Case Study

How Does a Bike-Share Navigate Speedy Success

August 12, 2022

Introduction

This is a Capstone project requirement for Google Data Analytics Professional Certificate. For this case study 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, Our team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, Our 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. I am going to analyze customer's trip details over a 12 month period (July 2021 - June 2022)

Stage 1

1.1 Business understanding

Cyclistic is bike sharing company based on Chicago it has 5,824 bicycles that are geo tracked and locked into a network of 692 stations. cyclistic has two kind of customers casual riders who purchase bike for single ride or full day pass and another kind of customer is member rider who purchase annual membership.

1.2 Business task

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno(The director of marketing) believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Morenobelieves there is a very good chance to convert casual riders into members.

Exploring the Business task into problem statement for finding solution.

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?

1.3 Business task assigned to marketing analyst team for finding the business solutions.

How do annual members and casual riders use Cyclistic bikes differently

Stage 2

2.1 Prepare data for exploration

After understanding business task we move to colect,organize,store and check the crediblity of data.

key tasks

1. Download data and store it appropriately.

2. Identify how its organized.

3. Short and filter data.

4. Determine the credibility of data

2.2 Download data and store it appropriately.

Cyclistic Recent 12 month bike ride data has been downloaded from here https://divvy-tripdata.s3.amazonaws.com/index.html

12 months of bike data has been stored on respective path.

C: \Users\Praveen\Onedrive\Desktop\Case_study_bike_share\Bike_share_months_data

2.3 Identify the data how its organized and need to import data sets to R studio.

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

Installing tidyverse package which is esensial for data analysis in R

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Admin\AppData\Local\Temp\Rtmp650wtF\downloaded_packages

library(tidyverse)
```

```
library(readr)
```

```
Jul=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Aug=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Sep=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Oct=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Nov=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Dec=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Jan=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Feb=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Mar=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Apr=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
May=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

```
Jun=read.csv("C:\\Users\\Praveen\\Onedrive\\Desktop\\Case_study_bike_share\\Bike_share_months_data\\202
```

2.4 Importing CSV files from respective path

```
str(Jul)
```

2.5 Undestanding data type such as number, text or string, boolean with the help str() function which is help to display even the internal structure of large lists which are nested.

```
## 'data.frame':    822410 obs. of  13 variables:
##  $ ride_id           : chr  "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8A
##  $ rideable_type     : chr  "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-07-02 14:44:36" "2021-07-07 16:57:42" "2021-07-25 11:30:55" "2021-0
##  $ ended_at          : chr  "2021-07-02 15:19:58" "2021-07-07 17:16:09" "2021-07-25 11:48:45" "2021-0
##  $ start_station_name: chr  "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave &
##  $ start_station_id  : chr  "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr  "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbard
##  $ end_station_id    : chr  "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "member" "member" ...
```

Understanding dataset

```
colnames(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(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"
```

2.5 Comparing column name of dataset to combine all datasets.

```
install.packages("Janitor")
```

Installing required packages.

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## 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: Perhaps you meant 'janitor' ?
```

```
library(janitor)
```

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

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

Here comparing the all dataset columns

