Cyclistic Bike Share

Haripriya Rajendran

2023-06-06

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

Cyclistic is a bike share company who plans to convert more customers being casual riders to membership holders. It is planning a marketing campaign for which a few of the questions are answered in this case study.

I will be using all 6 phases of Data Analytics as taught in Google Data Analytics certificate course.

Phase 1 - Ask:

Ask phase involves identifying the business task, asking SMART and effective questions, identifying the stakeholders and planning on problem solving.

Business task:

Convert more customers being casual riders to membership holders in order to increase the profit.

Key stakeholders:

The key stakeholders here are the Lily Moreno, the director of marketing and our manager and Cyclistic executive team,who will approve the marketing program.

Questions planned to be answered: (Business Task)

- How do the annual members and casual riders use Cyclistic bikes differently?
- If they are different, what strategy we can propose for the successful conversion?

Phase 2 - Prepare:

Prepare phase involves finding where the data is stored, how it is organized, how it is reliable and whether it ROCCCs. We also have to see from which party the data is from and how the privacy and security is taken care of.

- Where is the data located?

  Data is located under AWS S3 bucket

- Identify how it's organized.

  Data is organized as monthwise for the last three years and prior to them till 2013, they are all stored quarterwise. But we needed only previous 12 months of data.

- Are there issues with bias or credibility in this data? Does your data ROCCC?

  We are not sure on who collected the data. It may have been collected by Cyclistic and stored in AWS. In that case, the data is credible.

  - R - Reliable ? Yes, the data is reliable assuming the company itself uploaded them to AWS

  - O - Organized ? Yes, the data is organized yearly quarter or month wise
  - C - Comprehensive ? Yes, the data has the required details to answer the business question
  - C - Current ? Yes, the data is available till 2023
  - C - Cited ? There are no citations on the details on where the data is collected.

- How are you addressing licensing, privacy, security, and accessibility?

  Since it is an open source data, it's easily accessible. We can consider this as first party data since the company itself collected their own customer's data. The data has been made available by Motivate International Inc. under this license

Datasource Used:

Downloaded all the zipped files from this location dataset

Phase 3 - Process:

Process phase involves cleaning, organizing, making the data readily accessible for analysis.

Choosing the tools.

I have chosen RStudio for data cleaning and analysis.

Install the required Packages.

Install and load the required packages

```
knitr::opts_knit$set(root.dir = '/cloud/project/data')
```

```
# Install required packages
# tidyverse for data import and wrangling
# lubridate for date functions
# ggplot for visualization

install.packages('tidyverse')
install.packages('lubridate')
install.packages('ggplot2')
```

```
library(tidyverse)  #helps wrangle data
library(lubridate)  #helps wrangle date attributes
library(ggplot2)  #helps visualize data
#getwd() #displays your working directory
```

Transform the data and document the process.

In the below code chunks, we will be checking the data for errors, clean the data and document them. We will be checking the data column names and whether they need any transformation before merging. We will also be converting all csv to a single data frame.

Since importing all 12 month files leads to crash, we are importing only last 6 months data.

```
#Generating all filenames automatically
all_months <- seq(as.Date("2022-11-01"), as.Date("2023-04-01"), by="month")
all_months <- format(all_months, "%Y%m")

for (each_month in all_months){
  assign(sprintf('data_%s', each_month), read_csv(sprintf('%s-divvy-tripdata.csv',each_month)))
}
```

```
head(data_202211)
```

Check if data fetched properly

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 BCC66FC6FAB27CC7 electric_bike 2022-11-10 06:21:55 2022-11-10 06:31:27
## 2 772AB67E902C180F classic_bike  2022-11-04 07:31:55 2022-11-04 07:46:25
## 3 585EAD07FDEC0152 classic_bike  2022-11-21 17:20:29 2022-11-21 17:34:36
## 4 91C4E7ED3C262FF9 classic_bike  2022-11-25 17:29:34 2022-11-25 17:45:15
## 5 709206A3104CABC8 classic_bike  2022-11-29 17:24:25 2022-11-29 17:42:51
## 6 11DE62E16D1A6BD1 classic_bike  2022-11-04 14:40:47 2022-11-04 14:52:35
## # i 9 more variables: start_station_name <chr>, 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>
```

