Case Study: Bike Sharing

Daniel Amos O.

2022-03-22

Case Study: How Does a Bike-Share Navigate Speedy Success?

Scenario

I'm 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.

Ask:

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? We will be using R for our analysis to get a variety of insights.

Business Task:

Analyze user behaviors between annual members and casual riders. Make recommendations regarding how to convert casual riders into annual members.

Set up Environment

Simplify working directory to make data calls easier and fast Load the required libraries 'tidyverse', 'lubridate', 'dplyr'

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate) #helps to wrangle date attribute
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
install.packages("skimr") #to get the summary data
```

```
## Installing package into 'C:/Users/Dr. IJ/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository http://cran.rstudio.com/src/contrib:
##   cannot open URL 'http://cran.rstudio.com/src/contrib/PACKAGES'
```

```
## Warning: package 'skimr' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository http://cran.rstudio.com/bin/windows/contrib/4.1:
##   cannot open URL 'http://cran.rstudio.com/bin/windows/contrib/4.1/PACKAGES'
```

```
install.packages("janitor")
```

```
## Installing package into 'C:/Users/Dr. IJ/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository http://cran.rstudio.com/src/contrib:
##   cannot open URL 'http://cran.rstudio.com/src/contrib/PACKAGES'
```

```
## Warning: package 'janitor' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository http://cran.rstudio.com/bin/windows/contrib/4.1:
##   cannot open URL 'http://cran.rstudio.com/bin/windows/contrib/4.1/PACKAGES'
```

```
library(skimr) #to get the summary data
library(janitor)
```

```
##
## Attaching package: 'janitor'

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

```
library(dplyr)
```

Prepare:

Step 1: Download all the datasets needed for this project. I will be using 12 datasets from Jan_2021 to Dec_2021

I used the Cyclistic's historical trip data that is made available by Motivate International Inc. to analyze and identify trends. I downloaded trip data between Jan 2021 to Dec 2021. You can access the datasets here [link] (https://divvy-tripdata.s3.amazonaws.com/index.html). It is a public datasets provided by Motivate International Inc. I download the the files on the main directory.

I saved the 12 downloaded files and unzipped into a folder on my desktop and used appropriate file-naming conventions for easy referencing. Now i imported my 12 datasets into Rstudio by running the codes below to import the .csv datasets I imported each file to the R studio to check for data integrity, missing information and incorrect values. The only null values were in the starting and ending station id/name columns.

```
Trips_dec <- read_csv("202112-divvy-tripdata.csv")
```

```
## Rows: 247540 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_jan <- read_csv("202101-divvy-tripdata.csv")
```

```
## Rows: 96834 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_feb <- read_csv("202102-divvy-tripdata.csv")
```

```
## Rows: 49622 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_mar <- read_csv("202103-divvy-tripdata.csv")
```

```
## Rows: 228496 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_apr <- read_csv("202104-divvy-tripdata.csv")
```

```
## Rows: 337230 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_may <- read_csv("202105-divvy-tripdata.csv")
```

```
## Rows: 531633 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_jun <- read_csv("202106-divvy-tripdata.csv")
```

```
## Rows: 729595 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_jul <- read_csv("202107-divvy-tripdata.csv")
```

```
## Rows: 822410 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_aug <- read_csv("202108-divvy-tripdata.csv")
```

```
## Rows: 804352 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_sep <- read_csv("202109-divvy-tripdata.csv")
```

```
## Rows: 756147 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_oct <- read_csv("202110-divvy-tripdata.csv")
```

```
## Rows: 631226 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_nov <- read_csv("202111-divvy-tripdata.csv")
```

```
## Rows: 359978 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.
```

The R programming language was the tool of choice to process this data. Each file had a consistent format regarding column names, column order, etc. by using the codes below

```
spec(Trips_jan)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_feb)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
```

```
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_mar)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_apr)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_may)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
```

```
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_jun)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_jul)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_aug)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
```

```
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_sep)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_oct)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_nov)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
```

```
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
spec(Trips_dec)
```

```
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

I checked the data columns for name consistency this will aid the combining of the various data files into one to further my cleaning and analysis

```
colnames(Trips_jan)
```

```
##  [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"
```

```
colnames(Trips_feb)
```

```
##  [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"
```

```
colnames(Trips_mar)
```

```
##  [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"
```

```
colnames(Trips_apr)
```

```
##  [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"
```

```
colnames(Trips_may)
```

```
##  [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"
```

```
colnames(Trips_jul)
```

```
##  [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"
```

```
colnames(Trips_jun)
```

```
##  [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"
```

```
colnames(Trips_aug)
```

```
##  [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"
```

```
colnames(Trips_sep)
```

```
##  [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"
```

```
colnames(Trips_oct)
```

```
##  [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"
```

```
colnames(Trips_nov)
```

```
##  [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"
```

Compare the dataframes

