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

Wael Chmaisani

2023-06-17

Contents

1. Introduction

The following study case is one of the Capstone Project by Google Data Analytics Professional Certificate.

Cyclistic is a bike-share program that features more than 5,800 bicycles and 600 docking stations.

The director of marketing of Cyclistic,Lily Moreno, believes the company future success depends on maximizing the number of 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. Cyclistic finance analysts have concluded that annual members are much more profitable than casual riders. From these insights, our team will design a new marketing strategy to convert casual riders into annual members.

In order to answer the key business questions, you will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

2. Stakeholders

Cyclistic: A bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.

Lily Moreno: The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.

Cyclistic marketing analytics team: A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals — as well as how you, as a junior data analyst, can help Cyclistic achieve them.

Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

3. Ask Questions To Answer Business Problem

Three questions will guide the future marketing program:

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would casual riders buy Cyclistic annual memberships?
- How can Cyclistic use digital media to influence casual riders to become members?

4. Prepare Data

Data location:

We will use Cyclistic historical trip data to analyze and identify trends. Download the previous 12 months of Cyclistic trip data here.

Data Licence:

The data has been made available by Motivate International Inc. under this licence where we can addressing licensing, privacy, security, and accessibility of data.

Data Organization:

The data is organized in chronological order and it obeys naming conventions. The data is unbiased since there is no overrepresenting or underrepresentation of certain members over the years. The study is limited for the last 12 months of Cyclistic trip data from Juin 2022 to April 2023, which is stored in separate files for each month.

Data decription:

The attributes of data are summarized in the following table:

Column Name	Desciption
ride_id	ID of ride
rideable_type	Type of bike like docked, electric or classic
started_at	Stating datetime of ride
ended_at	Ending datetime of ride
start_station_name	Station name of ride starting
start_station_id	Station ID of ride starting
end_station_name	Station name of ride ending
end_station_id	Station ID of ride ending
start_lat	Latitude of ride starting
start_lng	Longitude of ride starting
end_lat	Latitude of ride ending
end_lng	Longitude of ride ending
member_casual	Type of member as casual or membership in Cyclistic

Data ROCCC validation:

In order to inspect the credibility and the bias of data, we will check the ROCCC of data:

- Reliable: the data is not complete since some recorded months do not have the name and id of the starting and ending station. The data is not reliable.
- Original: The data was generated by Lyft Bikes and Scooters, LLC (“Bikeshare”) which is operates the City of Chicago’s (“City”) Divvy bicycle sharing service. The City permits Bikeshare to make certain Divvy system data owned by the City (“Data”) available to the public. We can consider the data is

original since the Bike-share firm itself collect the data and it will do the study.
- Comprehensive: it has enough information that can answer our asked questions.
- Current: the data was recorded between years 2022 and 2023, we can consider it is a current data.
- Cited: the data is available on “https://divvy-tripdata.s3.amazonaws.com/index.html” site but this is not a reliable reference. The data is not cited.

overall, the criteria of integrity do not fully applied on our data.

3. Process Data

Since we will work with a large data, R programming language is preferred.

Install Required Library

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("data.table")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("gglot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
## Warning: package 'gglot2' 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
```

```
install.packages("scales")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

- installing *binary* package ‘data.table’ …
- DONE (data.table)

The downloaded source packages are in ‘/tmp/RtmpyDRjZy/downloaded_packages’

Load packages

```
library(data.table)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(lubridate)
library(dplyr)
library(ggplot2)
library(scales)
```

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

Import data

```
trip_202206 <- read_csv("202206-divvy-tripdata.csv")
trip_202207 <- read_csv("202207-divvy-tripdata.csv")
trip_202208 <- read_csv("202208-divvy-tripdata.csv")
trip_202209 <- read_csv("202209-divvy-publictripdata.csv")
trip_202210 <- read_csv("202210-divvy-tripdata.csv")
trip_202211 <- read_csv("202211-divvy-tripdata.csv")
trip_202212 <- read_csv("202212-divvy-tripdata.csv")
trip_202301 <- read_csv("202301-divvy-tripdata.csv")
trip_202302 <- read_csv("202302-divvy-tripdata.csv")
trip_202303 <- read_csv("202303-divvy-tripdata.csv")
trip_202304 <- read_csv("202304-divvy-tripdata.csv")
trip_202305 <- read_csv("202305-divvy-tripdata.csv")
```

Check the column names of each data frame before merging them

```
colnames(trip_202206)
colnames(trip_202207)
colnames(trip_202208)
colnames(trip_202209)
colnames(trip_202210)
colnames(trip_202211)
colnames(trip_202212)
colnames(trip_202301)
colnames(trip_202302)
colnames(trip_202303)
colnames(trip_202304)
colnames(trip_202305)
```