```
#checking the earliest set and the latest set has same set of columns
all(colnames(data_202211) == colnames(data_202304)) #should return TRUE
```

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

```
all_df <- bind_rows( data_202211, data_202212, data_202301, data_202302, data_202303, data_202304)
head(all_df)
```

Combine all data together

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 BCC66FC6FAB27CC7 electric_bike 2022-11-10 06:21:55 2022-11-10 06:31:27
## 2 772AB67E902C180F classic_bike  2022-11-04 07:31:55 2022-11-04 07:46:25
## 3 585EAD07FDEC0152 classic_bike  2022-11-21 17:20:29 2022-11-21 17:34:36
## 4 91C4E7ED3C262FF9 classic_bike  2022-11-25 17:29:34 2022-11-25 17:45:15
## 5 709206A3104CABC8 classic_bike  2022-11-29 17:24:25 2022-11-29 17:42:51
## 6 11DE62E16D1A6BD1 classic_bike  2022-11-04 14:40:47 2022-11-04 14:52:35
## # i 9 more variables: start_station_name <chr>, 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>
```

```
rm("data_202211","data_202212", "data_202301", "data_202302", "data_202303", "data_202304")
```

Removing the other data frames from memory, since we have less RAM

```
str(all_df)
```

Checking all the columns and their values

```
## spc_tbl_ [1,585,555 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1585555] "BCC66FC6FAB27CC7" "772AB67E902C180F" "585EAD07FDEC0152" "91C4
##  $ rideable_type     : chr [1:1585555] "electric_bike" "classic_bike" "classic_bike" "classic_bike"
##  $ started_at        : POSIXct[1:1585555], format: "2022-11-10 06:21:55" "2022-11-04 07:31:55" ...
##  $ ended_at          : POSIXct[1:1585555], format: "2022-11-10 06:31:27" "2022-11-04 07:46:25" ...
```

```
##  $ start_station_name: chr [1:1585555] "Canal St & Adams St" "Canal St & Adams St" "Indiana Ave & Roo
##  $ start_station_id  : chr [1:1585555] "13011" "13011" "SL-005" "SL-005" ...
##  $ end_station_name  : chr [1:1585555] "St. Clair St & Erie St" "St. Clair St & Erie St" "St. Clair S
##  $ end_station_id    : chr [1:1585555] "13016" "13016" "13016" "13016" ...
##  $ start_lat         : num [1:1585555] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1585555] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:1585555] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1585555] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:1585555] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. 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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
all_df$ride_length <- difftime(all_df$ended_at, all_df$started_at)
#The above line added "secs" to the time, e.g 572 secs, so to remove that properly we did the below

is.factor(all_df$ride_length)
```

Creating additional columns

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

```
is.numeric(all_df$ride_length)
```

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

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

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

```
#create columns for year, month, date
all_df$year  <- format(as.Date(all_df$started_at), "%Y")
all_df$month  <- format(as.Date(all_df$started_at), "%m")
all_df$day  <- format(as.Date(all_df$started_at), "%d")
all_df$day_of_week  <- format(as.Date(all_df$started_at), "%A")

#all_df %>% select(year, month, day, day_of_week)
```

```
unique(all_df$member_casual)
```

Create columns for year, month, date

```
## [1] "member" "casual"
```

```
unique(all_df$rideable_type)
```

```
## [1] "electric_bike" "classic_bike"  "docked_bike"
```

Some ride lengths are zero which are irrelevant

```
#dim(all_df[(all_df$ride_length < 0),])

all_df <- all_df[(all_df$ride_length >= 0),]
```

Phase 4 : Analyze

We have cleaned the data and created required columns. Now we will group by and analyze.