```
compare_df_cols(Trips_jan, Trips_feb, Trips_mar, Trips_apr, Trips_may, Trips_jun, Trips_jul, Trips_aug,
```

```
##  [1] column_name Trips_jan   Trips_feb   Trips_mar   Trips_apr   Trips_may
##  [7] Trips_jun   Trips_jul   Trips_aug   Trips_sep   Trips_oct   Trips_nov
## [13] Trips_dec
## <0 rows> (or 0-length row.names)
```

Combine the dataframes together

Combines the dataframes together assigning it to a variable all_trips

```
all_trips <- bind_rows(Trips_jan, Trips_feb, Trips_mar, Trips_apr, Trips_may, Trips_jun, Trips_jul, Trip
```

Check the first few rows of all_trips

```
head(all_trips)
```

```
## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 E19E6F~ electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44 California Ave ~
## 2 DC88F2~ electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12 California Ave ~
## 3 EC45C9~ electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14 California Ave ~
## 4 4FA453~ electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55 California Ave ~
## 5 BE5E8E~ electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45 California Ave ~
## 6 5D8969~ electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54 California Ave ~
## # ... with 8 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>
```

Process

Exploring the data

Removing unused columns

I proceed to removing the columns i wont be needing for the analysis, first check the column names

```
colnames(all_trips)
```

```
##  [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"
```

Proceed to removing the unused columns

```
all_trips <- all_trips %>%
  select(-c(start_lat,start_lng, end_lng, end_lat))
```

inspect the colnames of the combined data all_trips

```
colnames(all_trips)
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "start_station_id"
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

I will use skim function to get summary of the data and check for missing data

```
skim(all_trips)
```

Table 1: Data summary

Name	all_trips
Number of rows	5595063
Number of columns	9
Column type frequency:	
character	7
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	16	16	0	5595063	0
rideable_type	0	1.00	11	13	0	3	0
start_station_name	690809	0.88	3	53	0	847	0
start_station_id	690806	0.88	3	36	0	834	0
end_station_name	739170	0.87	10	53	0	844	0
end_station_id	739170	0.87	3	36	0	832	0
member_casual	0	1.00	6	6	0	2	0

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2021-01-01 00:02:05	2021-12-31 23:59:48	2021-08-01 01:52:11	4677998
ended_at	0	1	2021-01-01 00:08:39	2022-01-03 17:32:18	2021-08-01 02:21:55	4671372

I will be doing a proper calculations, our interest lies on how much time each ride takes, and time analysis on month and days of the week. By adding columns for date list of month, day, and year of each ride The default format is yyyymm-dd

```
all_trips$date <- as.Date(all_trips$started_at)
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")
```

Add ride_length calculation

Add ride_length calculation to all_trips(in_seconds) using the difftime() to calculate the time difference between dates

```
all_trips$ride_length <- difftime(all_trips$ended_at, all_trips$started_at)
```

Convert "ride_length" from factor to numeric

I convert the "ride_length" from factor to numeric to run calculations on the column

```
is.factor(all_trips$ride_length)
```

```
## [1] FALSE
```

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)
```

```
## [1] TRUE
```

Remove bad data

The all_trips dataframe includes entries when bikes were taken out of stations and ride_length was negative, skim to check for missing data and then remove such data

```
skim(all_trips$ride_length)
all_trips_v2 <- all_trips[!(all_trips$ride_length<0),]
```

Analyze

Conduct discriptive analysis on the ride_length in seconds (the mean, median, max, and min)

```
mean(all_trips_v2$ride_length)
```

```
## [1] 1316.18
```

```
median(all_trips_v2$ride_length)
```

```
## [1] 720
```

```
max(all_trips_v2$ride_length)
```

```
## [1] 3356649
```

```
min(all_trips_v2$ride_length)
```

```
## [1] 0
```

compare the memberships of the riders i.e member and casual

```
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                1920.1327
## 2                     member                 818.0129
```

```
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                       958
## 2                     member                       576
```

```
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                   3356649
## 2                     member                     93596
```

```
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                       0
## 2                     member                       0
```

We can see the average ride by each day of week by member casual users

```
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                   Friday                1820.9160
## 2                      member                   Friday                 799.4950
## 3                      casual                   Monday                1912.5269
## 4                      member                   Monday                 794.8517
## 5                      casual                 Saturday                2082.3740
## 6                      member                 Saturday                 915.8742
## 7                      casual                   Sunday                2253.9949
## 8                      member                   Sunday                 939.4763
## 9                      casual                 Thursday                1662.1955
## 10                     member                 Thursday                 766.5710
## 11                     casual                  Tuesday                1678.3396
## 12                     member                  Tuesday                 767.2874
## 13                     casual                Wednesday                1659.4383
## 14                     member                Wednesday                 769.1496
```

Rearrange the average ride time by day of the week

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "
```