```
compare_df_cols(Jul,Aug,Sep,Oct,Nov,Dec,Jan,Feb,Mar,Apr,May,Jun)
```

```
##           column_name       Jul       Aug       Sep       Oct       Nov
## 1            end_lat   numeric   numeric   numeric   numeric   numeric
## 2            end_lng   numeric   numeric   numeric   numeric   numeric
## 3      end_station_id character character character character character
## 4    end_station_name character character character character character
## 5            ended_at character character character character character
## 6       member_casual character character character character character
## 7             ride_id character character character character character
## 8       rideable_type character character character character character
## 9           start_lat   numeric   numeric   numeric   numeric   numeric
## 10          start_lng   numeric   numeric   numeric   numeric   numeric
## 11   start_station_id character character character character character
## 12 start_station_name character character character character character
## 13         started_at character character character character character
##          Dec       Jan       Feb       Mar       Apr       May       Jun
## 1    numeric   numeric   numeric   numeric   numeric   numeric   numeric
## 2    numeric   numeric   numeric   numeric   numeric   numeric   numeric
## 3  character character character character character character character
## 4  character character character character character character character
## 5  character character character character character character character
## 6  character character character character character character character
## 7  character character character character character character character
## 8  character character character character character character character
## 9    numeric   numeric   numeric   numeric   numeric   numeric   numeric
## 10   numeric   numeric   numeric   numeric   numeric   numeric   numeric
## 11 character character character character character character character
## 12 character character character character character character character
## 13 character character character character character character character
```

```
Yeardata<-rbind(Jul,Aug,Sep,Oct,Nov,Dec,Jan,Feb,Mar,Apr,May,Jun)
```

2.6 In all data sets each column and data types are similar so appropriate to combine all data sets.

```
head(Yeardata)
```

Rechecking the compined data sets by using head and tail function which are help to understand below and top of data sets column name and data type.

```
##            ride_id rideable_type          started_at            ended_at
## 1 0A1B623926EF4E16   docked_bike 2021-07-02 14:44:36 2021-07-02 15:19:58
## 2 B2D5583A5A5E76EE  classic_bike 2021-07-07 16:57:42 2021-07-07 17:16:09
## 3 6F264597DDBF427A  classic_bike 2021-07-25 11:30:55 2021-07-25 11:48:45
## 4 379B58EAB20E8AA5  classic_bike 2021-07-08 22:08:30 2021-07-08 22:23:32
## 5 6615C1E4EB08E8FB electric_bike 2021-07-28 16:08:06 2021-07-28 16:27:09
## 6 62DC2B32872F9BA8 electric_bike 2021-07-29 17:09:08 2021-07-29 17:15:00
##             start_station_name start_station_id               end_station_name
## 1 Michigan Ave & Washington St            13001   Halsted St & North Branch St
## 2   California Ave & Cortez St            17660           Wood St & Hubbard St
## 3         Wabash Ave & 16th St           SL-012           Rush St & Hubbard St
## 4   California Ave & Cortez St            17660        Carpenter St & Huron St
## 5   California Ave & Cortez St            17660 Elizabeth (May) St & Fulton St
## 6   California Ave & Cortez St            17660  Albany Ave & Bloomingdale Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   KA1504000117  41.88398 -87.62468 41.89937 -87.64848        casual
## 2          13432  41.90036 -87.69670 41.88990 -87.67147        casual
## 3   KA1503000044  41.86038 -87.62581 41.89017 -87.62619        member
## 4          13196  41.90036 -87.69670 41.89456 -87.65345        member
## 5          13197  41.90035 -87.69668 41.88659 -87.65839        casual
## 6          15655  41.90033 -87.69674 41.91389 -87.70513        casual
```

```
tail(Yeardata)
```

```
##                  ride_id rideable_type          started_at            ended_at
## 5900380 F1C4F671FE0FDDD1 electric_bike 2022-06-26 19:50:04 2022-06-26 19:55:23
## 5900381 7B3B2890CA85E05D  classic_bike 2022-06-25 00:56:48 2022-06-25 01:01:39
## 5900382 1E993989CC66BCBC  classic_bike 2022-06-25 00:56:25 2022-06-25 01:00:26
## 5900383 AEA166296BC67566 electric_bike 2022-06-12 12:47:12 2022-06-12 12:47:36
## 5900384 B9F527742959CF03  classic_bike 2022-06-12 13:28:46 2022-06-12 13:53:11
## 5900385 D241310352E26484  classic_bike 2022-06-12 14:40:51 2022-06-12 15:08:14
##                    start_station_name start_station_id
## 5900380    Clarendon Ave & Junior Ter            13389
## 5900381  Sheffield Ave & Kingsbury St            13154
## 5900382  Sheffield Ave & Kingsbury St            13154
## 5900383 Milwaukee Ave & Fullerton Ave              428
## 5900384        Clark St & Randolph St     TA1305000030
## 5900385     Blue Island Ave & 18th St            13135
```

