Citi bike - Cyclistic - Jan'22 to Dec'22

Parthebhan Pari

2023-11-13

Introduction

In 2016, Cyclistic launched a successful bike-share oering. Since then, the program has grown to a eet 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 toany other station in the system anytime.

Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the exibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Ask

Three questions will guide the future marketing program:

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

#Data sources

User data from the past 12 months, January 2022 - December 2022 has been made available. Each data set is in csv format and details every ride logged by Cyclistic customers. This data has been made publicly available via license by Motivate International Inc. and the city of Chicago available here. All user's personal data has been scrubbed for privacy.

Prepare

Load the necessary libraries that will be utilized for the project

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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
```

```r
library(dplyr)
library(janitor)
```

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

```r
library(skimr)
library(ggplot2)
library(lubridate)
```

load datasets into RStudio

```r
trip01 <- read.csv("202201-divvy-tripdata.csv")
trip02 <- read.csv("202202-divvy-tripdata.csv")
trip03 <- read.csv("202203-divvy-tripdata.csv")
trip04 <- read.csv("202204-divvy-tripdata.csv")
trip05 <- read.csv("202205-divvy-tripdata.csv")
trip06 <- read.csv("202206-divvy-tripdata.csv")
trip07 <- read.csv("202207-divvy-tripdata.csv")
trip08 <- read.csv("202208-divvy-tripdata.csv")
trip09 <- read.csv("202209-divvy-tripdata.csv")
trip10 <- read.csv("202210-divvy-tripdata.csv")
trip11 <- read.csv("202211-divvy-tripdata.csv")
trip12 <- read.csv("202212-divvy-tripdata.csv")
```

Combine every dataset to consolidate analysis

```r
citi_bike_2022 <- rbind(trip01,trip02,trip03,trip04,trip05,trip06,trip07,trip08,trip09,trip10,trip11,tri
```

View newly created dataset

```r
colnames(citi_bike_2022)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

Prepare

Firstly remove all the irrelevent columns that won't be used for analysis

```
citi_bike_2022 <- citi_bike_2022 %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, start_station_id,end_station_id, end_station_name))
```

Review of the data and its parameters.

```
colnames(citi_bike_2022) #List of column names
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "member_casual"
```

```
nrow(citi_bike_2022) #How many rows are in data frame?
```

```
## [1] 5667717
```

```
dim(citi_bike_2022) #Dimensions of the data frame?
```

```
## [1] 5667717       6
```

```
head(citi_bike_2022, 6) #See the first 6 rows of data frame.  Also tail(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 C2F7DD78E82EC875 electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44
## 2 A6CF8980A652D272 electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17
## 3 BD0F91DFF741C66D  classic_bike 2022-01-25 04:53:40 2022-01-25 04:58:01
## 4 CBB80ED419105406  classic_bike 2022-01-04 00:18:04 2022-01-04 00:33:00
## 5 DDC963BFDDA51EEA  classic_bike 2022-01-20 01:31:10 2022-01-20 01:37:12
## 6 A39C6F6CC0586C0B  classic_bike 2022-01-11 18:48:09 2022-01-11 18:51:31
##              start_station_name member_casual
## 1      Glenwood Ave & Touhy Ave        casual
## 2      Glenwood Ave & Touhy Ave        casual
## 3 Sheffield Ave & Fullerton Ave        member
## 4      Clark St & Bryn Mawr Ave        casual
## 5   Michigan Ave & Jackson Blvd        member
## 6         Wood St & Chicago Ave        member
```

```
str(citi_bike_2022) #See list of columns and data types (numeric, character, etc)
```

```
## 'data.frame':    5667717 obs. of  6 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
```

```r
summary(citi_bike_2022) #inspect the date and its dimensions before moving onto cleaning
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5667717     Length:5667717     Length:5667717     Length:5667717
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##  start_station_name member_casual
##  Length:5667717     Length:5667717
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

Additional columns must be created for date and time.

