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

Ask

What is the problem you are trying to solve?
How do annual members and casual riders use Cyclistic bikes differently?

How can your insights drive business decisions?
Recognize how annual members and casual riders differ, would help to design a new marketing strategy to convert casual riders into annual members at converting casual riders into annual members maximizing the number of annual memberships
From these insights, your team will design a new marketing strategy to convert casual riders into annual members.

Key tasks
Business Task
Understand how casual riders and annual members use Cyclistic bikes differently

Key Stakeholders
Cyclistic executive team: Detail-oriented executive team will decide whether to approve the recommended marketing program.
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.
Deliverable
Design marketing strategies aimed at converting casual riders into annual members.

Prepare

Where is your data located?
For this analysis, we have access to a primary data source downloaded from Cyclystic trip data. It is licensed under Motivate International Inc. This dataset consists of quantitative measurements collected from bike trackers; they do not gather personal data from users.

How is the data organized?
This data is organized in Qualitative data is also called categorical data since this data is grouped according to two Cyclistic members categories.

- Customers who purchase single-ride or full-day passes are referred to as **casual riders.**
- Customers who purchase **annual memberships.**

Are there issues with bias or credibility in this data?

It is unbiased data and is ready for analysis.

Reliable – Data is Reliable.
Original - Yes
Comprehensive - Yes
Current – Yes
Cited - Yes

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

License: The data has been made available by Motivate International Inc. under this license.
License URL: https://ride.divvybikes.com/data-license-agreement

Privacy: This is public information that you may use to investigate how different consumer types utilize Cyclistic bikes.

Security: Data-privacy concerns prevent you from using passengers' personally identifying information. See Prohibited Conduct section in the license.

Accessibility: The City of Chicago owns all right, title, and interest in the Data. Bikeshare may modify or cease providing any or all of the Data at any time, without notice, in its sole discretion. For more information, please see the Ownership and Provision of Data section of the license.

How did you verify the data's integrity?
All dataset has identical columns and has the correct type of data.

How does it help you answer your question?
Dataset has relevant data to answer your question.

Are there any problems with the data?
The data has been de-personalized to safeguard the privacy of users. In particular, this means it is not possible to connect past purchases to credit card numbers and determine if casual riders live in the service area or purchased multiple single passes.

Key tasks

credibility of the data.
I looked at the most recent 12 months of data at the time of writing, from April 2022 – May 2023. Since 2021, these files contain the following fields.

ride_id (categorical): Unique number assigned to a ride trip.

rideable_type (categorical): Type of bike used during trip; standard two-wheel bike, reclining bike, hand tricycle, or cargo bike.

started_at (datetime): Start date and time for the trip

ended_at (datetime): End data and time for the trip

start_station_name (categorical): Name of the station where the trip started

start_station_id (categorical): Unique identification code assigned to the start station.

end_station_name (categorical): Name of the station where the trip ended.

end_station_id (categorical): Unique identification code assigned to the end station.

start_lat (numeric): Latitude coordinate of where the trip started.

start_lng (numeric): Longitude coordinate of where the trip started.

end_lat (numeric): Latitude coordinate of where the trip ended.

end_lng (numeric): Longitude coordinate of where the trip ended.

member_casual (categorical): Customer type; “member” = annual member, “casual” = casual rider.

Process

Tools Used:

R Studio was the primary tool for this project. It includes a large range of functions and can easily manage large volumes of data.

Load Libraries
Read dataset

```
4
5  #Load Library
6  library("ggplot2")
7  library("lubridate")
8  library("tidyverse")
9  library("knitr")
10 library("dplyr")
11 options(dplyr.summarise.inform = FALSE)
12
13 #Read datasets
14 df1 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202204-divvy-tripdata.csv")
15 df2 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202205-divvy-tripdata.csv")
16 df3 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202206-divvy-tripdata.csv")
17 df4 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202207-divvy-tripdata.csv")
18 df5 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202208-divvy-tripdata.csv")
19 df6 <- read_csv("/Users/priya/personal/Capstone/Bike-Share/Dataset/file/202209-divvy-tripdata.csv")
5:14  (Top Level)                                                                    R Script
```

```
Console  Terminal  Background Jobs
R 4.2.0 · ~/personal/Capstone/Bike-Share/Dataset/programming/

> library("tidyverse")
── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
✔ dplyr   1.1.2     ✔ stringr 1.5.0
✔ forcats 1.0.0     ✔ tibble  3.2.1
✔ purrr   1.0.1     ✔ tidyr   1.3.0
✔ readr   2.1.4
── Conflicts ──────────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package to force all conflicts to become errors
```