Merge the 12 data frame into one data frame naming cyclitic_df

```
cyclitic_df = bind_rows(trip_202206,trip_202207,trip_202208,trip_202209,trip_202210,trip_202211,trip_202212,trip_20230
```

Display the columns and the first several rows of data of cyclitic_df

```
head(cyclitic_df)
```

Return summaries of each column in cyclitic_df data arranged horizontally

```
str(cyclitic_df)
```

```
spc_tbl_ [5,829,030 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
$ ride_id : chr [1:5829030] "600CFD130D0FD2A4" "F5E6B5C1682C6464" "B6EB6D27BAD771D2"
"C9C320375DE1D5C6" ...
$ rideable_type : chr [1:5829030] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
$ started_at : POSIXct[1:5829030], format: "2022-06-30 17:27:53" "2022-06-30 18:39:52" ...
$ ended_at : POSIXct[1:5829030], format: "2022-06-30 17:35:15" "2022-06-30 18:47:28" ...
$ start_station_name: chr [1:5829030] NA NA NA NA ...
$ start_station_id : chr [1:5829030] NA NA NA NA ...
$ end_station_name : chr [1:5829030] NA NA NA NA ...
$ end_station_id : chr [1:5829030] NA NA NA NA ...
$ start_lat : num [1:5829030] 41.9 41.9 41.9 41.8 41.9 ...
$ start_lng : num [1:5829030] -87.6 -87.6 -87.7 -87.7 -87.6 ...
$ end_lat : num [1:5829030] 41.9 41.9 41.9 41.8 41.9 ...
$ end_lng : num [1:5829030] -87.6 -87.6 -87.6 -87.7 -87.6 ...
$ member_casual : chr [1:5829030] "casual" "casual" "casual" "casual" ...
- 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")=
```

Obtain the number of rows of cyclitic_df

```
nrow(cyclitic_df)
```

```
[1] 5829030
```

Remove the four columns concerning the latitude and longitude

```
cyclitic_df <- cyclitic_df %>%
select(ride_id, rideable_type, started_at, ended_at, start_station_name,start_station_id, end_station_name,
end_station_id, member_casual)
```

Remove the missing value

```
cyclitic_df <- cyclitic_df %>% drop_na()
```

In this study, the data is very large so we decide remove the trip entries with missing values. But, in real scenario we will discuss the outcomes of this action. We check the number of rows after remove N/A values and we obtain 4494681 rows. It is means that the data has 1334349 records of missing starting and ending stations.

Add new column naming "ride_length" for the length of each ride by subtracting the "started_at" column from the "ended_at"" columns

```
cyclitic <- mutate(cyclitic, ride_length = started_at - ended_at)
```

Add new column naming "day_of_week" for the day of ride

```
cyclitic_df$day_of_week <- weekdays(as.Date(started_at))
```

Remove duplicate data

```
cyclitic_df <- distinct(cyclitic_df)
```

We note that there is no duplicate rows.

Save the clean cyclitic_df data frame as a csv file on desktop

```
write.csv(cyclitic_df,"C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_df.csv", row.names=FALSE)
```

4. Analyze Data

Now that our data is stored appropriately and has been prepared for analysis.

Descriptive Analysis:

Summary statistics for ride length in seconds:

```
summary(cyclitic_df$ride_length)
```

```
  Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
  0.0     347.0     609.0     977.2    1090.0 1922127.0
```

Ride length mean for each member type:

```
cyclitic_df %>% group_by(member_casual) %>% summarize (ride_length_mean_for_member = mean(ride_length))
```

member_casual ride_length_mean_for_member

1 casual	1363.
2 member	732.

from here we can notice that the mean of ride length is greater for casual riders than annual members.

Frequency for each membership type of our data:

```
cyclitic_df %>% group_by(member_casual) %>% summarize(Freq=n())
```

member_casual	Freq
1 casual	1747907
2 member	2746774

the number of member riders is greater than of casual riders by 998867.

Frequency of rides for each membership type grouped by day of rider:

```
cyclitic_df %>% group_by(member_casual,day_of_week) %>% summarize(Freq=n())
```

member_casual	day_of_week	Freq
1 casual	Friday	258175
2 casual	Monday	194974
3 casual	Saturday	350970
4 casual	Sunday	289572
5 casual	Thursday	233766
6 casual	Tuesday	202337
7 casual	Wednesday	218113
8 member	Friday	387064
9 member	Monday	374374
10 member	Saturday	343725
11 member	Sunday	303839
12 member	Thursday	442117
13 member	Tuesday	438809
14 member	Wednesday	456846

The most number of bicycle rents was in weekend for the casual riders whereas it was in weekdays for member riders.