```r
citi_bike_2022$date <- as.Date(citi_bike_2022$started_at)
citi_bike_2022$month <- format(as.Date(citi_bike_2022$date), "%m")
citi_bike_2022$day <- format(as.Date(citi_bike_2022$date), "%d")
citi_bike_2022$year <- format(as.Date(citi_bike_2022$date), "%Y")
citi_bike_2022$day_of_week <- format(as.Date(citi_bike_2022$date), "%A")
citi_bike_2022$time <- format(citi_bike_2022$started_at, format= "%H:%M")
citi_bike_2022$time <- as.POSIXct(citi_bike_2022$time, format= "%H:%M")
```

Calculated filed that shows the time of each unique ride

```r
citi_bike_2022$ride_length <- (as.double(difftime(citi_bike_2022$ended_at,citi_bike_2022$started_at))) /
```

Check data structure. Confirm data types for time/date

```r
str(citi_bike_2022) #confirm data type is double
```

```
## 'data.frame':    5667717 obs. of  13 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
##  $ date              : Date, format: "2022-01-13" "2022-01-10" ...
##  $ month             : chr  "01" "01" "01" "01" ...
##  $ day               : chr  "13" "10" "25" "04" ...
##  $ year              : chr  "2022" "2022" "2022" "2022" ...
##  $ day_of_week       : chr  "Thursday" "Monday" "Tuesday" "Tuesday" ...
##  $ time              : POSIXct, format: NA NA ...
##  $ ride_length       : num  2.95 4.35 4.35 14.93 6.03 ...
```

Alter data type for time

```
citi_bike_2022$ride_length <- as.numeric(as.character(citi_bike_2022$ride_length))
```

Remove all blank entries from the dataset

```
citi_bike_2022<- citi_bike_2022[!(citi_bike_2022$start_station_name == "HQ QR" | citi_bike_2022$ride_le
```

Observe the newly created column for the backup dataset

```
summary(citi_bike_2022$ride_length)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00     5.82    10.28    19.45    18.47 41387.25
```

Analyze data

#Calculating the mean, median, max, min - figures to determine statisical spread of membership type

```
aggregate(citi_bike_2022$ride_length ~ citi_bike_2022$member_casual, FUN = mean)
```

```
##   citi_bike_2022$member_casual citi_bike_2022$ride_length
## 1                       casual                   29.14572
## 2                       member                   12.71401
```

```
aggregate(citi_bike_2022$ride_length ~ citi_bike_2022$member_casual, FUN = median)
```

```
##   citi_bike_2022$member_casual citi_bike_2022$ride_length
## 1                       casual                  13.000000
## 2                       member                   8.833333
```

```
aggregate(citi_bike_2022$ride_length ~ citi_bike_2022$member_casual, FUN = max)
```

```
##   citi_bike_2022$member_casual citi_bike_2022$ride_length
## 1                       casual                   41387.25
## 2                       member                    1559.90
```

```
aggregate(citi_bike_2022$ride_length ~ citi_bike_2022$member_casual, FUN = min)
```

```
##   citi_bike_2022$member_casual citi_bike_2022$ride_length
## 1                       casual                          0
## 2                       member                          0
```

Order day's of week within new dataset for future use

```
citi_bike_2022$day_of_week <- ordered(citi_bike_2022$day_of_week, levels=c("Sunday", "Monday", "Tuesday
```

Create a weekday field as well as view column specifics

```
citi_bike_2022  %>%
  mutate(day_of_week = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, day_of_week ) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n())
```

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

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides
##    <chr>         <ord>                 <int>
##  1 casual        Sun                  389011
##  2 casual        Mon                  277671
##  3 casual        Tue                  263731
##  4 casual        Wed                  274354
##  5 casual        Thu                  309327
##  6 casual        Fri                  334698
##  7 casual        Sat                  473185
##  8 member        Sun                  387208
##  9 member        Mon                  473335
## 10 member        Tue                  518618
## 11 member        Wed                  523867
## 12 member        Thu                  532255
## 13 member        Fri                  467083
## 14 member        Sat                  443274
```