Merge Dataset

```
#Merge Datasets
df_bike <- bind_rows(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
colnames(df_bike)

#datatype of the DataFrame
#str(df_bike)
sapply(df_bike,class)

#summary of the Dataset
summary(df_bike)
fileConn<-file("output.txt")
writeLines(summary(df_bike), fileConn)
close(fileConn)

#Check number of rows
nrow(df_bike)
```

Clean and organize dataset

Format Date Column

```
############ CELAN DATASET ########################
#is.na(df_bike)
colSums(is.na(df_bike))

#drop empty cells
bike_share_clean <- drop_na(df_bike)

#Clean Date
bike_share_clean$date <- as.Date(bike_share_clean$started_at)
bike_share_clean$month <- format(as.Date(bike_share_clean$started_at), "%m")
bike_share_clean$day <-format(as.Date(bike_share_clean$started_at), "%d")
bike_share_clean$year <-format(as.Date(bike_share_clean$started_at), "%y")
bike_share_clean$day_of_week <-format(as.Date(bike_share_clean$started_at), "%A")

#Calculate Ride Duration
bike_share_clean$ride_duration <-difftime(bike_share_clean$ended_at,bike_share_clean$started_at)/60

str(bike_share_clean$ride_duration) #check datatype of ride_duration column

#Convert the ride_duration column to numeric
bike_share_clean$ride_length <-as.numeric(bike_share_clean$ride_duration)

str(bike_share_clean$ride_length) #check datatype of ride_length column
```

```
> #Check number of rows
> nrow(df_bike)
[1] 5803720
> #is.na(df_bike)
> colSums(is.na(df_bike))
          ride_id      rideable_type         started_at           ended_at start_station_name   start_station_id
                0                  0                  0                  0             839082             839214
 end_station_name     end_station_id          start_lat          start_lng            end_lat            end_lng
           896319             896460                  0                  0               5855               5855
    member_casual
                0
```

Filter

	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual	date	month	day	year	day_of_week	ride_duration
	605	42.01913	-87.67353	42.05294	-87.67345	member	2022-04-06	04	06	22	Wednesday	11.80000000 secs
	TA1307000120	41.85308	-87.63193	41.88189	-87.64879	member	2022-04-24	04	24	22	Sunday	20.16666667 secs
	TA1307000120	41.87184	-87.64664	41.88189	-87.64879	member	2022-04-20	04	20	22	Wednesday	6.13333333 secs
	KA1706005007	41.85308	-87.63193	41.86749	-87.63219	casual	2022-04-22	04	22	22	Friday	9.38333333 secs
	TA1305000032	41.87181	-87.64657	41.88224	-87.64107	member	2022-04-16	04	16	22	Saturday	5.68333333 secs
	13011	41.88462	-87.64457	41.87926	-87.63990	member	2022-04-21	04	21	22	Thursday	4.30000000 secs
	13011	41.88462	-87.64457	41.87926	-87.63990	member	2022-04-04	04	04	22	Monday	4.63333333 secs
	TA1306000015	41.87771	-87.63532	41.88578	-87.65102	member	2022-04-05	04	05	22	Tuesday	12.40000000 secs
e	331	41.90966	-87.64812	41.90967	-87.64813	member	2022-04-29	04	29	22	Friday	0.53333333 secs
e	331	41.90953	-87.64808	41.90967	-87.64813	member	2022-04-29	04	29	22	Friday	1.48333333 secs
	13102	41.85313	-87.63187	41.85761	-87.61941	casual	2022-04-23	04	23	22	Saturday	13.76666667 secs
St	TA1307000115	41.90967	-87.64813	41.90461	-87.64055	member	2022-04-13	04	13	22	Wednesday	3.48333333 secs
	TA1309000001	41.87184	-87.64664	41.86488	-87.64707	casual	2022-04-13	04	13	22	Wednesday	5.20000000 secs
	13016	41.87613	-87.62974	41.89435	-87.62280	casual	2022-04-02	04	02	22	Saturday	10.28333333 secs
	TA1307000130	41.87184	-87.64664	41.87174	-87.65103	member	2022-04-06	04	06	22	Wednesday	3.10000000 secs
	15653	41.87184	-87.64664	41.90977	-87.70528	member	2022-04-11	04	11	22	Monday	31.35000000 secs