Check the first few rows of the all_trips_v2 data

```
head(all_trips_v2)
```

```
## # A tibble: 6 x 15
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 E19E6F~ electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44 California Ave ~
## 2 DC88F2~ electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12 California Ave ~
## 3 EC45C9~ electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14 California Ave ~
## 4 4FA453~ electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55 California Ave ~
## 5 BE5E8E~ electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45 California Ave ~
## 6 5D8969~ electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54 California Ave ~
## # ... with 10 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, member_casual <chr>, date <date>, month <chr>,
## #   day <chr>, year <chr>, day_of_week <ord>, ride_length <dbl>
```

Since i want to know how annual members and casual riders use use the cyclistic bike differently i will analyze the cyclistic data all_trips_v2 by type and weekday using the mutate() using wday() to create weekday field, group by user type and weekday and calculate the number of rides and average duration

```
all_trips_v2 %>%
  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)
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              481104            2254.
##  2 casual        Mon              286373            1913.
##  3 casual        Tue              274388            1678.
##  4 casual        Wed              278948            1659.
##  5 casual        Thu              286064            1662.
##  6 casual        Fri              364075            1821.
##  7 casual        Sat              557994            2082.
##  8 member        Sun              376117             939.
##  9 member        Mon              416204             795.
## 10 member        Tue              465509             767.
## 11 member        Wed              477156             769.
## 12 member        Thu              451520             767.
## 13 member        Fri              446423             799.
## 14 member        Sat              433041             916.
```

Share

I created some visualization in R using the ggplot2 ## Visualizing my findings with the ggplot2 package First load the gggplot2 package

```
library(ggplot2)
```

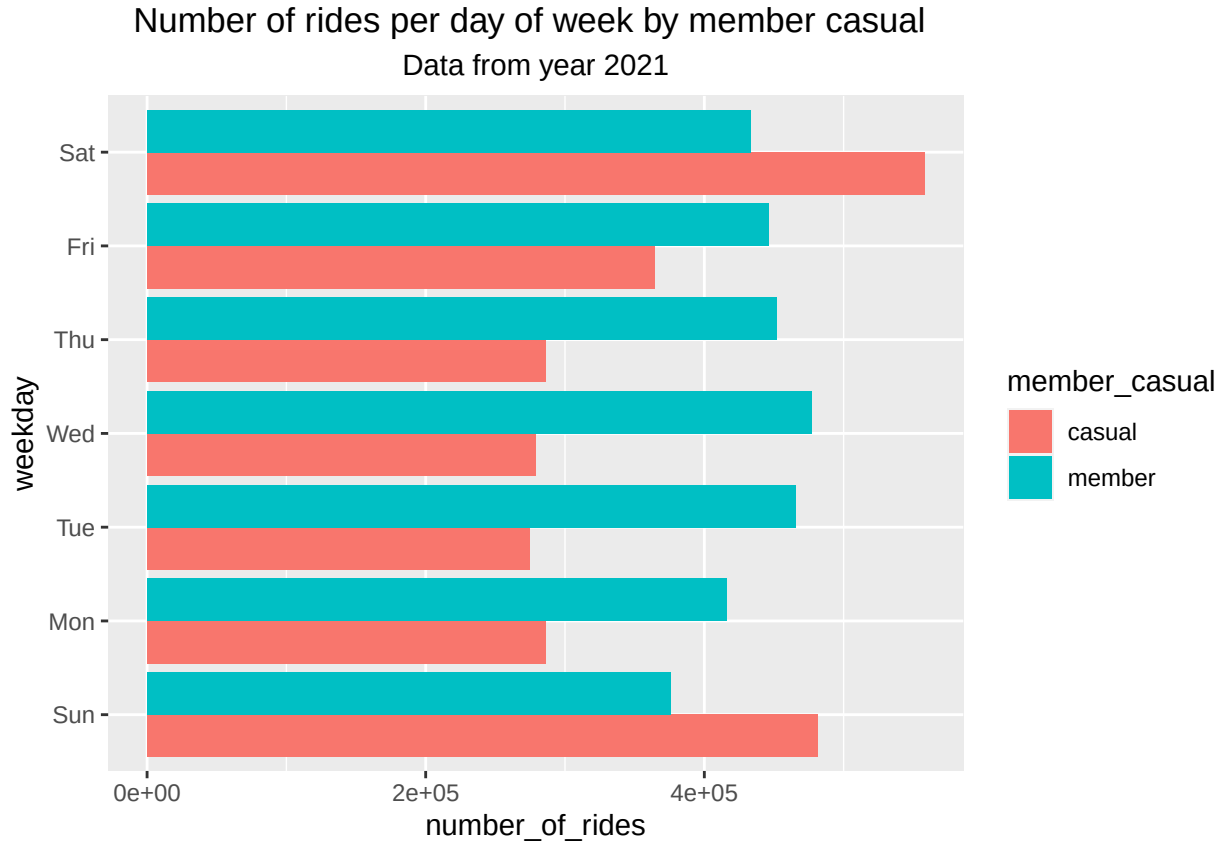
visualize the number of rides per day by member_casual

```
all_trips_v2 %>%
  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= number_of_rides, y = weekday, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of rides per day of week by member casual",
       subtitle = "Data from year 2021") +
```

```
  theme(plot.title = element_text(hjust = 0.4),
        plot.subtitle = element_text(hjust = 0.5))
```