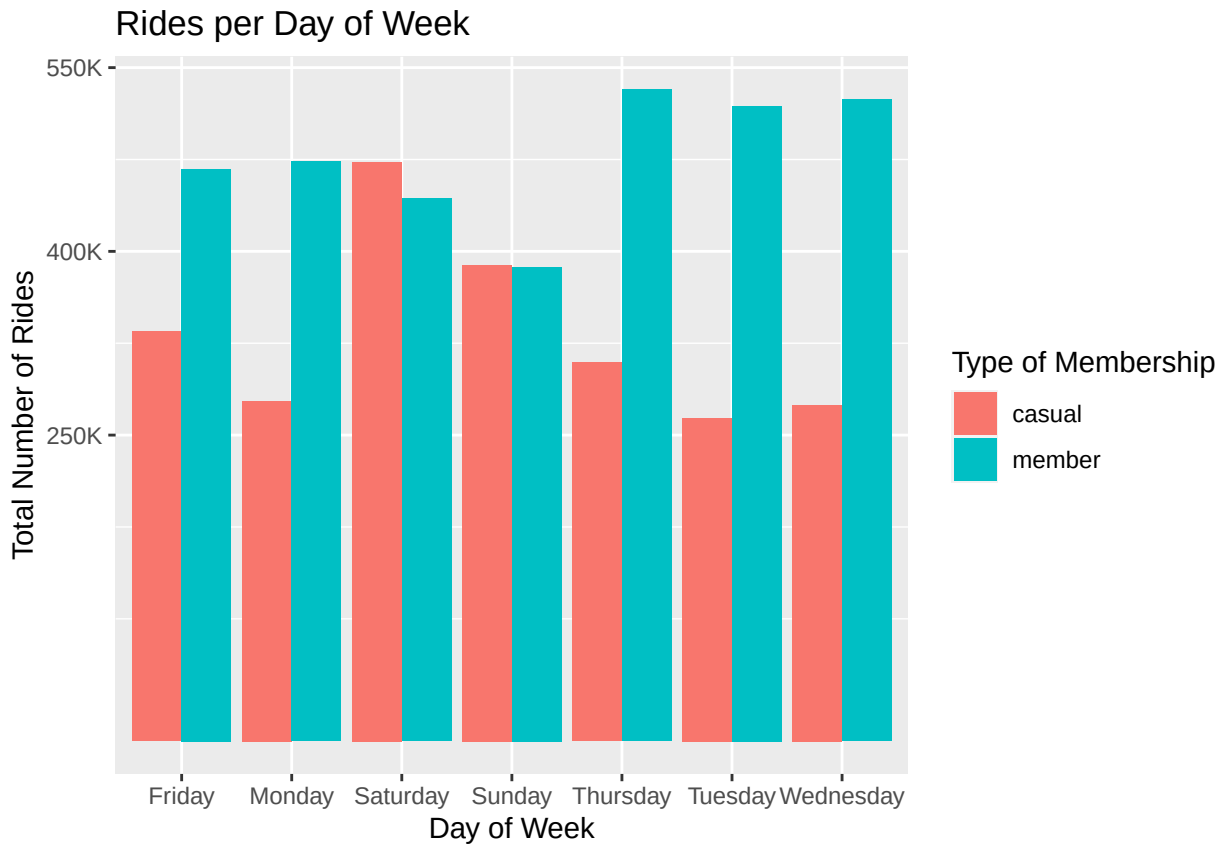
Data Visualiation's

Ride count in days of a weekdays / weekend (Casual vs member users)

```
citi_bike_2022$day_of_week  <- format(as.Date(citi_bike_2022$date), "%A")

citi_bike_2022 %>%                              #total rides broken down by weekday
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) + geom_col(position = "dodge"
  labs(x='Day of Week', y='Total Number of Rides', title='Rides per Day of Week', fill = 'Type of Membe
  scale_y_continuous(breaks = c(250000, 400000, 550000), labels = c("250K", "400K", "550K"))
```

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




```
View(citi_bike_2022)
```

```
citi_2020_weekday_weekends <- citi_bike_2022 %>%
  group_by(member_casual,day_type = ifelse(day_of_week %in% c("Saturday", "Sunday"), "Weekend", "Weekday
  summarise(total_rides = n()) %>%
  mutate(percentage= total_rides*100/sum(total_rides)) %>%
  arrange(day_type)
```

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

```
print(citi_2020_weekday_weekends)
```

```
## # A tibble: 4 x 4
## # Groups:   member_casual [2]
##   member_casual day_type total_rides percentage
##   <chr>         <chr>          <int>      <dbl>
## 1 casual        Weekday      1459781       62.9
## 2 member        Weekday      2515158       75.2
## 3 casual        Weekend       862196       37.1
## 4 member        Weekend       830482       24.8
```

Key Finding:

The rides per day of week show casual riders peak on the Saturday and Sunday while members peak Monday through Friday. This indicates members mainly use the bikes for their commutes and not leisure.