Wednesday

Analyze

Descriptive analysis

Summarize Data

```
#Summarize the Data
mean(bike_share_clean$ride_length)
max(bike_share_clean$ride_length)
min(bike_share_clean$ride_length)
median(bike_share_clean$ride_length)

summary(bike_share_clean$ride_length)

#aggregating members_casual and ride_length(in seconds)
aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual, FUN = mean)


bike_share_clean %>%
  group_by(member_casual, day_of_week) %>%
  aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual+ day_of_week, FUN = mean)
  ggplot(data=bike_share_clean, aes(x=day_of_week, y=ride_length, fill = member_casual)) + geom_bar(stat="identity")

bike_share_clean %>%
  aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual+ day_of_week, FUN = max)

bike_share_clean %>%
  aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual+ day_of_week, FUN = min)

#Summarize no of rides by weekdays
bike_share_clean %>%
  group_by(day_of_week,member_casual) %>%
  summarise(number_of_ride = n()) %>%
  arrange(day_of_week)
```

Average ride_length by user type

```
 group_by(member_casual, day_of_week) %>%
 aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual+ day_of_week, FUN = mean)
bike_share_clean$member_casual day_of_week bike_share_clean$ride_length
                        casual      Friday                     22.02221
                        member      Friday                     12.03466
                        casual      Monday                     23.88877
                        member      Monday                     11.78626
#Summarize no of rides by weekdays
bike_share_clean %>%
  group_by(day_of_week,member_casual) %>%
  summarise(number_of_ride = n()) %>%
  arrange(day_of_week)

#Summarize no of rides by month
bike_share_clean %>%
  group_by(month) %>%
  summarise(ride_len = sum(ride_length)) %>%
  arrange(month)

##Summarize no of rides by membership
bike_share_clean %>%
  group_by(member_casual) %>%
  summarise(no_of_rides = length(ride_length)) %>%
  arrange(member_casual)
```

```
> #aggregating members_casual and ride_length(in seconds)
> aggregate(bike_share_clean$ride_length ~ bike_share_clean$member_casual, FUN = mean)
  bike_share_clean$member_casual bike_share_clean$ride_length
1                         casual                     23.33149
2                         member                     12.26705
>
```

Summarize no of rides by weekdays

```
> #summary of the Dataset
> summary(df_bike)
   ride_id          rideable_type        started_at                      ended_at                    start_station_name
 Length:5803720     Length:5803720     Min.   :2022-04-01 00:01:48.00   Min.   :2022-04-01 00:02:15.00   Length:5803720
 Class :character   Class :character   1st Qu.:2022-06-18 23:27:00.25   1st Qu.:2022-06-18 23:51:55.75   Class :character
 Mode  :character   Mode  :character   Median :2022-08-13 11:37:32.00   Median :2022-08-13 12:00:07.50   Mode  :character
                                       Mean   :2022-08-25 07:04:55.95   Mean   :2022-08-25 07:23:54.70
                                       3rd Qu.:2022-10-14 18:04:21.00   3rd Qu.:2022-10-14 18:19:10.25
                                       Max.   :2023-03-31 23:59:28.00   Max.   :2023-04-03 11:41:11.00

 start_station_id   end_station_name   end_station_id       start_lat       start_lng         end_lat          end_lng
 Length:5803720     Length:5803720     Length:5803720     Min.   :41.64   Min.   :-87.84   Min.   : 0.00   Min.   :-88.14
 Class :character   Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
 Mode  :character   Mode  :character   Mode  :character   Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
                                                          Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
                                                          3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
                                                          Max.   :42.07   Max.   :-87.52   Max.   :42.37   Max.   :  0.00
                                                                                           NA's   :5855    NA's   :5855
 member_casual
 Length:5803720
 Class :character
 Mode  :character
```

```
> bike_share_clean %>%
+   group_by(day_of_week,member_casual) %>%
+   summarise(number_of_ride = n()) %>%
+   arrange(day_of_week)
# A tibble: 14 × 3
# Groups:   day_of_week [7]
   day_of_week member_casual number_of_ride
   <chr>       <chr>                  <int>
 1 Friday      casual                253859
 2 Friday      member                378111
 3 Monday      casual                209510
 4 Monday      member                384815
 5 Saturday    casual                365805
 6 Saturday    member                345042
 7 Sunday      casual                301980
 8 Sunday      member                306767
 9 Thursday    casual                232497
10 Thursday    member                431252
11 Tuesday     casual                202967
12 Tuesday     member                433013
13 Wednesday   casual                205855
14 Wednesday   member                430889
>
```