Let's have day_of_week to be in order

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

Check ride length by day of the week

```
all_df %>%
  group_by(day_of_week) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
```

```
## # A tibble: 7 x 5
##   day_of_week number_of_rides mean_ride_length max_ride_length min_ride_length
##   <ord>                 <int>            <dbl>           <dbl>           <dbl>
## 1 Sunday               176672            1081.         2016224               0
## 2 Monday               204206             784.         1176756               0
## 3 Tuesday              259376             767.         1103729               0
## 4 Wednesday            260081             787.         1131946               0
## 5 Thursday             259747             822.         1002844               0
## 6 Friday               226382             857.         1149426               0
## 7 Saturday             199045            1099.          925311               0
```

Analyze weekly data Check ride length by day of the week and by member_casual

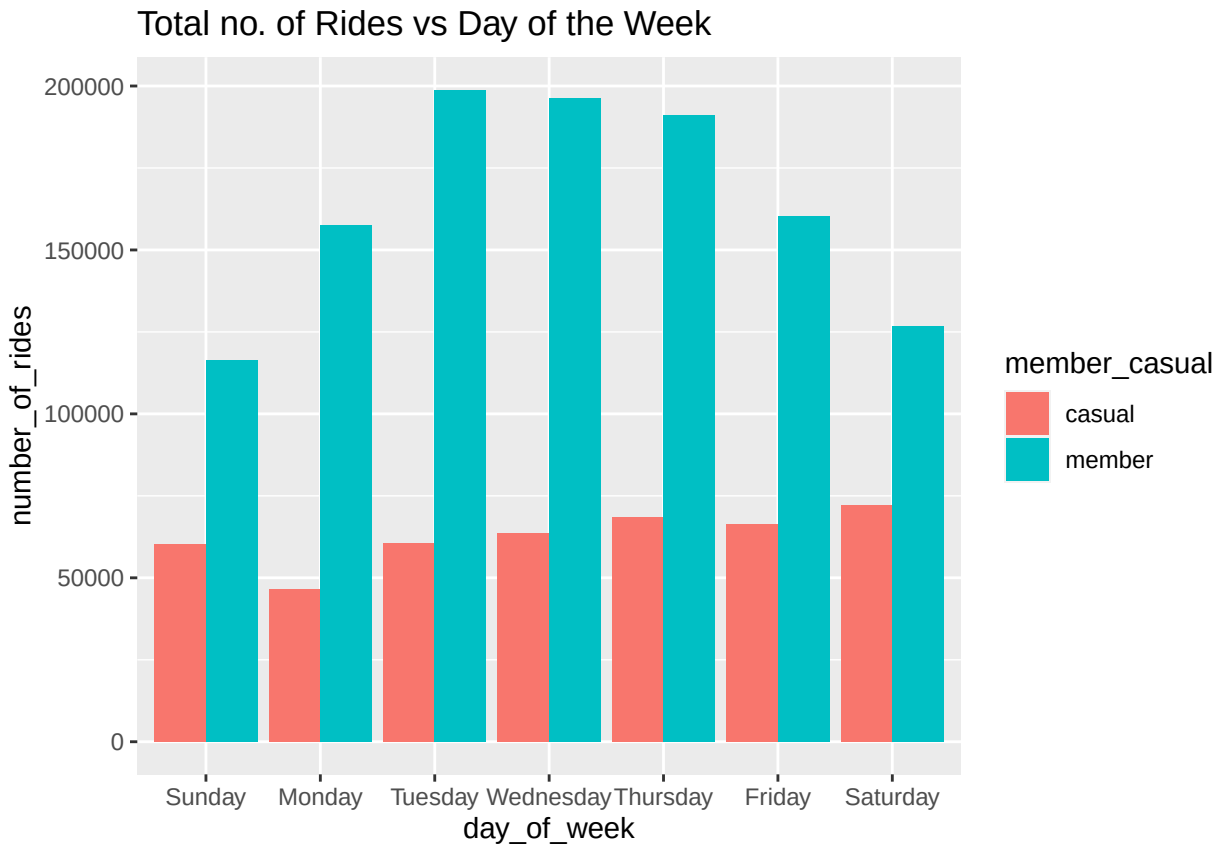
```
all_df %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
```

```
## # A tibble: 14 x 6
## # Groups:   day_of_week [7]
##    day_of_week member_casual number_of_rides mean_ride_length max_ride_length
##    <ord>       <chr>                  <int>            <dbl>           <dbl>
##  1 Sunday      casual                 60299            1790.         2016224
##  2 Sunday      member                116373             713.           89996
##  3 Monday      casual                 46659            1327.         1176756
##  4 Monday      member                157547             623.           89996
##  5 Tuesday     casual                 60527            1200.         1103729
##  6 Tuesday     member                198849             636.           89996
```

```
##  7 Wednesday   casual                63687           1249.         1131946
##  8 Wednesday   member               196394            638.           89996
##  9 Thursday    casual                68523           1308.         1002844
## 10 Thursday    member               191224            647.           89996
## 11 Friday      casual                66240           1340.         1149426
## 12 Friday      member               160142            657.           89996
## 13 Saturday    casual                72215           1768.          925311
## 14 Saturday    member               126830            717.           93580
## # i 1 more variable: min_ride_length <dbl>
```