Frequency of rides for each membership type grouped by the type of bike:

```
cyclitic_df %>% group_by(member_casual,rideable_type) %>% summarize(Freq=n())
```

member_casual	rideable_type	Freq
1 casual	classic_bike	855180
2 casual	docked_bike	154494
3 casual	electric_bike	738233
4 member	classic_bike	1729767
5 member	electric_bike	1017007

We can notice that the docked bike is exclusive for the casual riders only.

5. Share Data

Now that we have performed our analysis and gained some insights into our data, we will create visualizations to share our findings.

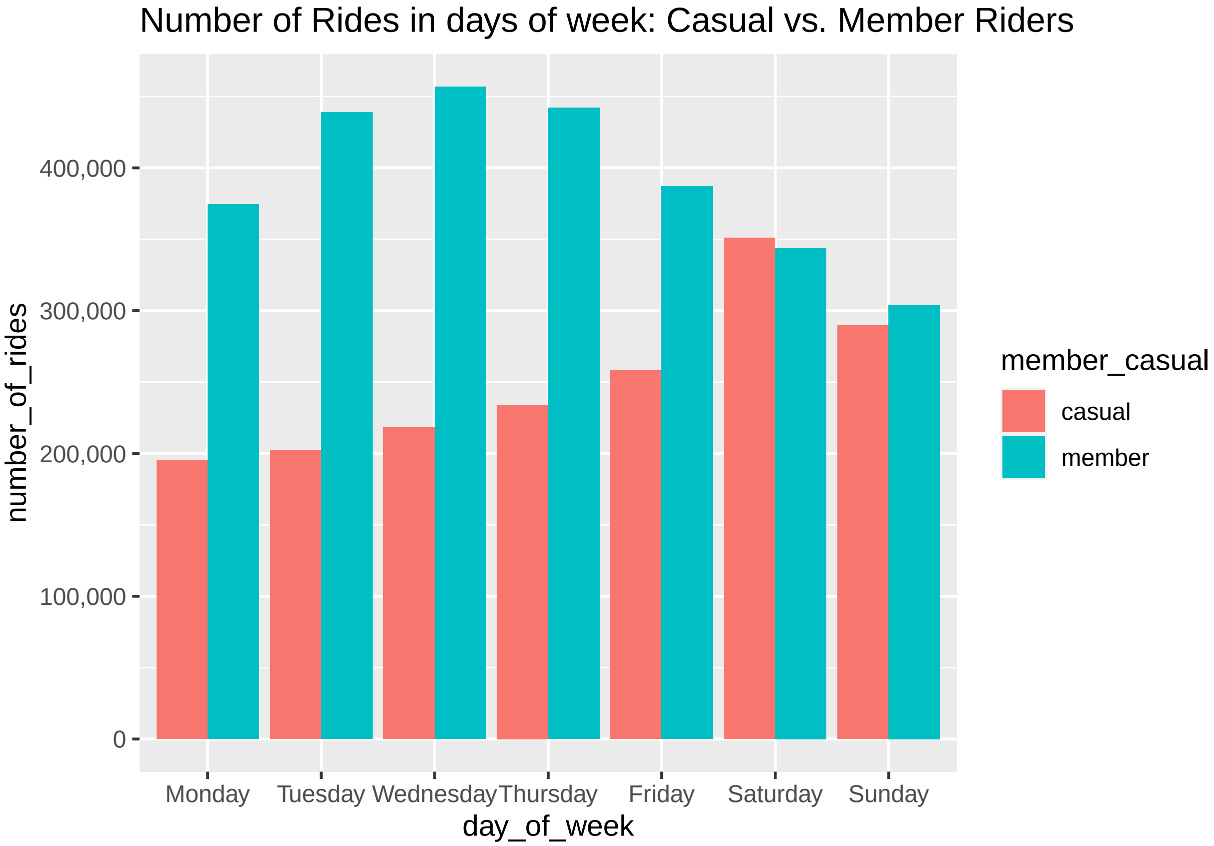
bar chart plot of number of rides for casual and annual member in different days of week:

```
cyclitic_plot1 <- cyclitic %>%
group_by(member_casual,day_of_week) %>%
summarize(number_of_rides=n())
write.csv(cyclitic_plot1,“C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot1.csv”, row.names=FALSE)
```

```
cyclitic_plot1 <-read_csv("cyclitic_plot1.csv")
```

```
## Rows: 14 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, day_of_week
## dbl (1): number_of_rides
##
## 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.
```

```
cyclitic_plot1 %>%
  ggplot(aes(x = day_of_week,y = number_of_rides, fill= member_casual)) +
         geom_col(position = "dodge") +
         labs(title = "Number of Rides in days of week: Casual vs. Member Riders")+
         scale_x_discrete(limits = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sun
         scale_y_continuous(labels = comma)
```



bar chart plot of number of rides for casual and annual member in last 12 months:

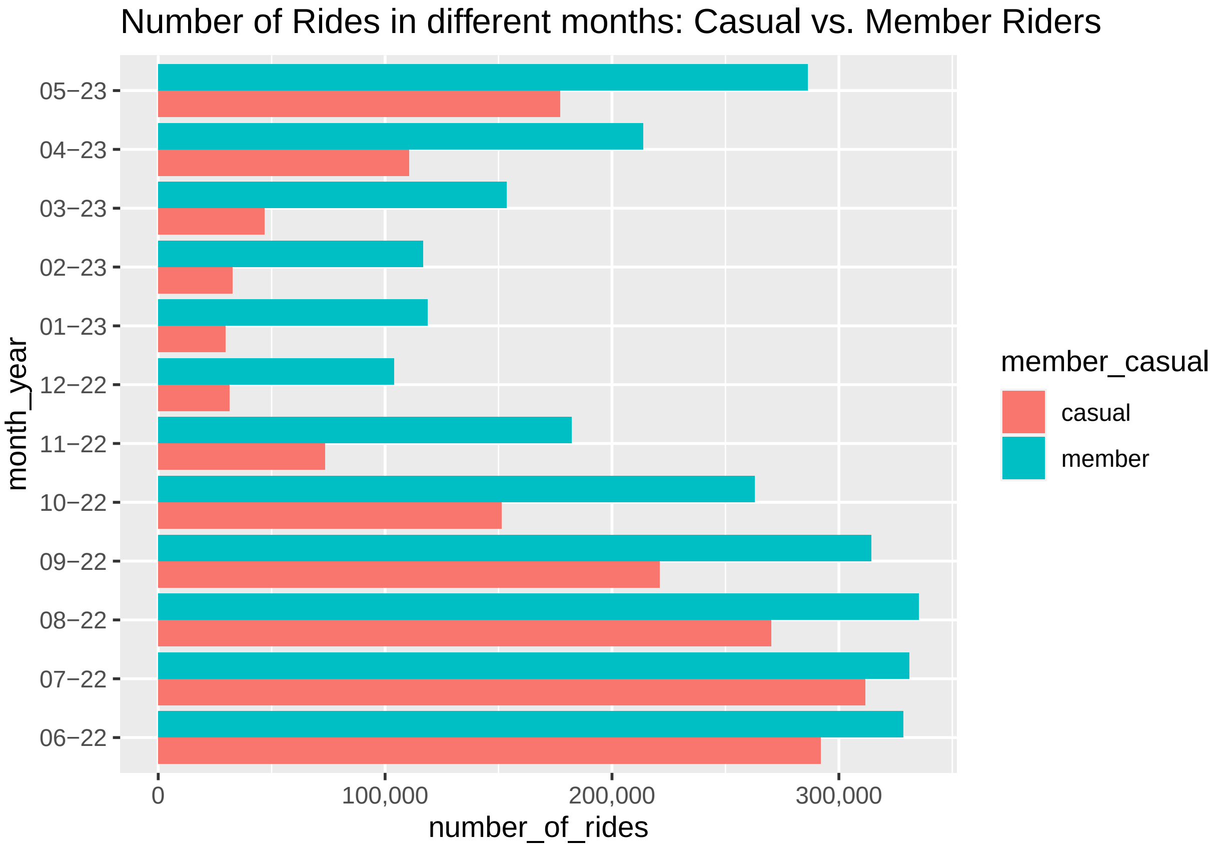
cyclitic$month_year <- format(started_at, format="%m-%y")

cyclitic_plot2 <- cyclitic %>%
group_by(member_casual,month_year) %>%
summarize(number_of_rides=n())
write.csv(cyclitic_plot2,"C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot2.csv", row.names=FALSE)

```
cyclitic_plot2 <-read_csv("cyclitic_plot2.csv")
```

```
## Rows: 24 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, month_year
## dbl (1): number_of_rides
##
## 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.
```

```
cyclitic_plot2 %>%
  ggplot(aes(x = month_year,y = number_of_rides, fill = member_casual)) +
         geom_col(position = "dodge") +
         labs(title = "Number of Rides in different months: Casual vs. Member Riders")+
         scale_x_discrete(limits = c("06-22","07-22","08-22","09-22","10-22","11-22","12-22","01-23","0
         scale_y_continuous(labels = comma)+
         coord_flip()
```