fig 1

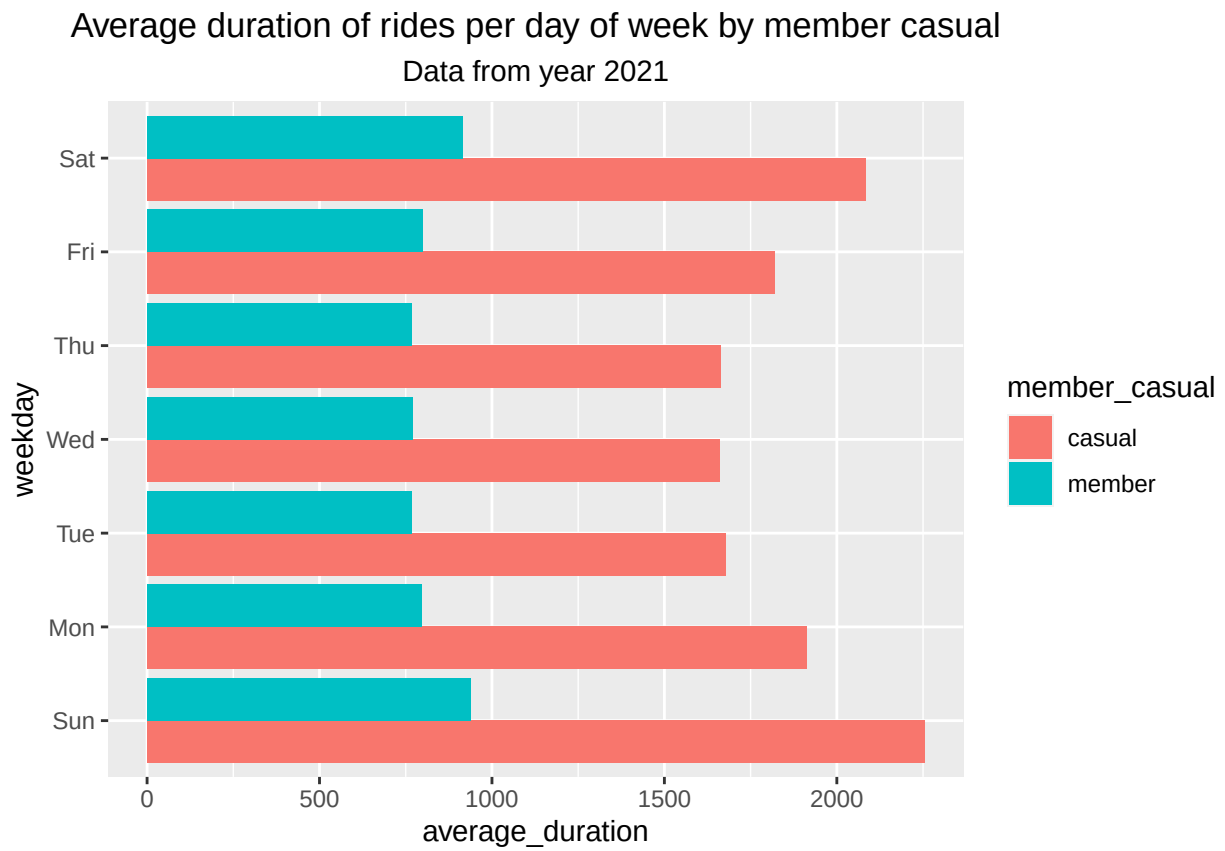
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