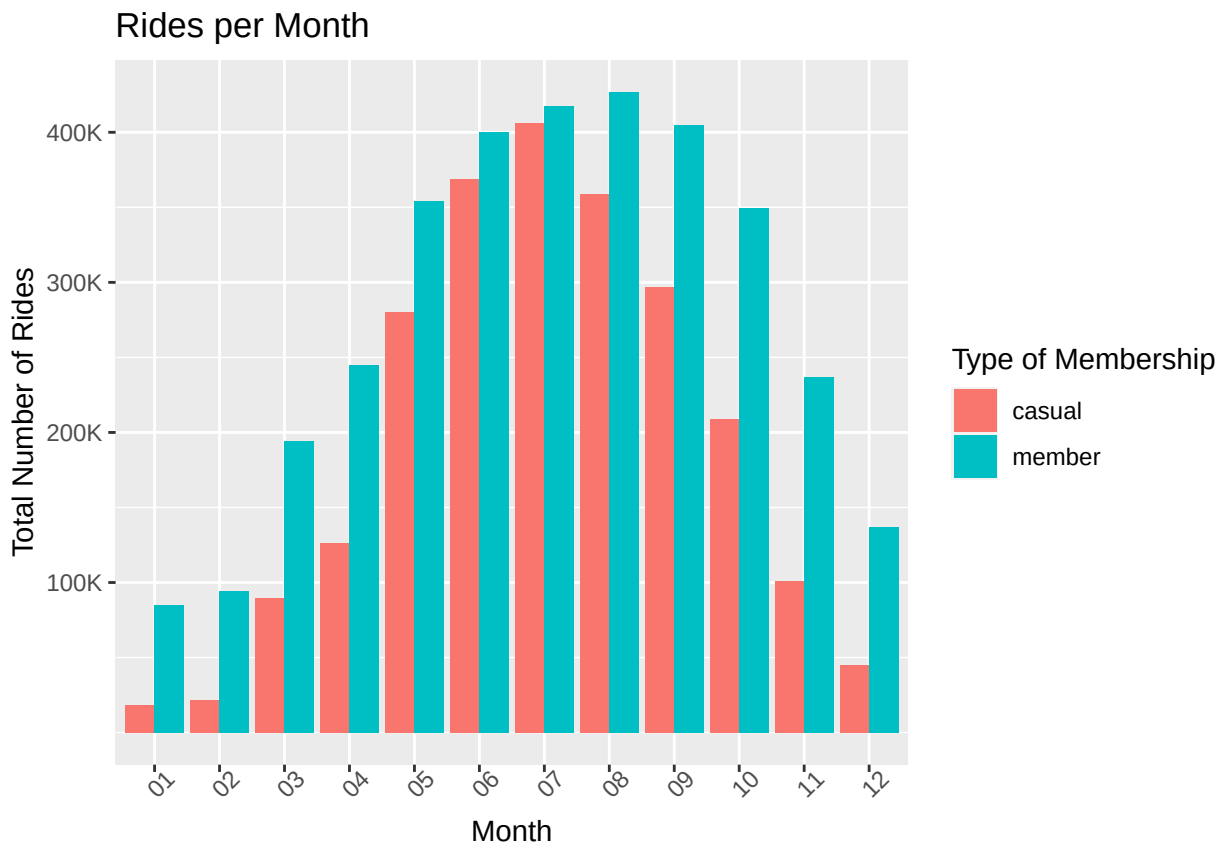
Number of rides per month (Casual vs member users)

```
citi_2020_month <- citi_bike_2022 %>%
  group_by(member_casual,month) %>%
  summarise(n=n())
```

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

```
citi_bike_2022 %>%   #total rides broken down by month
  group_by(member_casual, month) %>%
  summarise(total_rides = n(),`average_duration_(mins)` = mean(ride_length)) %>%
  arrange(member_casual) %>%
  ggplot(aes(x=month, y=total_rides, fill = member_casual)) + geom_col(position = "dodge") +
  labs(x= "Month", y= "Total Number of Rides", title = "Rides per Month", fill = "Type of Membership") +
  scale_y_continuous(breaks = c(100000, 200000, 300000, 400000), labels = c("100K", "200K", "300K", "400
```

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



```r
print(citi_2020_month)
```

```
## # A tibble: 24 x 3
## # Groups:   member_casual [2]
##    member_casual month      n
##    <chr>         <chr>  <int>
##  1 casual        01     18520
##  2 casual        02     21416
##  3 casual        03     89880
##  4 casual        04    126417
##  5 casual        05    280414
##  6 casual        06    369044
##  7 casual        07    406046
##  8 casual        08    358917
##  9 casual        09    296694
## 10 casual        10    208988
## # i 14 more rows
```