Average ride length (in seconds) during days of week: Casual vs. Member Riders:

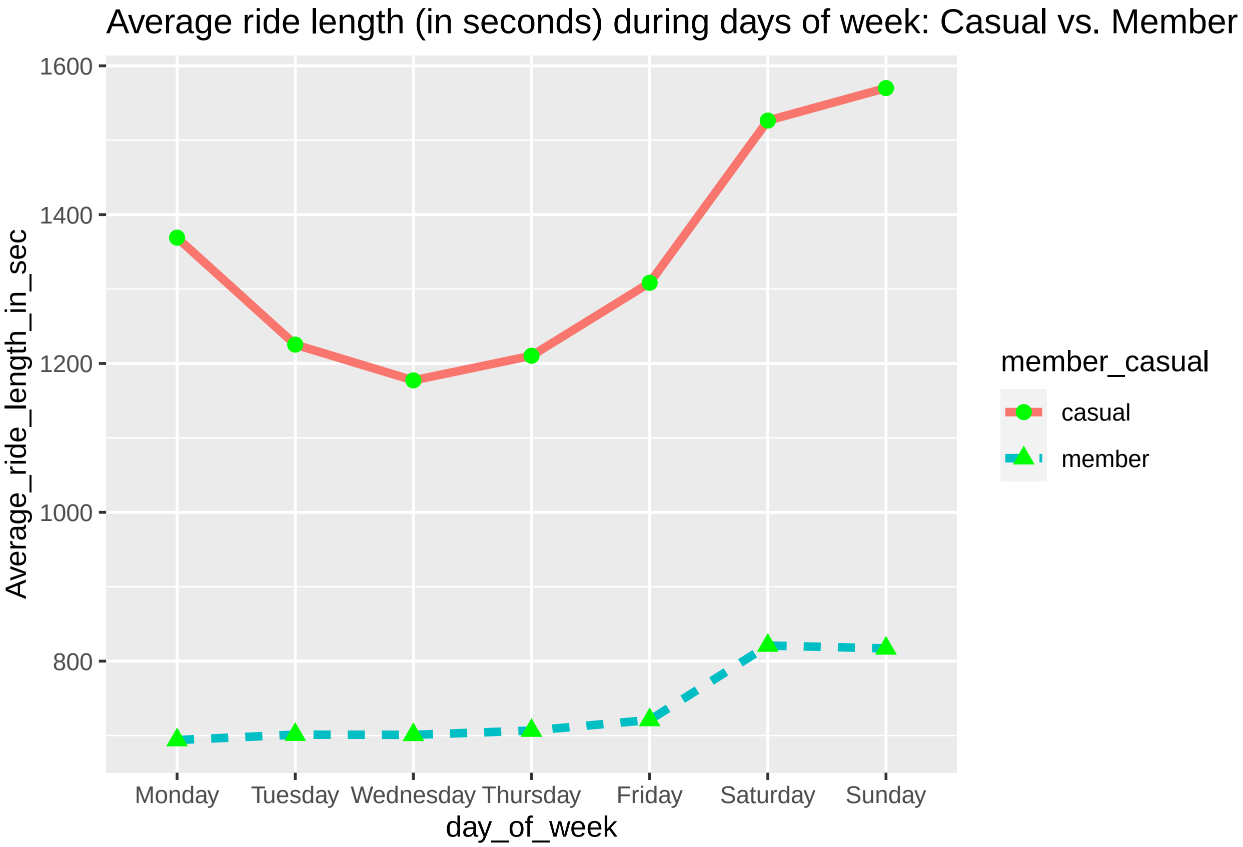
cyclitic_plot3 <- cyclitic %>%
group_by(member_casual,day_of_week) %>%
summarize(Average_ride_length_in_sec=mean(ride_length))
write.csv(cyclitic_plot3,“C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot3.csv”, row.names=FALSE)

```
cyclitic_plot3 <-read_csv("cyclitic_plot3.csv")
```

```
## Rows: 14 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, day_of_week
## dbl (1): Average_ride_length_in_sec
##
## 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.
```

```
cyclitic_plot3 %>%
   ggplot(aes(x=day_of_week, y=Average_ride_length_in_sec, group=member_casual)) +