Summarize no of rides by every month

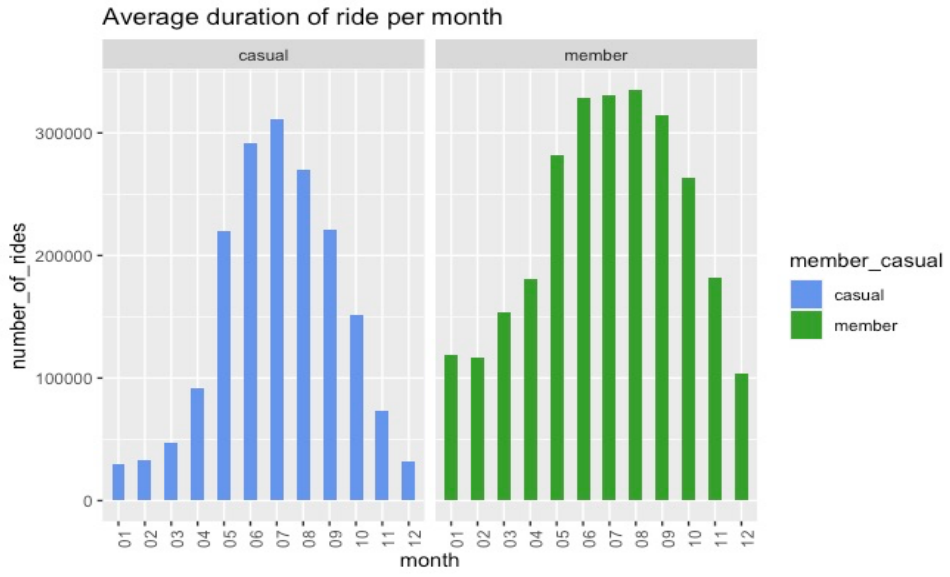
```
bike_share_clean %>%
  group_by(member_casual,month) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual,month) %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual )) +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_col(width=0.5, position = position_dodge(width=0.5)) +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  labs(title = "Average duration of ride per month") + facet_wrap(~member_casual)+
  scale_fill_manual(values = c("#6495ED", "#33A02C"))
```

```
> bike_share_clean %>%
+   group_by(month) %>%
+   summarise(ride_len = sum(ride_length)) %>% arrange(month)
# A tibble: 12 × 2
   month  ride_len
   <chr>     <dbl>
 1 01      1627889.
 2 02      1796350.
 3 03      2345128.
 4 04      4481824.
 5 05      9856679.
 6 06     11799033.
 7 07     12290303.
 8 08     10682306.
 9 09      8782948.
10 10      6166945.
11 11      3240026.
```