visualize the average ride duration per day of week by member casual #### Fig 2

```
all_trips_v2 %>%
  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= average_duration, y = weekday, fill = member_casual))+
  geom_col(position = "dodge") +
  labs(title = "Average duration of rides per day of week by member casual",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

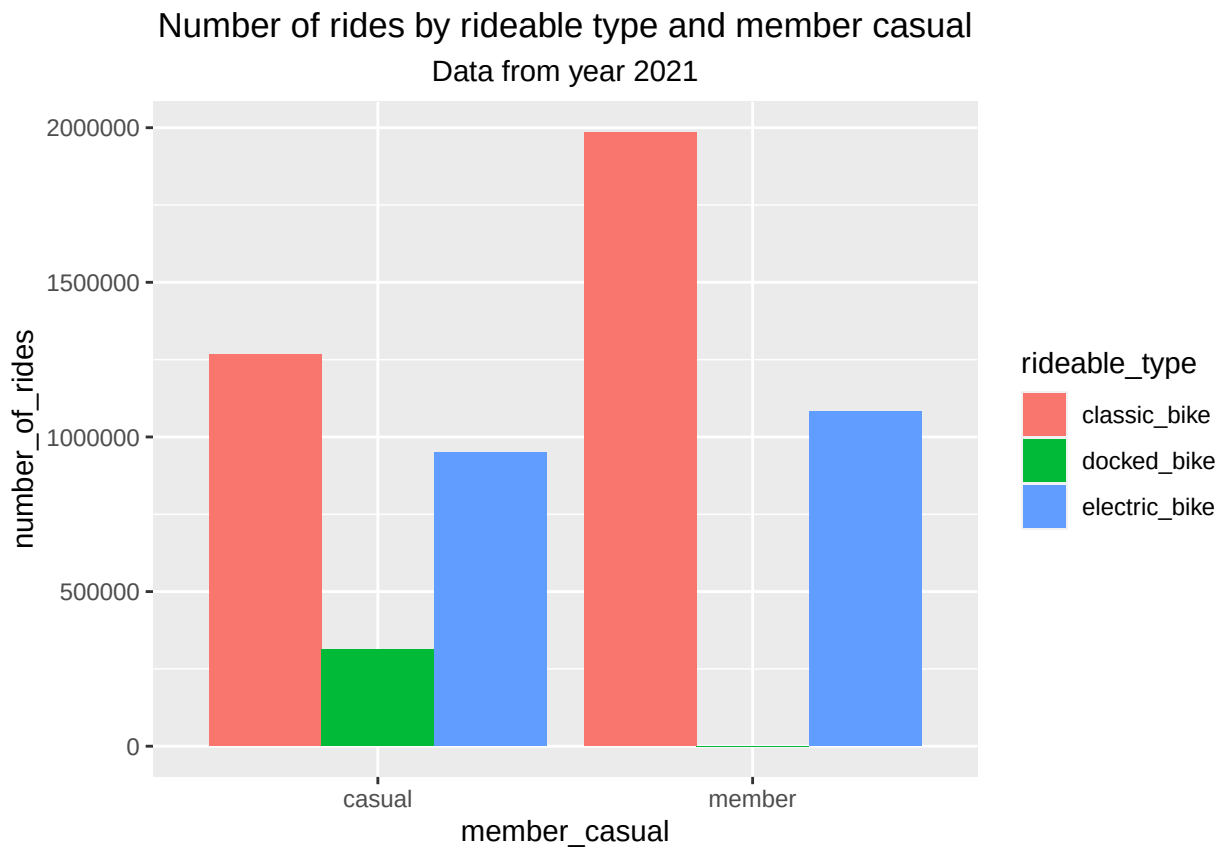
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



visualize number of rides by rider_type #### Fig 3

```
all_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual) %>%
  ggplot(aes(x= member_casual, y = number_of_rides, fill = rideable_type))+
  geom_col(position = "dodge") +
  labs(title = "Number of rides by rideable type and member casual",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