Let's plot the number of rides by rider type and the day of the week and see if we get any insights.

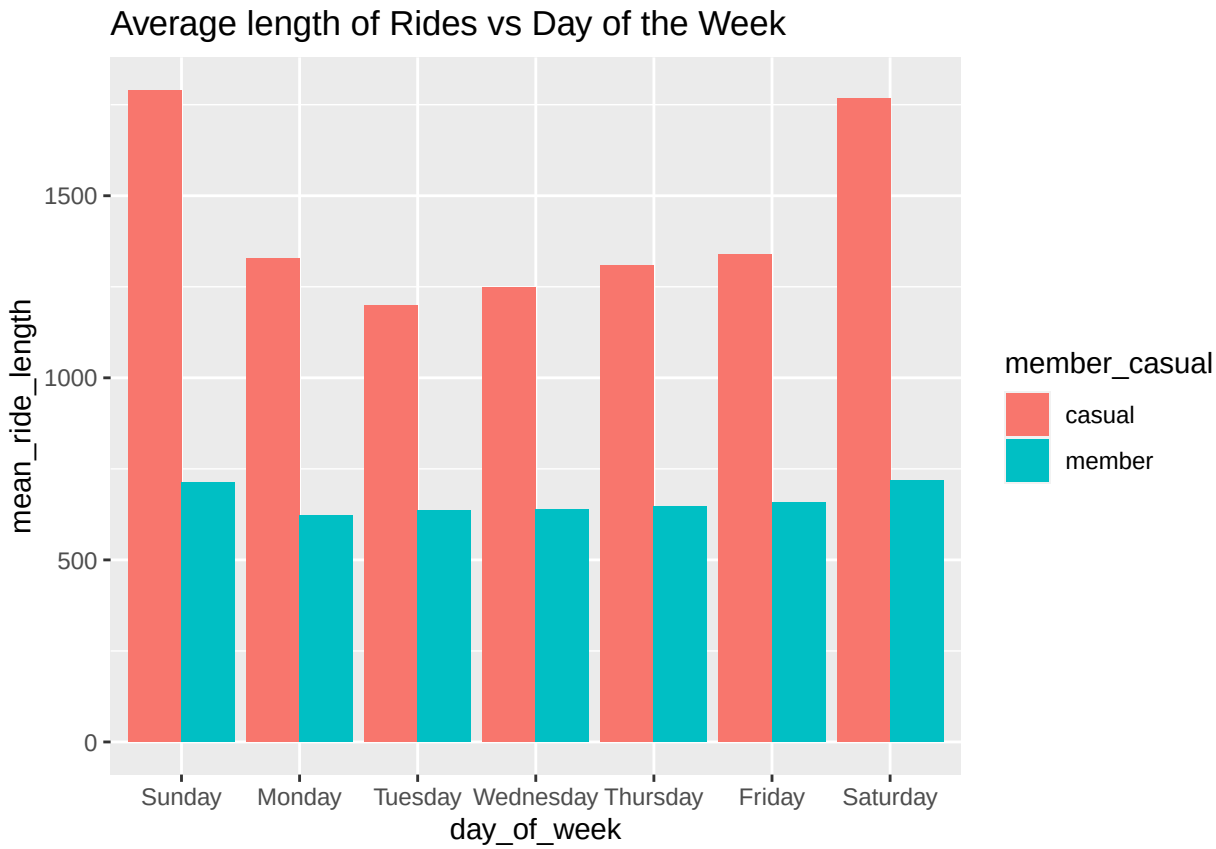
```
all_df %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = day_of_week, y= number_of_rides, fill = member_casual)) + geom_col(position =
 labs(title = "Total no. of Rides vs Day of the Week")
```