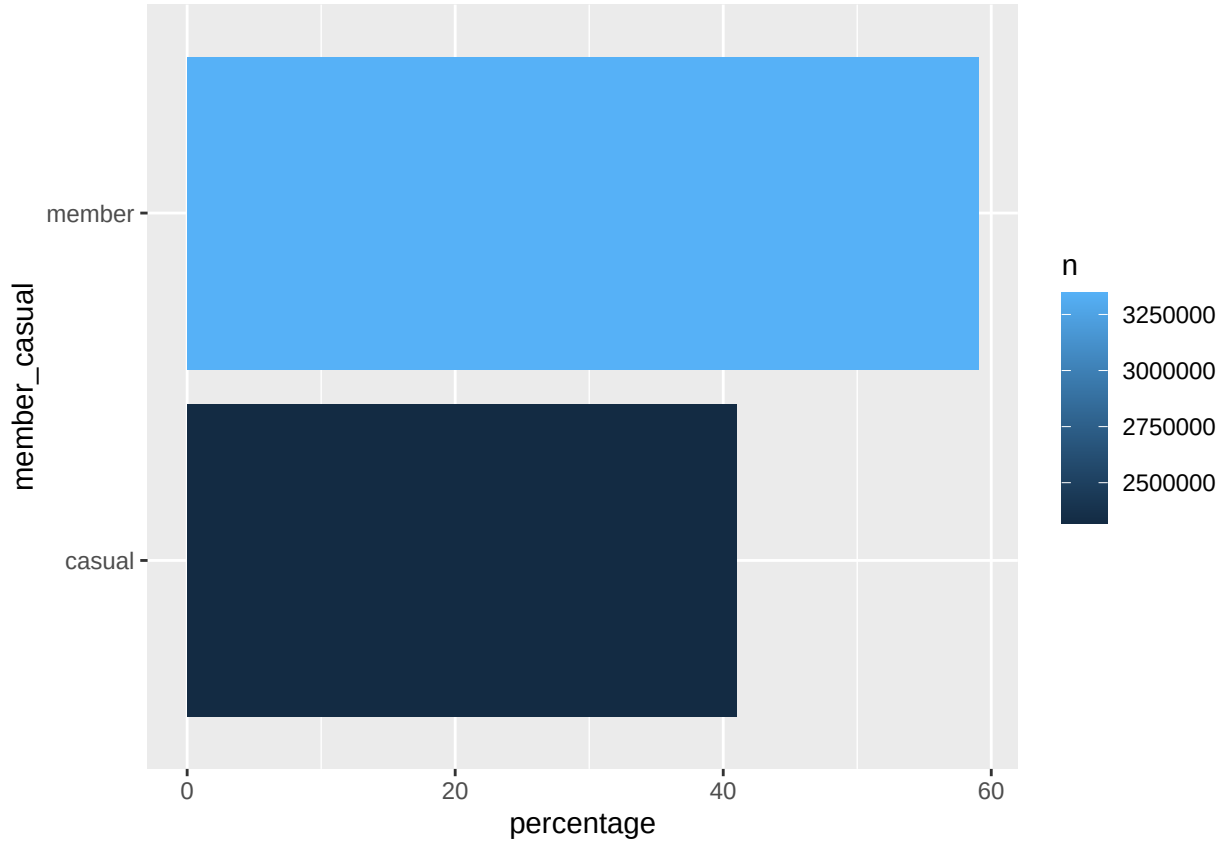
Key Finding:

The rides per month show that casual riders were a lot more active during the summer months than the long-term. Conversly, the winter months show very little activity on the part of the casual users. The long-term users are more active in the winter and spring months.

To find memeber_casual count

```r
citi_2020_Member_causal <-citi_bike_2022 %>%
  group_by(member_casual) %>%
  summarise(n = n()) %>%
  mutate(percentage= n*100/sum(n)) %>%
  arrange(percentage, n)

  citi_2020_Member_causal %>%
  ggplot(aes(x = percentage, y = member_casual, fill = n)) + geom_col( )
```




```
print(citi_2020_Member_causal)
```

```
## # A tibble: 2 x 3
##   member_casual       n percentage
##   <chr>           <int>      <dbl>
## 1 casual        2321977       41.0
## 2 member        3345640       59.0
```

#To find member_casual Vs rideable_type

```
# Percentage

citi_2020_Member_causal_rideable <-citi_bike_2022 %>%
  group_by(member_casual,rideable_type) %>%
  summarise(n = n()) %>%
  mutate(percentage= n*100/sum(n)) %>%
  arrange(percentage, n)
```