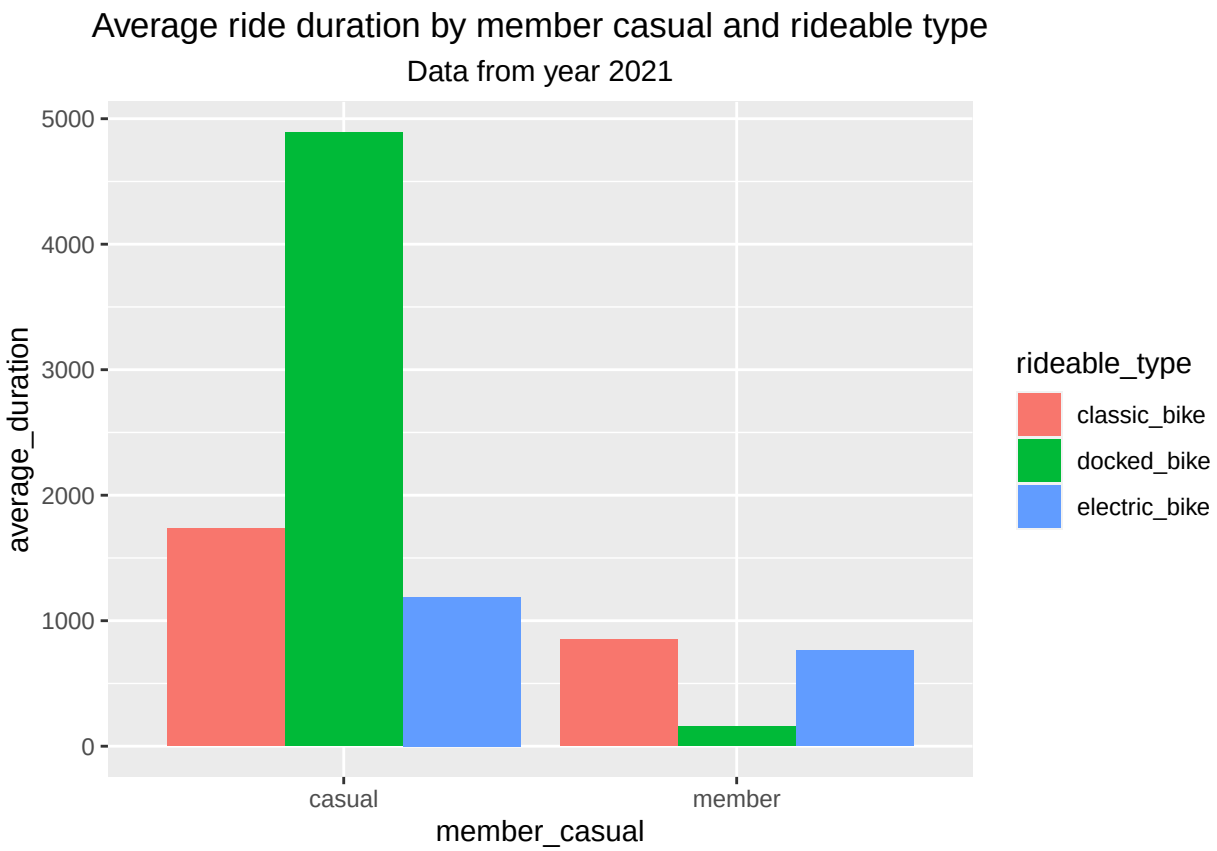
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



visualize the average ride duration by rider type on rideable type #### Fig 4

```
all_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual) %>%
  ggplot(aes(x= member_casual, y = average_duration, fill = rideable_type))+
  geom_col(position = "dodge") +
  labs(title = "Average ride duration by member casual and rideable type",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

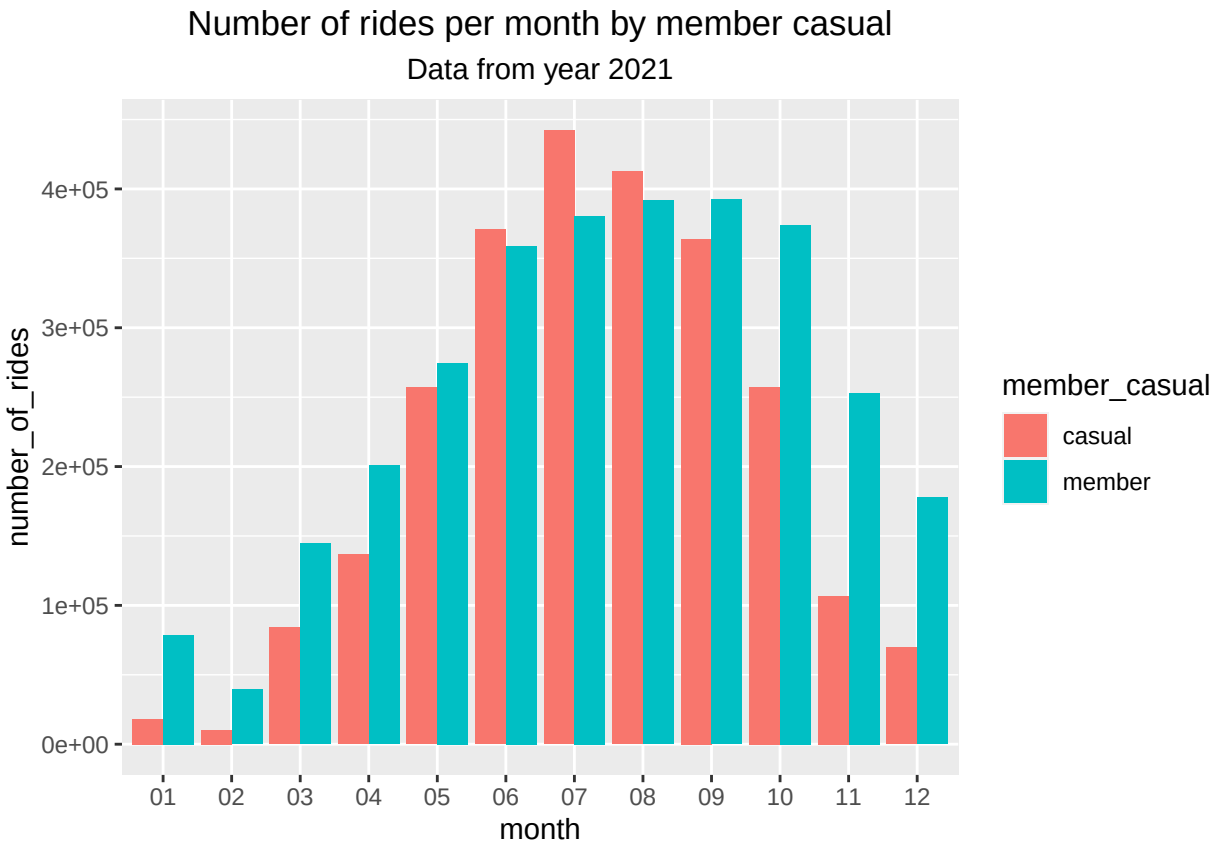
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



visualize the number of rides by each months #### Fig 5