   geom_line(aes(linetype=member_casual,color=member_casual),size=1.5)+
   geom_point(aes(shape=member_casual),color="green",size=2.5)+
   labs(title = "Average ride length (in seconds) during days of week: Casual vs. Member Riders")+
   scale_x_discrete(limits = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



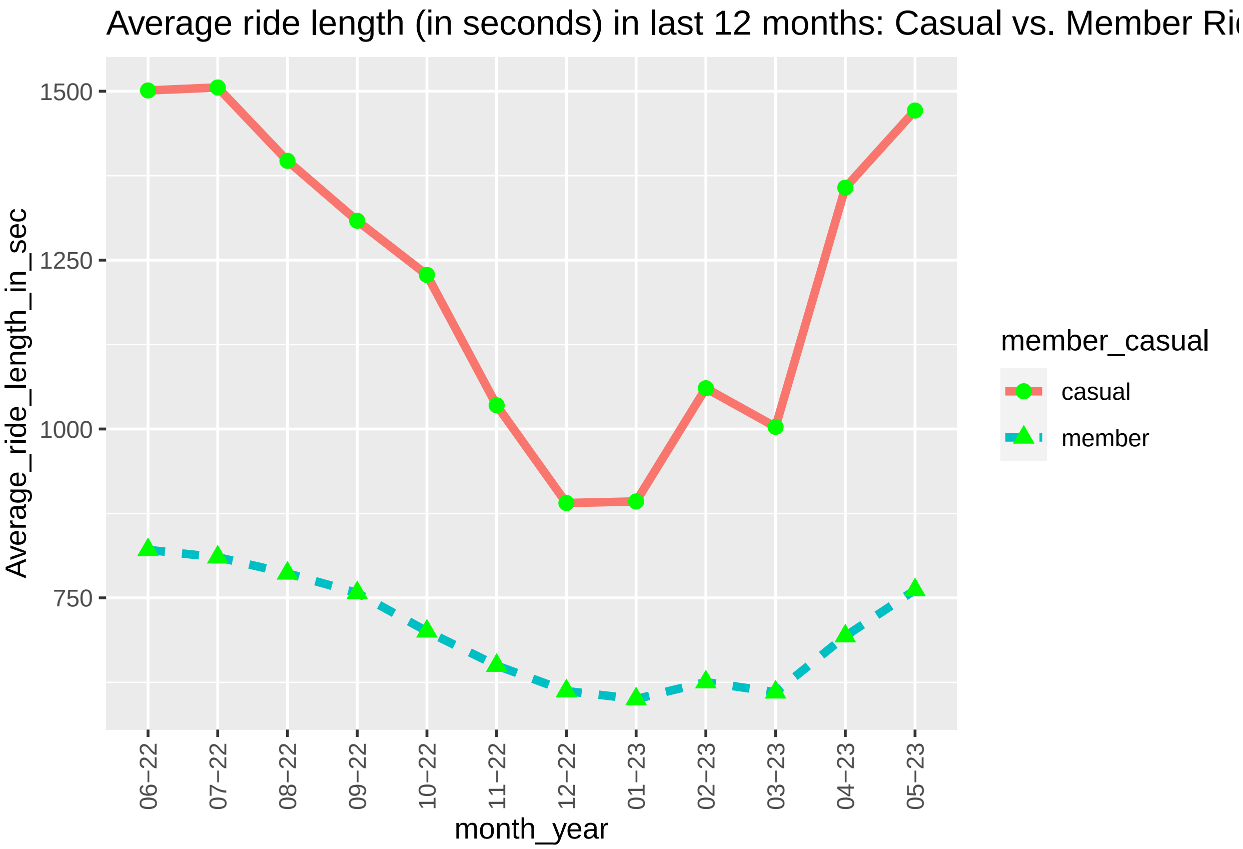
Average ride length (in seconds) in last 12 months: Casual vs. Member Riders:

cyclitic_plot4 <- cyclitic %>%
group_by(member_casual,month_year) %>%
summarize(Average_ride_length_in_sec=mean(ride_length))
write.csv(cyclitic_plot4,“C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot4.csv”, row.names=FALSE)

