casual)) +
  geom_col() +
  labs(title = "Average duration per user type", x= "User Type", y= "Average duration")
```

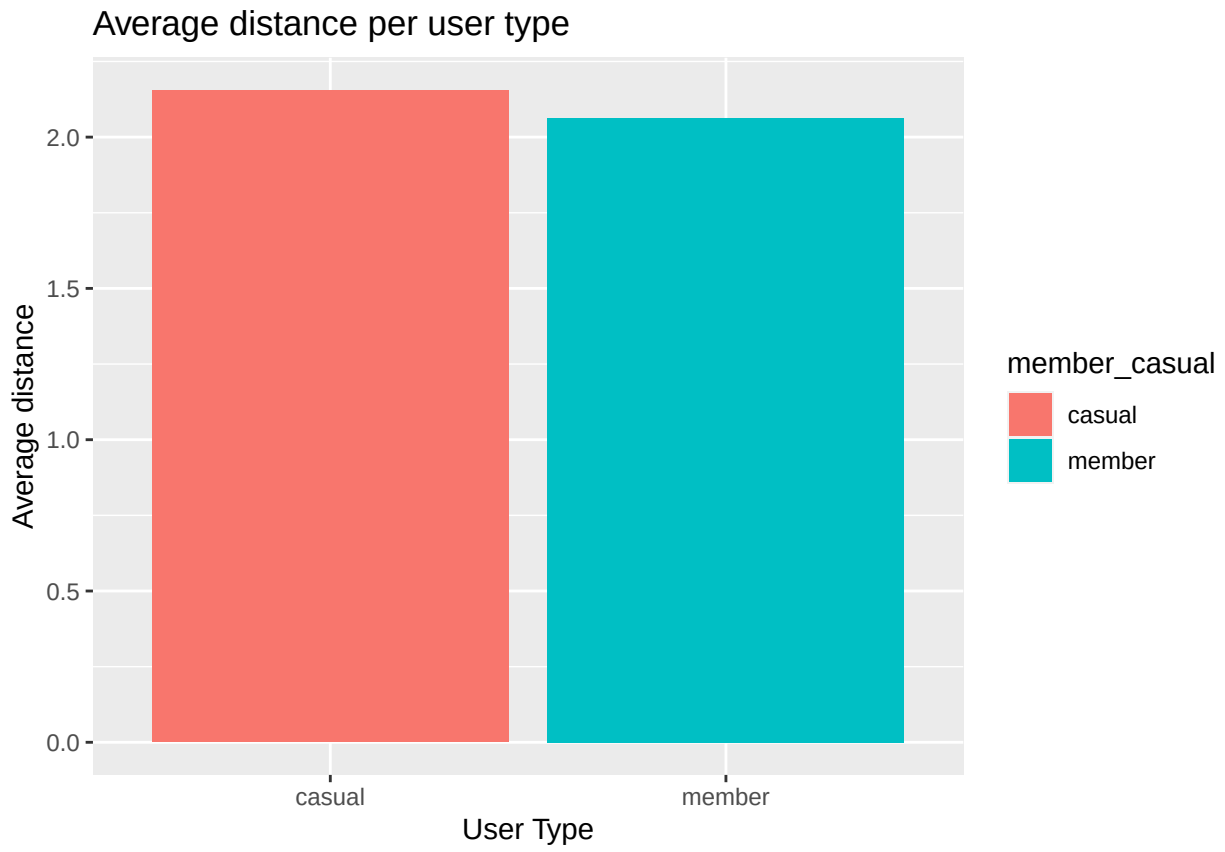
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



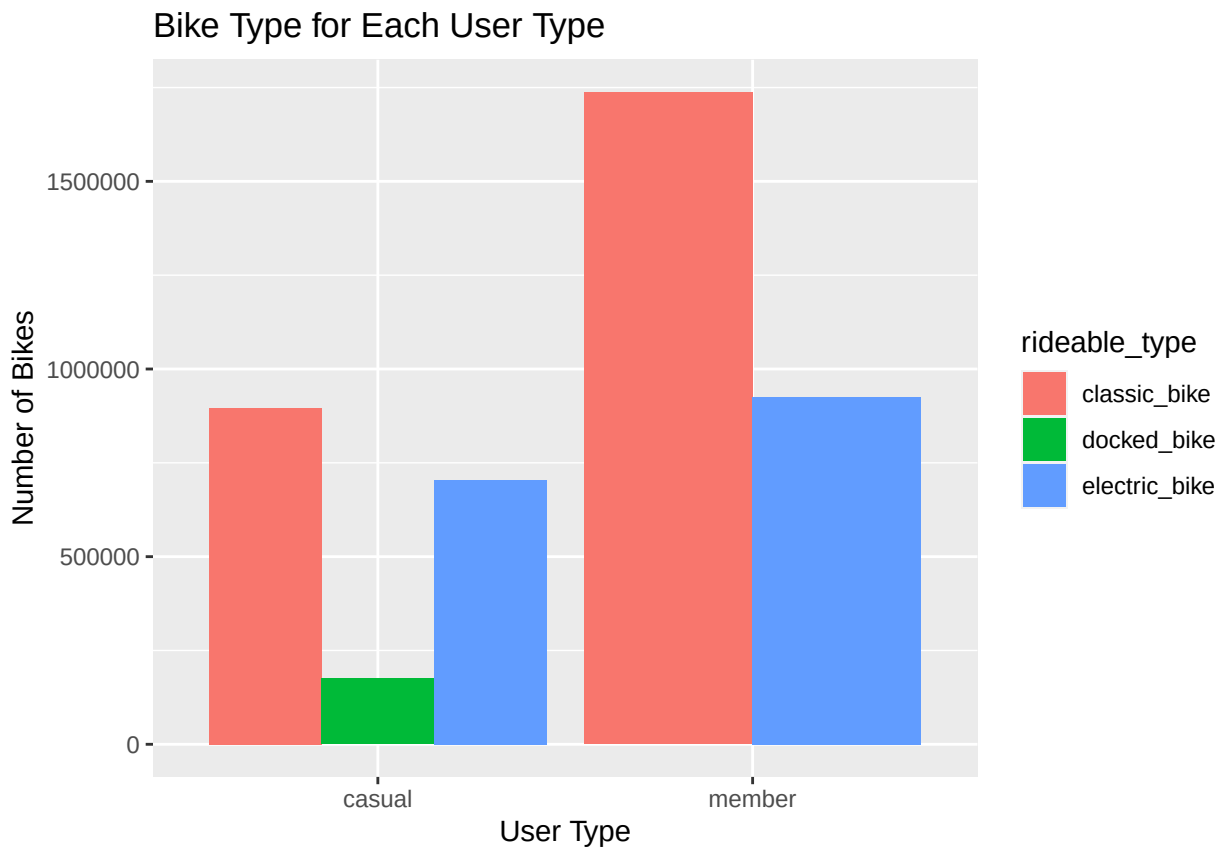