```
##                      end_station_name end_station_id start_lat start_lng
## 5900380    Clarendon Ave & Junior Ter          13389  41.96100 -87.64946
## 5900381  Sheffield Ave & Kingsbury St          13154  41.91052 -87.65311
## 5900382  Sheffield Ave & Kingsbury St          13154  41.91052 -87.65311
## 5900383 Milwaukee Ave & Fullerton Ave            428  41.92000 -87.70000
## 5900384        Clark St & Randolph St   TA1305000030  41.88458 -87.63189
## 5900385     Blue Island Ave & 18th St          13135  41.85756 -87.66154
##          end_lat   end_lng member_casual
## 5900380 41.96100 -87.64960        member
## 5900381 41.91052 -87.65311        member
## 5900382 41.91052 -87.65311        member
## 5900383 41.92000 -87.70000        casual
## 5900384 41.88458 -87.63189        casual
## 5900385 41.85756 -87.66154        casual
```

2.7 Determine the credibility of data we use ROCCC method to identify data credibility

R & O - Reliable and original: Data is originaly collected by cyclistic its primary source and original.

C-Comprehensive : Data has important formation to solve problem so its comprehensive

C- Current : Data is not outdated its current data.

C- Cited : As data is maintained and trusted by cyclistic its cited data.

stage 3

3 Process.

In data analysis steps this one is very imported.In this steps using some functions for cleaning data and organized in a proper way.

3.1 Following are some key tasks of data analysis process stage

1.Check the data for errors.

2.Check for duplicate data.

3.Organize the data in a appropriate way

4. Choose your tools.

5. Transform the data so you can work with it effectively.

6. Document the cleaning process.

3.2 For Cleaning and organising data in structured format so we have to install required packages.

```
install.packages("readr")
```

```
## Warning: package 'readr' is in use and will not be installed
```

```
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\Admin\AppData\Local\R\win-library\4.2\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\Admin\AppData\Local\R\win-library\4.2\dplyr\libs\x64\dplyr.dll:
## Permission denied

## Warning: restored 'dplyr'

##
## The downloaded binary packages are in
##  C:\Users\Admin\AppData\Local\Temp\Rtmp650wtF\downloaded_packages
```

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
install.packages("skimr")
```

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'skimr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Admin\AppData\Local\Temp\Rtmp650wtF\downloaded_packages
```

```
install.packages("here")
```

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'here' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Admin\AppData\Local\Temp\Rtmp650wtF\downloaded_packages
```

```
library(skimr)
library(here)
```

```
## here() starts at G:/case study
```

```
library(lubridate)
```

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

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

3.3 Checking for error , duplicate and treat null values

```
num_duplicates <- sum(duplicated(Yeardata$ride_id))

if (num_duplicates > 0) {
    CombinedData <- Yeardata %>%
        distinct(ride_id, .keep_all = TRUE)
    }

print(paste("Number of duplicates removed: ",num_duplicates))
```

Ensure there are no duplicates. The 'ride_id' variable will be checked to ensure that there are not any duplicate entries that could impact the results.

```
## [1] "Number of duplicates removed:  0"
```

```
skim_without_charts(Yeardata)
```

Table 1: Data summary

Name	Yeardata
Number of rows	5900385

Table 1: Data summary

Number of columns	13
Column type frequency:	
character	9
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	5900385	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	4924385	0
ended_at	0	1	19	19	0	4924865	0
start_station_name	0	1	0	64	836018	1294	0
start_station_id	0	1	0	44	836015	1158	0
end_station_name	0	1	0	64	892103	1316	0
end_station_id	0	1	0	44	892103	1172	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	45.64
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-73.80
end_lat	5374	1	41.90	0.05	41.39	41.88	41.90	41.93	42.17
end_lng	5374	1	-87.65	0.03	-88.97	-87.66	-87.64	-87.63	-87.49

```
Yeardata%>%
distinct(.keep_all = TRUE) %>%
skim_without_charts()
```