```
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 per month by member casual",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

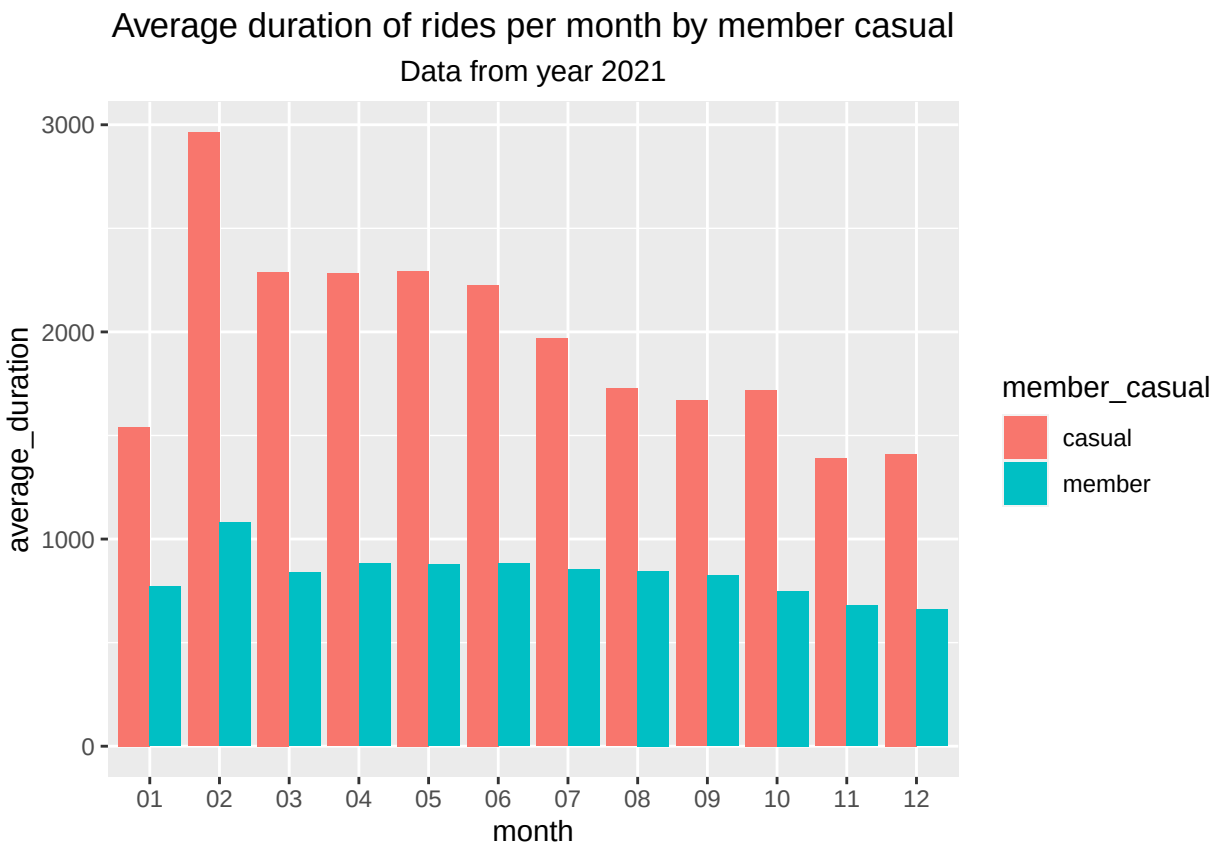
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