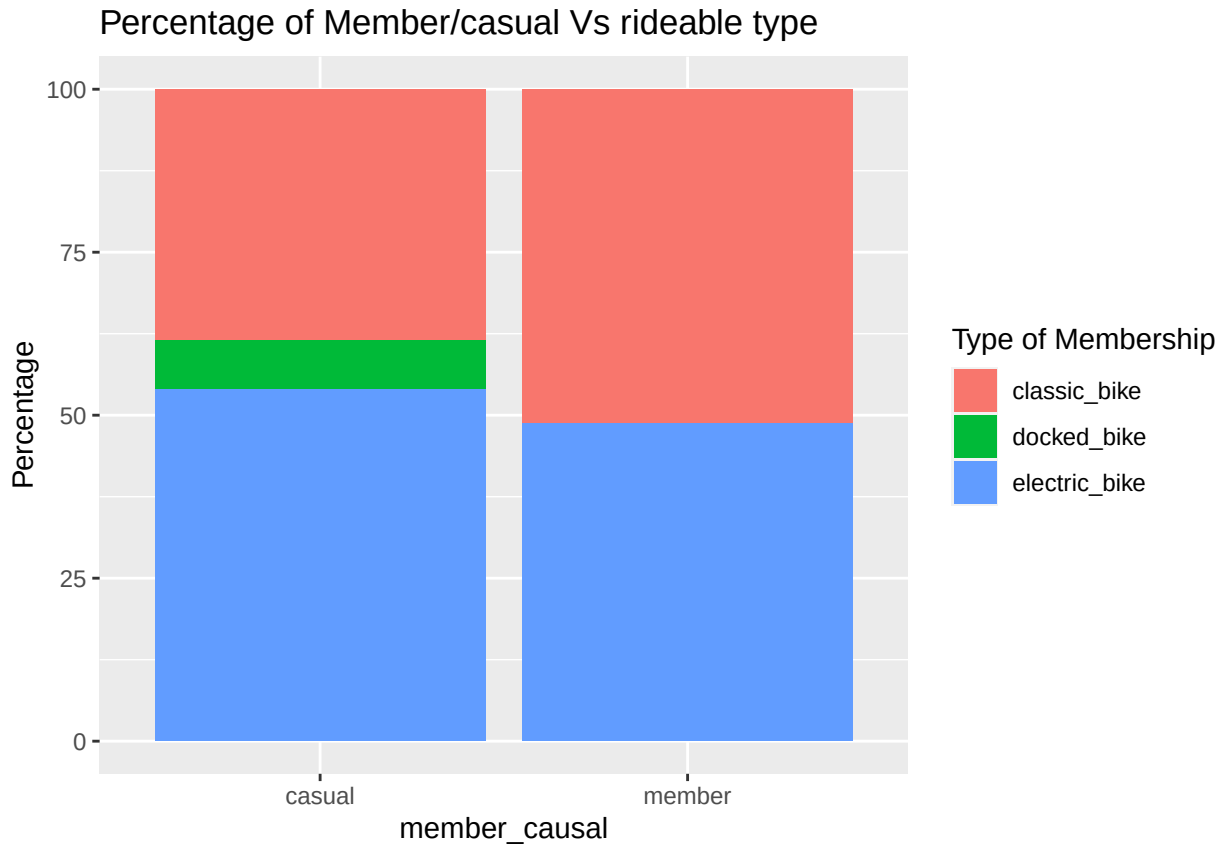
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