Table 4: Data summary

Name	Piped data
Number of rows	5900385
Number of columns	13
Column type frequency:	
character	9
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	5900385	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	4924385	0
ended_at	0	1	19	19	0	4924865	0
start_station_name	0	1	0	64	836018	1294	0
start_station_id	0	1	0	44	836015	1158	0
end_station_name	0	1	0	64	892103	1316	0
end_station_id	0	1	0	44	892103	1172	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	45.64
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-73.80
end_lat	5374	1	41.90	0.05	41.39	41.88	41.90	41.93	42.17
end_lng	5374	1	-87.65	0.03	-88.97	-87.66	-87.64	-87.63	-87.49

When we execute both skim_without_charts(Yeardata) and union_df %>% distinct(.keep_all = TRUE) %>% kim_without_charts() rows count remain same 5900385 so we finalized there is no duplicate rows.

```
table(Yeardata$member_casual)
```

3.4 How many observations fall under each rider type.

```
##
##  casual  member
## 2558227 3342158
```

Verify any missing or null values.

```
class(Yeardata$started_at)
```

Checking data type in started_at and ended_at column

```
## [1] "character"
```

```
class(Yeardata$ended_at)
```

```
## [1] "character"
```

3.5 Covertion of factor into 'POSIXct' 'POSIXt'(these are date formate includes for ymd_hms) format.

```
Yeardata$started_at=ymd_hms(Yeardata$started_at)
Yeardata$ended_at=ymd_hms(Yeardata$ended_at)
```

```
class(Yeardata$started_at)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(Yeardata$ended_at)
```

```
## [1] "POSIXct" "POSIXt"
```

```
Yeardata$weekday=weekdays(Yeardata$started_at)
```

Now its ready to extract date and time related information from this columns extraction of day name into new columns weekday as per our business task we might require weekday from started_at column

```
Yeardata$month=months(Yeardata$started_at)
```

By using months function adding new column as month

Checking data sets.

```
 str(Yeardata)
```

```
## 'data.frame':    5900385 obs. of  15 variables:
##  $ ride_id           : chr  "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8A
##  $ rideable_type     : chr  "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct, format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr  "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave &
##  $ start_station_id  : chr  "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr  "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbard
##  $ end_station_id    : chr  "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "member" "member" ...
##  $ weekday           : chr  "Friday" "Wednesday" "Sunday" "Thursday" ...
##  $ month             : chr  "July" "July" "July" "July" ...
```

3.6 Creating new column named as weekend_weekday based on weekday column

```
Yeardata$weekend_weekday=ifelse(Yeardata$weekday==c("Saturday","Sunday"),"weekend","weekday")
```

```
## Warning in Yeardata$weekday == c("Saturday", "Sunday"): longer object length is
## not a multiple of shorter object length
```

```
str(Yeardata)
```

```
## 'data.frame':    5900385 obs. of  16 variables:
##  $ ride_id           : chr  "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8A
##  $ rideable_type     : chr  "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct, format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr  "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave &
##  $ start_station_id  : chr  "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr  "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbard
##  $ end_station_id    : chr  "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "member" "member" ...
##  $ weekday           : chr  "Friday" "Wednesday" "Sunday" "Thursday" ...
##  $ month             : chr  "July" "July" "July" "July" ...
##  $ weekend_weekday   : chr  "weekday" "weekday" "weekday" "weekday" ...
```