lets visualize the average duration for each month by rider_type #### Fig 6

```
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 per month by member casual",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

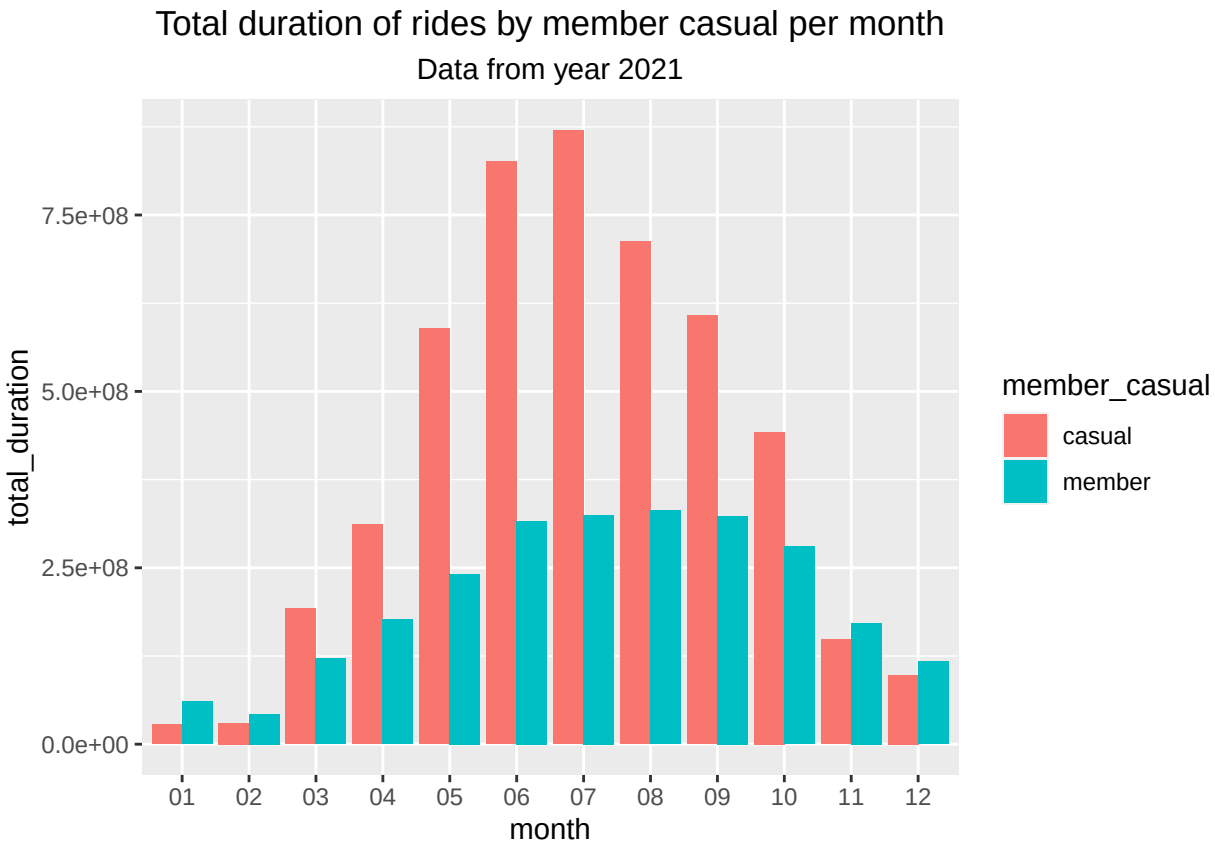
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



visualization total duration for each month #### Fig 7

```
all_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(),
            total_duration = sum(ride_length)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x= month, y= total_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Total duration of rides by member casual per month",
       subtitle = "Data from year 2021") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

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



Act My task here in this final stage is to state my findings on the analysis and give recommendation to the marketing director and stakeholder ## Key finding/ summary 1. From fig 1, the number of rides per day of the week by member casual there is more number of rides by the annual membership during the week than casual riders who have more rides during the weekends(Saturday & Sunday).

2. From the plot of the average duration of rides per day of the week by member casual, the average duration of rides by casual riders for each day is greater than the average duration of rides by an annual member.

3. From figure 3, number of rides by rideable type and member casual, it can be observed that the casual riders prefer the classic and electric bike to the docked bike while the annual members prefer the classic bike to the electric and docked bike.

4. From fig 4, average ride duration by member casual and rideable type, it can be seen that casual riders who use the docked bike had the highest average ride duration while classic bike used by annual members have the highest average ride duration.

5. Plot of the total duration of rides by member casual per month there was more total duration of rides by casual riders for most months except January, February, November, and December.

Recommendation

Based on my analysis here are my top 3 recommendation

1. The classic and electric bikes used by casual riders should be restricted to annual members-only, making the annual membership attractive to casual riders. From the number of rides by rideable type and member casual.

2. Reduce the ride duration by casual riders which in turn encourages annual membership sales since casual riders engage in longer ride duration(average duration of rides per month)

3. Launch a digital promotion and sales discount for an annual membership subscription to encourage more casual riders to purchase annual membership.