print(citi_2020_Member_causal_rideable)
```

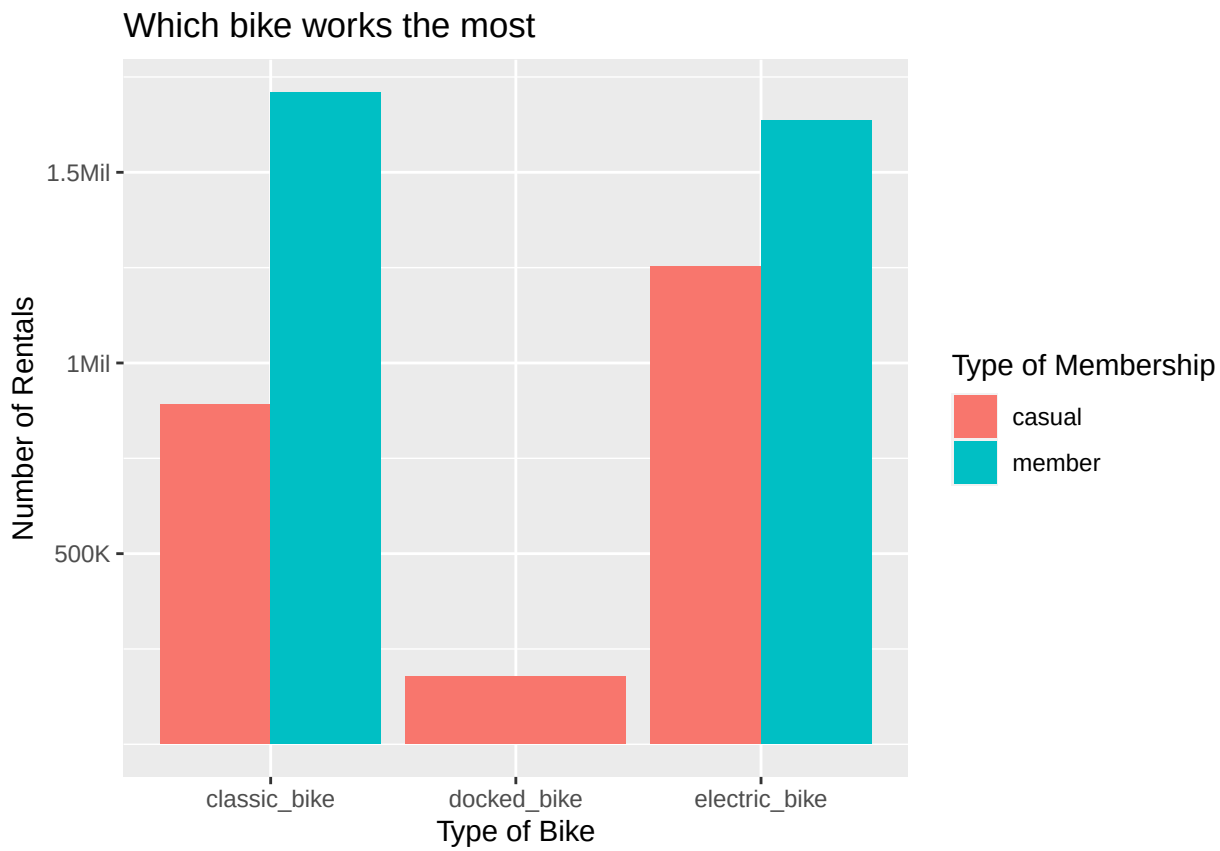
```
## # A tibble: 5 x 4
```