3.7 Creating New column called duration_hr substracting from ended_at to started_at

```
Yeardata$duration_hr=round(difftime(Yeardata$ended_at,Yeardata$started_at,units="hours"),digits=2)
```

```
str(Yeardata)
```

```
## 'data.frame':    5900385 obs. of  17 variables:
##  $ ride_id           : chr  "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8A
##  $ rideable_type     : chr  "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct, format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr  "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave &
##  $ start_station_id  : chr  "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr  "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbard
##  $ end_station_id    : chr  "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "member" "member" ...
##  $ weekday           : chr  "Friday" "Wednesday" "Sunday" "Thursday" ...
##  $ month             : chr  "July" "July" "July" "July" ...
```

```
##  $ weekend_weekday   : chr  "weekday" "weekday" "weekday" "weekday" ...
##  $ duration_hr       : 'difftime' num  0.59 0.31 0.3 0.25 ...
##   ..- attr(*, "units")= chr "hours"
```

```
head(Yeardata)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 0A1B623926EF4E16   docked_bike 2021-07-02 14:44:36 2021-07-02 15:19:58
## 2 B2D5583A5A5E76EE  classic_bike 2021-07-07 16:57:42 2021-07-07 17:16:09
## 3 6F264597DDBF427A  classic_bike 2021-07-25 11:30:55 2021-07-25 11:48:45
## 4 379B58EAB20E8AA5  classic_bike 2021-07-08 22:08:30 2021-07-08 22:23:32
## 5 6615C1E4EB08E8FB electric_bike 2021-07-28 16:08:06 2021-07-28 16:27:09
## 6 62DC2B32872F9BA8 electric_bike 2021-07-29 17:09:08 2021-07-29 17:15:00
##             start_station_name start_station_id               end_station_name
## 1 Michigan Ave & Washington St            13001   Halsted St & North Branch St
## 2   California Ave & Cortez St            17660           Wood St & Hubbard St
## 3         Wabash Ave & 16th St           SL-012           Rush St & Hubbard St
## 4   California Ave & Cortez St            17660        Carpenter St & Huron St
## 5   California Ave & Cortez St            17660 Elizabeth (May) St & Fulton St
## 6   California Ave & Cortez St            17660  Albany Ave & Bloomingdale Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual   weekday
## 1   KA1504000117  41.88398 -87.62468 41.89937 -87.64848        casual    Friday
## 2          13432  41.90036 -87.69670 41.88990 -87.67147        casual Wednesday
## 3   KA1503000044  41.86038 -87.62581 41.89017 -87.62619        member    Sunday
## 4          13196  41.90036 -87.69670 41.89456 -87.65345        member  Thursday
## 5          13197  41.90035 -87.69668 41.88659 -87.65839        casual Wednesday
## 6          15655  41.90033 -87.69674 41.91389 -87.70513        casual  Thursday
##   month weekend_weekday duration_hr
## 1  July         weekday  0.59 hours
## 2  July         weekday  0.31 hours
## 3  July         weekday  0.30 hours
## 4  July         weekday  0.25 hours
## 5  July         weekday  0.32 hours
## 6  July         weekday  0.10 hours
```

Stage 4

Analyze

4.1 Analyse step is most important in our data analysis process. It is detective kind of task in data analysis journey. During analysis we will discover trend,pattern and relation in Our analyze step should move with considering our business task understand differencebetween casual rider and member rider.

```
install.packages("ggplot2")
```

Installing required packages.

```
## Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Admin\AppData\Local\Temp\Rtmp650wtF\downloaded_packages
```