We can see clearly that whatever day of the week is, casual riders are booking less number of rides than the member riders

Let's plot the average ride length by rider type and the day of the week

```
all_df %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = day_of_week, y= mean_ride_length, fill = member_casual)) + geom_col(position
   labs(title = "Average length of Rides vs Day of the Week")
```

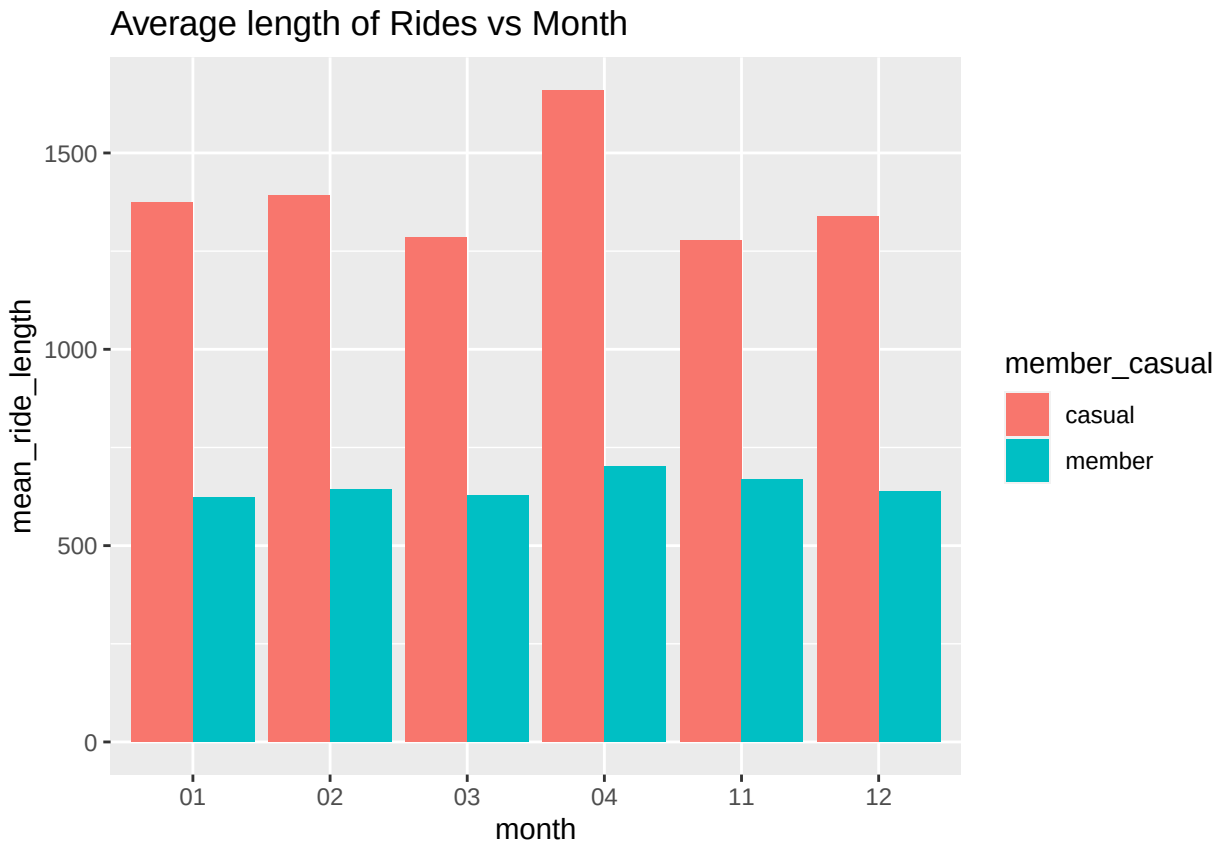


But when we check the average ride length, casual riders are riding more time than members.

It seems even though members are taking many number of rides, they are driving for a consistent amount of time and the ride length is significantly less than the casual riders. We can see clearly that during weekdays, the member riders have almost the same ride length indicating that they are using it for daily routine like riding to work or school.