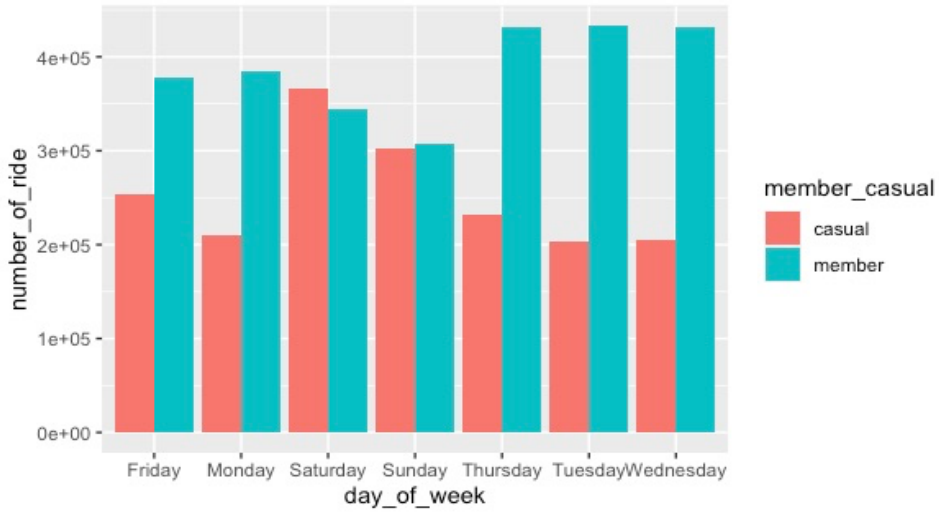
Share

Visualize data

Average ride duration of member and casual rider per month



Average duration of rider per day of the week

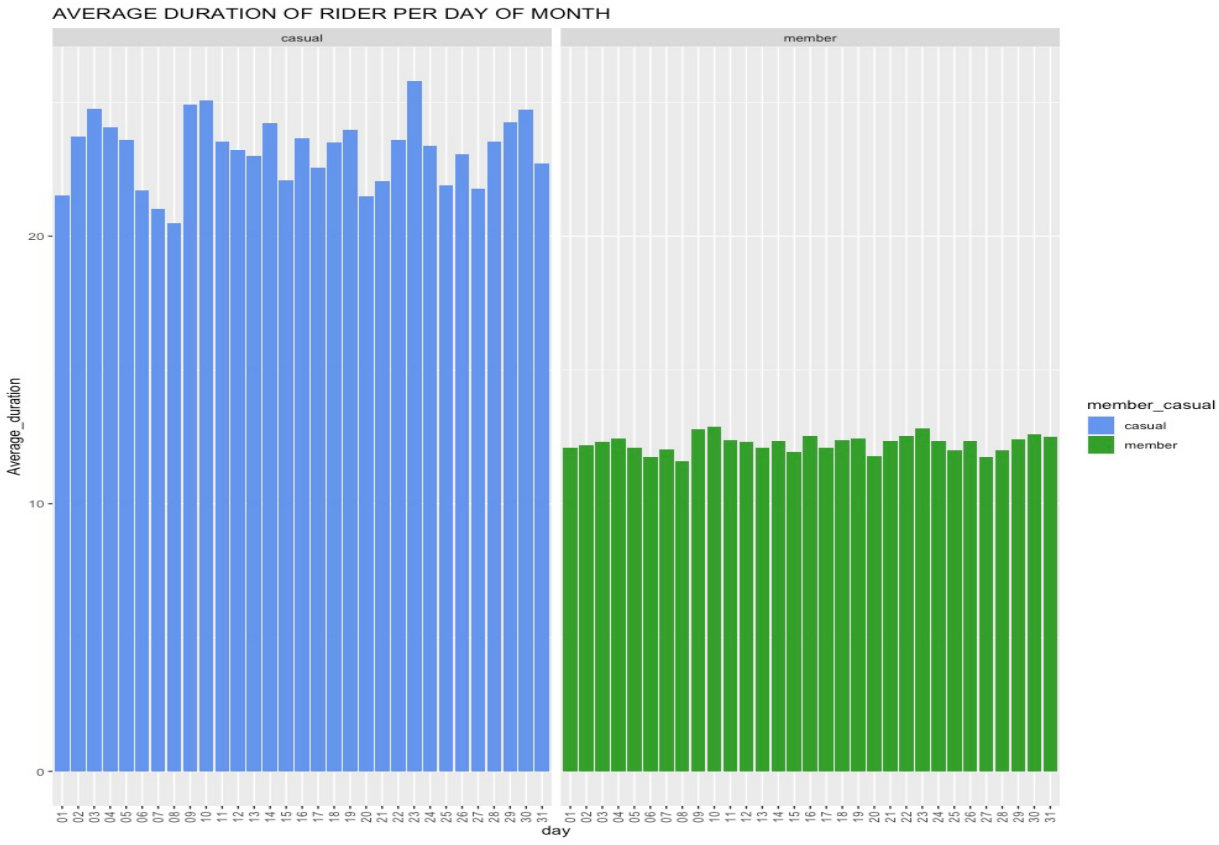


Average ride duration of member and casual rider per day of the Month

```
> week_day_ride <- as.data.frame(bike_share_clean %>%
+                                  group_by(day_of_week,member_casual) %>%
+                                  summarise(number_of_ride = n()))
> week_day_ride$number_of_ride <-as.numeric(week_day_ride$number_of_ride)
> sapply(week_day_ride,class)
    day_of_week  member_casual number_of_ride
    "character"    "character"      "numeric"
```