```
cyclitic_plot4 <-read_csv("cyclitic_plot4.csv")
```

```
## Rows: 24 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, month_year
## dbl (1): Average_ride_length_in_sec
##
## 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.
```

```
cyclitic_plot4 %>%
   ggplot(aes(x=month_year, y=Average_ride_length_in_sec, group=member_casual)) +
   geom_line(aes(linetype=member_casual,color=member_casual),size=1.5)+
   geom_point(aes(shape=member_casual),color="green",size=2.5)+
   labs(title = "Average ride length (in seconds) in last 12 months: Casual vs. Member Riders")+
   scale_x_discrete(limits = c("06-22","07-22","08-22","09-22","10-22","11-22","12-22","01-23","02-23",
   theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



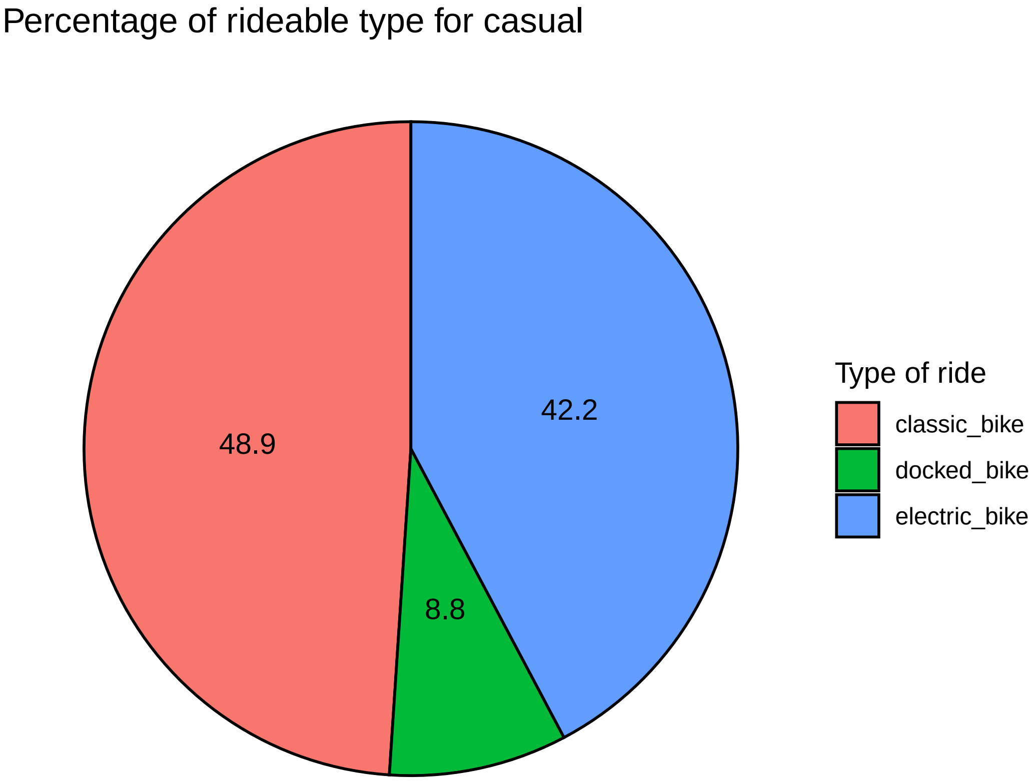
Pie char shows the percentage of rideable type for casual:

```
cyclitic_plot5 <- cyclitic %>%
group_by(member_casual,rideable_type) %>%
summarize(fequency_rideable_type_by_member=n()) %>%
mutate(percentage=(fequency_rideable_type_by_member/sum(fequency_rideable_type_by_member))*100)
%>%
filter(member_casual==“casual”)
cyclitic_plot5$percentage <- round(percentage,digits=1)
write.csv(cyclitic_plot5,“C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot5.csv”, row.names=FALSE)
```

```
cyclitic_plot5 <-read_csv("cyclitic_plot5.csv")
```

```
## Rows: 3 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, rideable_type
## dbl (2): fequency_rideable_type_by_member, percentage
##
## 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.
```

```
cyclitic_plot5 %>%
  ggplot(aes(x = "", y = percentage, fill = rideable_type)) +
  geom_col(color = "black") +
  geom_text(aes(label = percentage),
            position = position_stack(vjust = 0.5))+
  guides(fill = guide_legend(title = "Type of ride"))+
  labs(title="Percentage of rideable type for casual")+
  coord_polar(theta = "y") +
  theme_void()
```