```
## # Groups:   member_casual [2]
##   member_casual rideable_type       n percentage
##   <chr>         <chr>           <int>      <dbl>
## 1 casual        docked_bike    177474       7.64
## 2 casual        classic_bike   891443      38.4
## 3 member        electric_bike 1635897      48.9
## 4 member        classic_bike  1709743      51.1
## 5 casual        electric_bike 1253060      54.0
```

```
citi_2020_Member_causal_rideable %>%
  ggplot(aes(x = member_casual, y = percentage, fill = rideable_type)) + geom_col( )+
  labs(x='member_causal', y='Percentage', title='Percentage of Member/casual Vs rideable type ', fill =
```




```
citi_bike_2022 %>%    #looking at breakdown of bike types rented
  ggplot(aes(x = rideable_type, fill = member_casual)) + geom_bar(position = "dodge") +
  labs(x= 'Type of Bike', y='Number of Rentals', title='Which bike works the most', fill = 'Type of Meml
  scale_y_continuous(breaks = c(500000, 1000000, 1500000), labels = c("500K", "1Mil", "1.5Mil"))
```



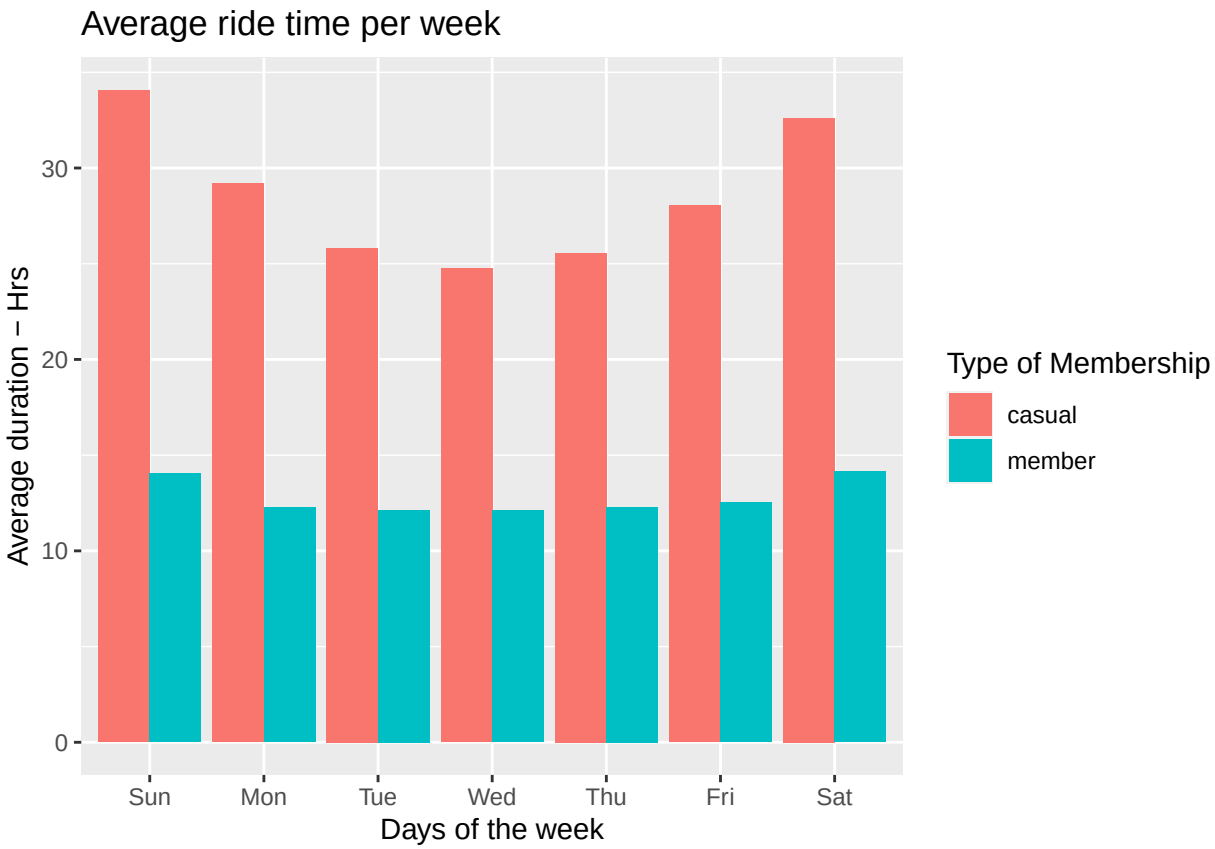
Key Finding:

The breakdown of which type of bike is the most popular among either type of user. Showing among the two types of bikes classic and electric. both types of memberships prefer using the classic bike more so than the electric bike. The long-term memebrs are also seen to be of the two types favours the classic bike.

Find the average time spent riding by each membership type per individual day

```
citi_bike_2022 %>%
  mutate(day_of_week = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(),average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") + labs(x='Days of the week', y='Average duration - Hrs', title='Average
```

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



Key Finding:

The average ride time shows a stark difference between the casuals and members. Casuals overall spend more time using the service than their full time member counter-parts.

Share

```
Casual users tended to ride more so in the warmer months of Chicago, namely June- August. Their particip
To further that the Casual demographic spent on average a lot longer time per ride than their long-term
The days of the week also further shows that causal riders prefer to use the service during the weekends
Long term riders tended to stick more so to classic bikes as opposed to the docked or electric bikes.
```

Act

This report recommends the following:

```
Introducing plans thats may be more appealing to casuals for the summer months. This marketing should be
The casual users might be more interested in a memebrship option that allows for per-use balance card. A
Membership rates specifically for the warmer months as well as for those who only ride on the weekends w
```

Things to Consider

Additional points that were not examined

The report understands the scope of this analysis is extremely limited and because of that fact, additional data, as well as data points may have been able to contribute to this report offering an even more granular analysis. The following are data points that could have enhanced the report:

Age and gender: This would add a dynamic to whether or not customers are being targeted across demograpic lines. Is the existing marketing effective? Is there potential for more inclusive targeting? Pricing structure: THe actual pricing plans data was not provided and would give further insight to which plans are the most popular and by (how much) when comparing them. It would also be effective to understanding the spending behaviour of casual user. Household income data: Pinpointing the average income of the long-term memebrs as compared to the casual counter-parts would allow for further analysis of what is the typical economic standing of each type of member, as well as providing the ability to analysis overall price sensitivity between the two different membership types.