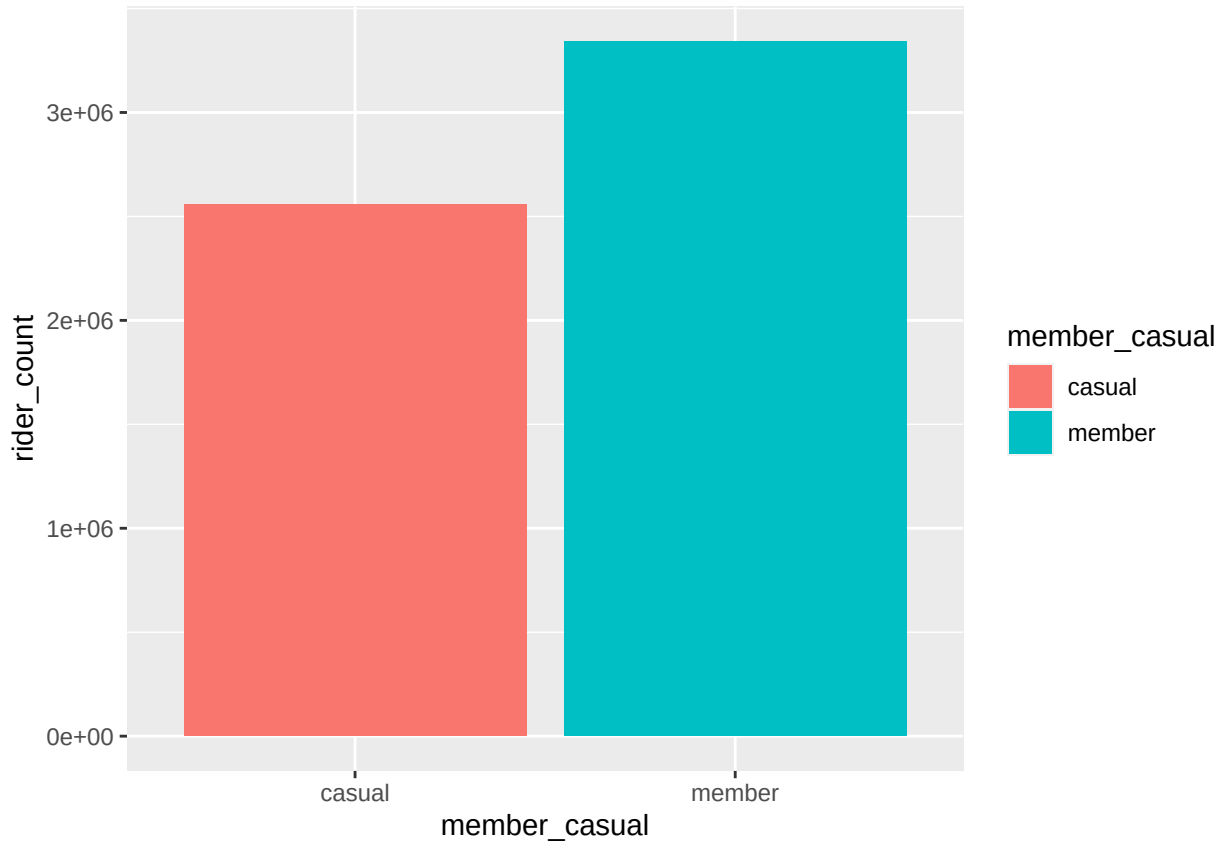
```
library(ggplot2)
```

```
Yeardata%>%
group_by(member_casual) %>%
summarise(rider_count = n())
```

Checking number of casual and member riders

```
## # A tibble: 2 x 2
##   member_casual rider_count
##   <chr>               <int>
## 1 casual            2558227
## 2 member            3342158
```

```
options(repr.plot.width = 5, repr.plot.height = 2.1)
Yeardata%>%
group_by(member_casual) %>%
summarise(rider_count = n())%>%
ggplot()+
geom_col(mapping = aes(x=member_casual,y=rider_count,fill=member_casual))
```



Total count of one year duration Member users are more than casual riders.