```
#Duration for weekdays graph
trips_clean %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(duration()),.groups = 'drop') %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = factor(day_of_week, weekday_order), y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = comma) +
  labs(title = "Number of rides by User type during the week",x="Days of the week",y="Number of rides",
  theme(legend.position="top")
```




```
#distance graph
ggplot(avg_distance_per_user, aes(x = member_casual, y = avg_dist, fill= member_casual)) +
  geom_col() +
  labs(title = "Average distance per user type", x= "User Type", y= "Average distance")
```




```
#bike type graph
trips_clean %>%
  ggplot(aes(x = member_casual, fill = rideable_type)) +
  geom_bar(position = "dodge") +
  labs(x = "User Type", y = "Number of Bikes") +
  ggtitle("Bike Type for Each User Type")
```



Conclusion

As shown by the analysis, there is not a significance difference in the average distance traveled between casual users and members. Members seem to have a similar number of rides during the weekdays but with less mean duration than the casual users. That signifies that casual users ride mostly for leisure than members who they seem to use bikes more like an alternative to public transport. There is not a significant difference in the rideable type for casual users, but among members there is a clear preference in classic bikes. The marketing team could focus on promoting the bike usage as an everyday transport solution. The short mean ride duration of member signifies that using bikes is a fast way to relocate without the need of waiting for public transport, searching for a parking space, and rely on other means of transport generally.