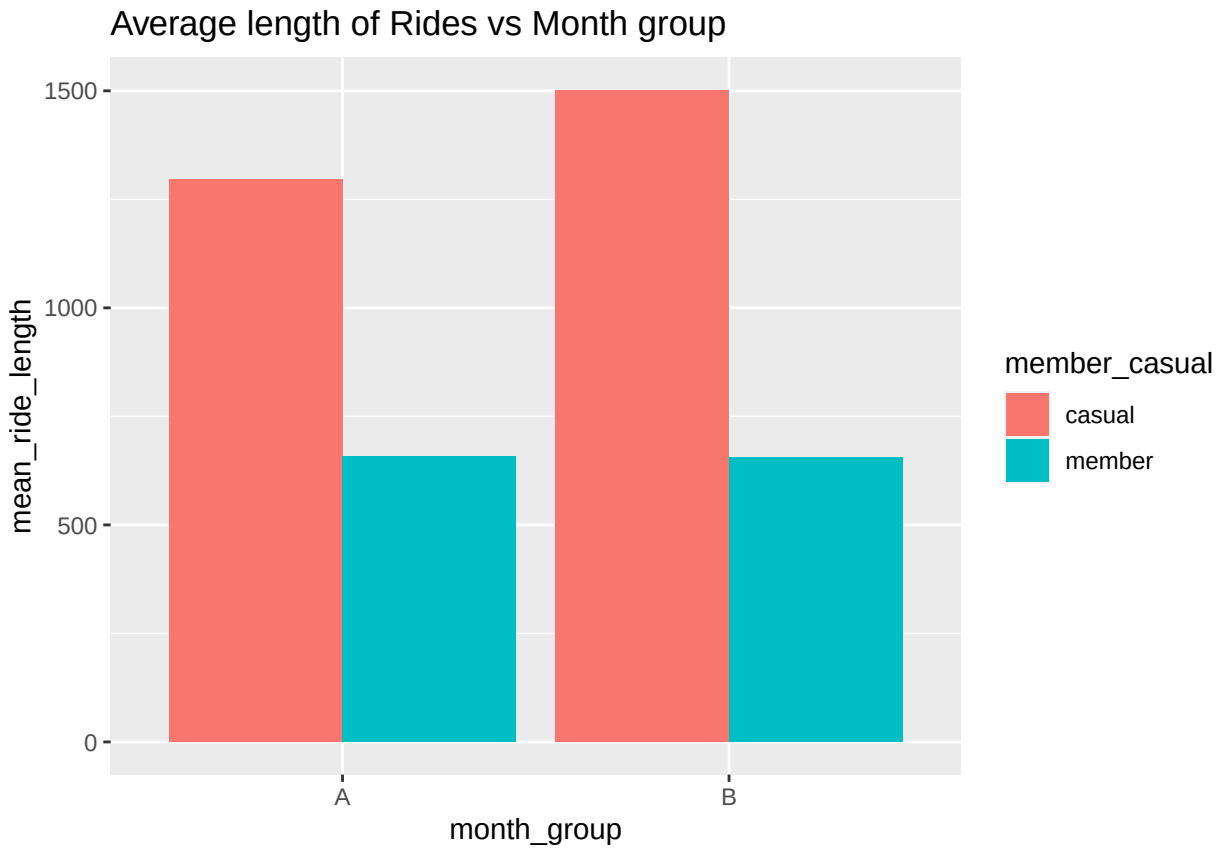
Analyze seasonal data Ideally, if one year data is taken, we could analyse them seasonally. But we have only 6 months data, so we will see by months

```
all_df %>%
  group_by(month, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = month, y= mean_ride_length, fill = member_casual)) + geom_col(position = "do
   labs(title = "Average length of Rides vs Month")
```