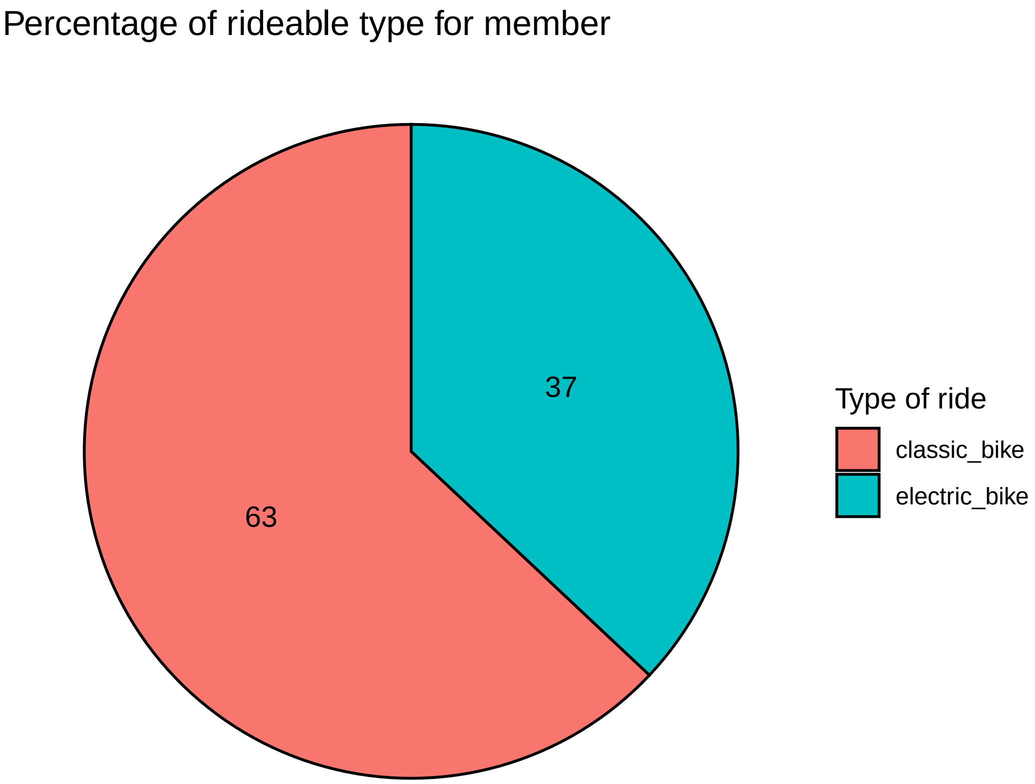
Pie char shows the percentage of rideable type for member:

```
cyclitic_plot6 <- cyclitic %>%
group_by(member_casual,rideable_type) %>%
summarize(fequency_rideable_type_by_member=n()) %>%
mutate(percentage=(fequency_rideable_type_by_member/sum(fequency_rideable_type_by_member))*100)
%>%
filter(member_casual==“member”)
cyclitic_plot6$percentage <- round(percentage,digits=1)
write.csv(cyclitic_plot6,“C:/Users/Toshiba/Desktop/Cyclitic_Trip_Data/cyclitic_plot6.csv”, row.names=FALSE)
```

```
cyclitic_plot6 <-read_csv("cyclitic_plot6.csv")
```

```
## Rows: 2 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): member_casual, rideable_type
## dbl (2): fequency_rideable_type_by_member, percentage
##
## 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.
```

```
cyclitic_plot6 %>%
  ggplot(aes(x = "", y = percentage, fill = rideable_type)) +
  geom_col(color = "black") +
  geom_text(aes(label = percentage),
          position = position_stack(vjust = 0.5))+
  guides(fill = guide_legend(title = "Type of ride"))+
  labs(title="Percentage of rideable type for member")+
  coord_polar(theta = "y") +
  theme_void()
```



In summary:
- The ride length average (in seconds) of casual riders is greater than that of member riders on all days and months. But the number of ride of casual riders is less than of that member on all months and days of week except on Saturday.
- The casual riders have maximum number of rides and maximum average ride length in weekend and the months of spring and summer (04-05-06-07-08-09-10). But for the member riders, they have maximum rides and maximum average length in weekdays.
- Both casual and member riders prefer classic and electric bikes but the docked bike is used only by casual riders.

6. Act

Finally, we can summarize our findings by the following recommendations:

- recommendation 1:
Cyclitic provides a discount (almost 20 %) in weekend for entertainment events in city where the started or ended station of ride is. The annual members can only benefit from this discount. Cyclitic company offers the upcoming events on its account on social media like instagram, twitter, facebook...

- recommendation 2:
In summer and spring months, Cyclitic offers a free service exclusive for annual members where this service involves the rider can take from started station another bike for his partner for free during his ride length.

- recommendation 3:
Cyclitic customizes a monthly draw for annual member who following its pages on social media. This draw offers one annual membership free and the draw is doing online on Cyclitic pages.

- recommendation 3:
Increase the rent payment (almost 10%) of a docked bike on the weekend under the pretext that for the high demand on it.