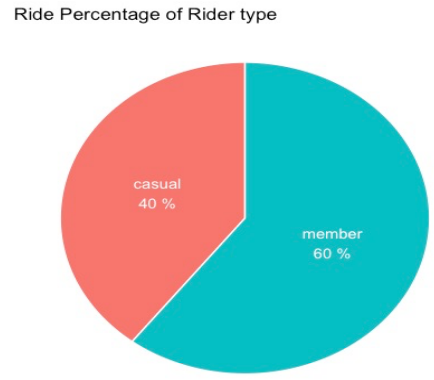
```
ggplot(week_day_ride, aes(x = day_of_week, y = number_of_ride, fill = member_casual))
  geom_bar(stat = "identity",position="dodge")
```

```
bike_share_clean %>% group_by(member_casual, day) %>%
  summarise(Average_duration = mean(ride_duration), .groups = 'drop') %>%
  ggplot(aes(x = day, y =Average_duration, fill = member_casual )) +
  geom_col(position = "dodge") + labs(title = "AVERAGE DURATION OF RIDER PER DAY")+
  facet_wrap(~member_casual) + scale_fill_manual(values = c("#6495ED", "#33A02C"))
```



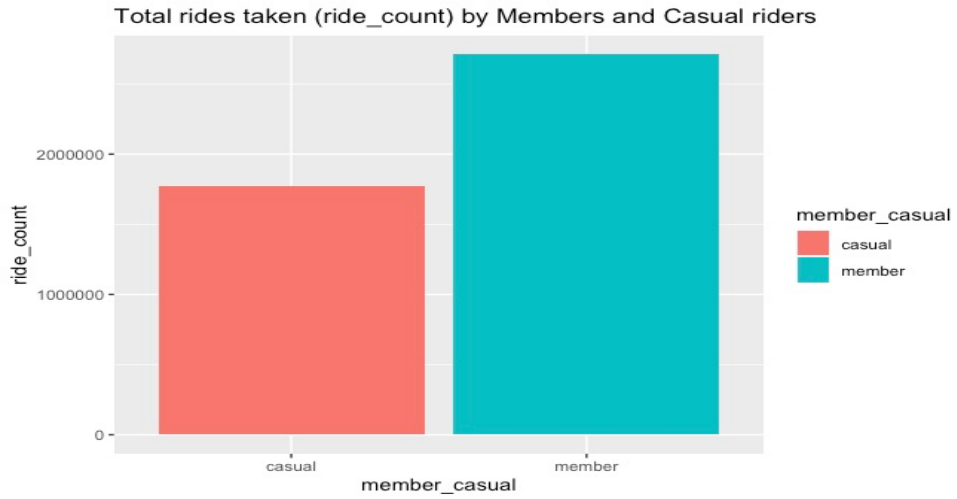
Rider type in percentage

```
bike_share_clean %>% group_by(member_casual) %>% summarize(number_of_rides = n()) %>%
  mutate(percentage = round(number_of_rides*100/ sum(number_of_rides))) %>%
  ggplot(aes(x = "", y = number_of_rides, fill = member_casual)) +
  geom_bar(width = 1, stat="identity", color = "white", show.legend = FALSE) +
  coord_polar("y", start=0) + geom_text(aes(label = paste(member_casual,paste(percentage,"%"), sep = "\n")), position = position_
  labs(title = "Ride Percentage of Rider type ") + theme_void()
```



Total ride count taken by Members and Casual riders

```
bike_share_clean %>%
  group_by(member_casual) %>%
  summarise(ride_count = length(ride_id)) %>%
  arrange(member_casual) %>%
  ggplot(aes(x = member_casual,y = ride_count,fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(title = "Total rides taken (ride_count) by Members and Casual riders")+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```



Act

After analyzing, we could discover insights that can assist organizations in finding the best method to convert casuals into members through data-driven decisions.

The total trip duration for casual riders and annual members are affected by the season. We could see that average duration of ride of casual and member are high in April, May, June and July.

The temperature is very low during the winter season, fewer people are willing to go out and people who need to travel daily for work will choose to take other public transport, this had caused the total trip duration are the lowest among another season.

We understand that casual members are taking longer ride in weekend motorcyclists will take longer rides.

Recommendation

Cyclistic may collaborate with organizations to offer unique weekend activities/or perks to entice people to become members.
Offer occasional membership subscription discount to casual riders on summer and weekends.
Encouraging casual users to have membership by lowering the renting price of bikes (member's deal) for weekend.