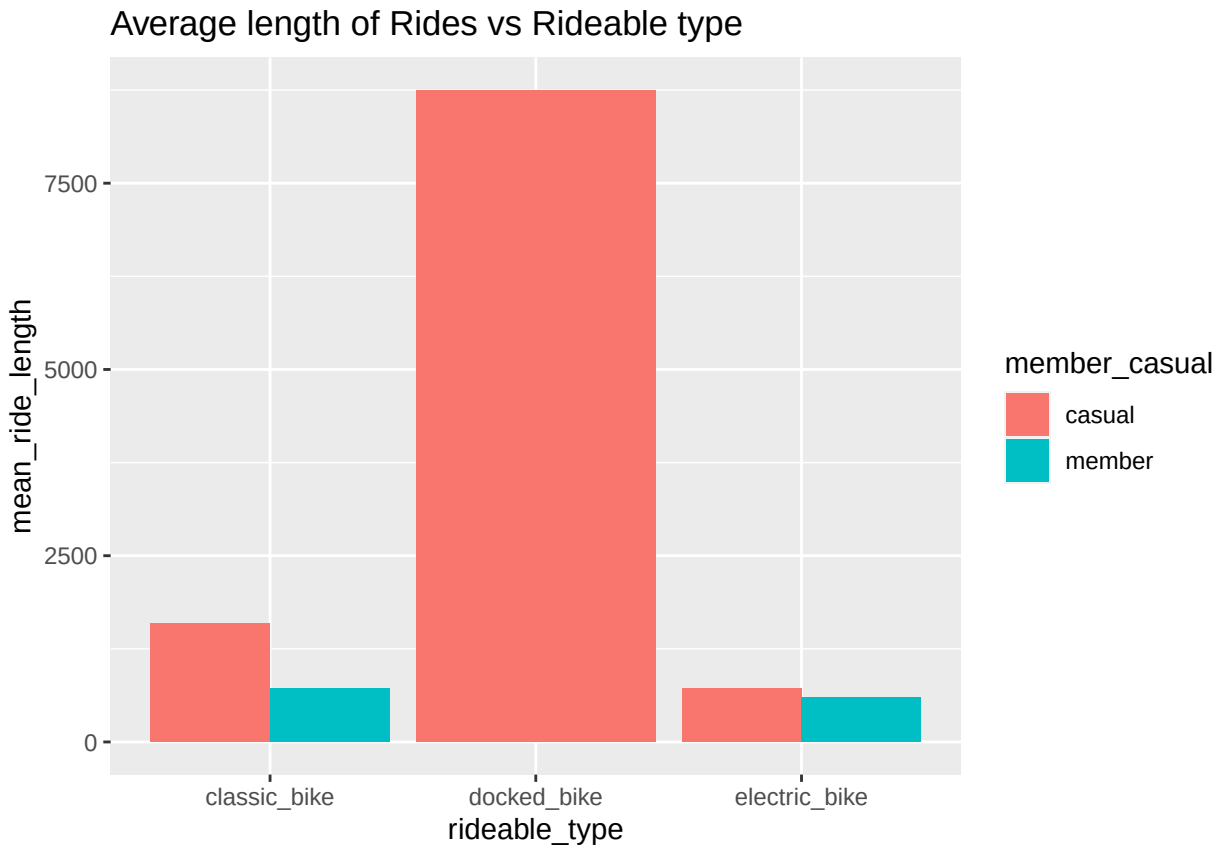
Grouping 3 months each to check if there are any changes in pattern.

```
all_df %>%
  mutate(month_group = ifelse(month %in% c(11,12,1), 'A', 'B')) %>%
  group_by(month_group, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = month_group, y= mean_ride_length, fill = member_casual)) + geom_col(position
   labs(title = "Average length of Rides vs Month group")
```

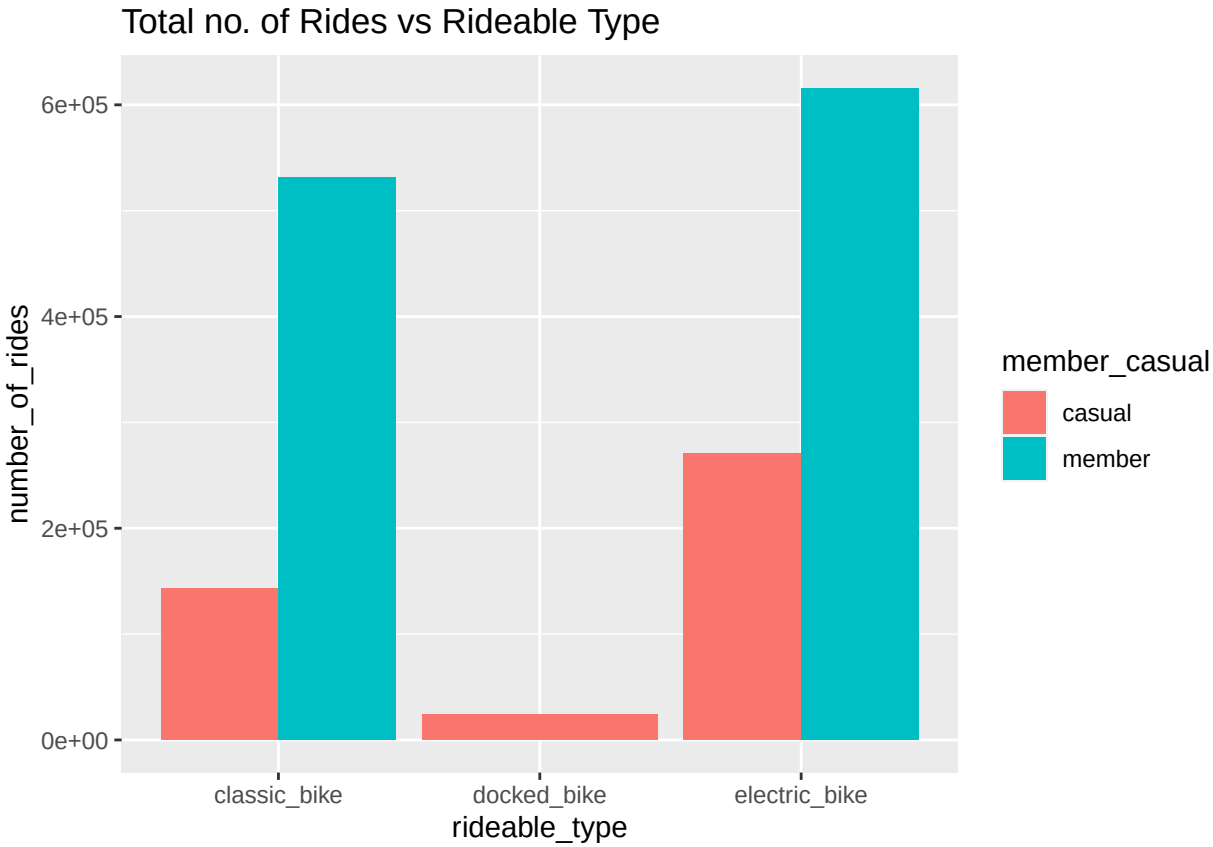


Let's see if some type of bike is preferred by casual or member riders

```
all_df %>%
  group_by(rideable_type, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = rideable_type, y= mean_ride_length, fill = member_casual)) + geom_col(positi
   labs(title = "Average length of Rides vs Rideable type")
```




```
all_df %>%
  group_by(rideable_type, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
  ggplot(mapping = aes(x = rideable_type, y= number_of_rides, fill = member_casual)) + geom_col(positio
   labs(title = "Total no. of Rides vs Rideable Type")
```



We can see docked_bike is mostly used by casual riders and they are taking it for long rides. We can use this information and plan on giving more offers in docked_bike on successful conversion of casual riders to member riders.

```
write.csv(all_df %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_rides = n(),mean_ride_length = mean(ride_length), max_ride_length = max(ride_leng
```

Export Summary data frame to a csv

Phase 5: Share

Determine the best way to share the findings. We have created few graphs during analysis for finding insights, for showing them a Powerpoint presentation would be a proper approach.

Create effective data visualizations. The earlier graphs we created are properly labelled and coloured so that it's easier to understand

Ensuring the work is accessible Since this is an R markdown, the html or pdf is shareable and accessible by everyone.

Phase 6 : Act

Below are the conclusions from our Analysis.

- **How are the riders different ?** It is clear that casual riders ride more than the member riders at any time of the year.

- **Weekend Pattern:** On Weekends, casual riders ride longer. We can advertise to offer them weekend discounts if they convert to a member.
- **Docked Bike usage discount:** Docked bike is being preferred the most by casual riders for long ride even if their number of rides is less. So, we can advertise to those casual riders who convert to memberships to give a discount when docked bike is used for long.
- **Survey:** Conducting a survey with the above insights to the casual riders might give some more additional data which we can use to expand our findings.