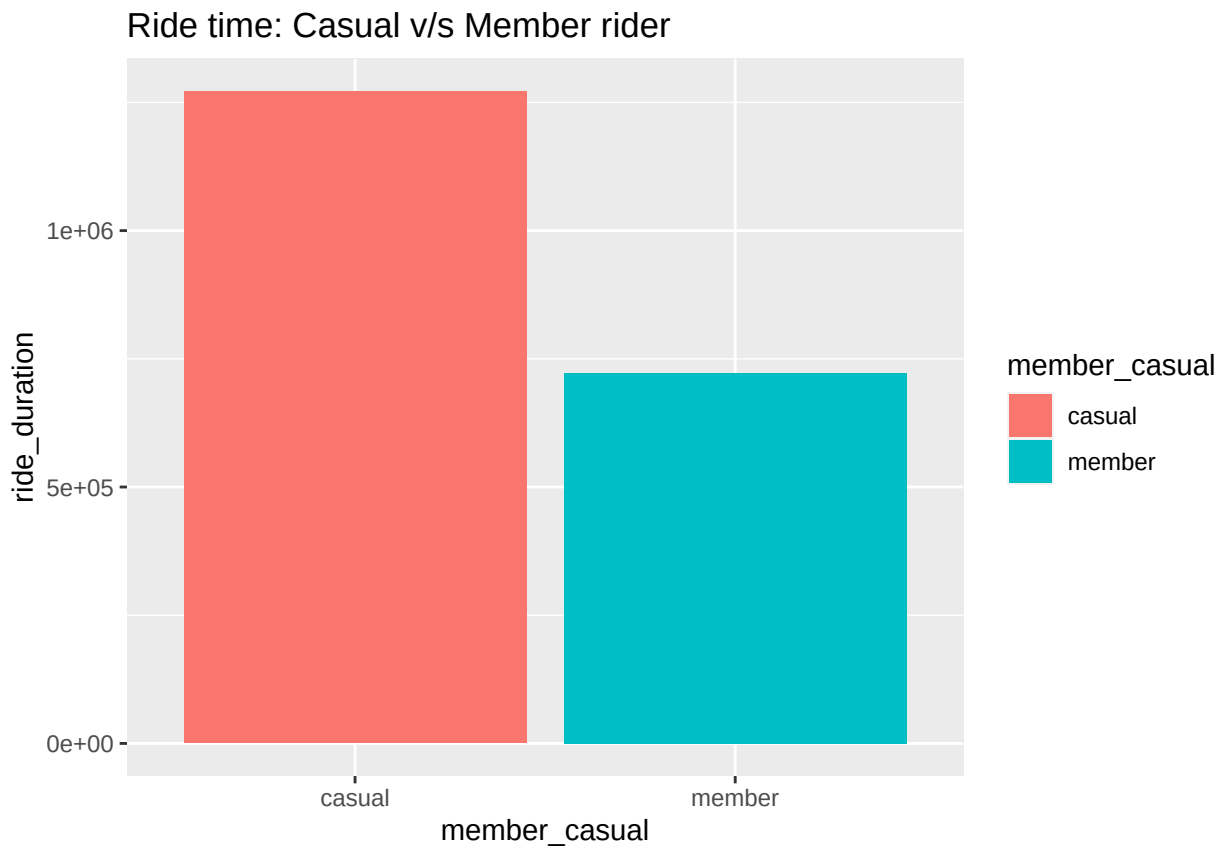
```
Yeardata %>%
group_by(member_casual) %>%
summarise(ride_duration=sum(duration_hr))
```

4.2 Comparing the total counts between casual and member riders.

```
## # A tibble: 2 x 2
##   member_casual ride_duration
##   <chr>         <drtn>
## 1 casual        1271364.0 hours
## 2 member         723103.5 hours
```

```
options(repr.plot.width = 5, repr.plot.height = 2.1)
Yeardata %>%
group_by(member_casual) %>%
summarise(ride_duration=sum(duration_hr))%>%
ggplot()+
geom_col(mapping =aes(x=member_casual,y=ride_duration,fill=member_casual))+
labs(title = "Ride time: Casual v/s Member rider" )
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



Casual riders are less than member rider, but as per the time duration casual riders rode more compare to member riders.

4.3 comparing average timing between Casual and Member riders

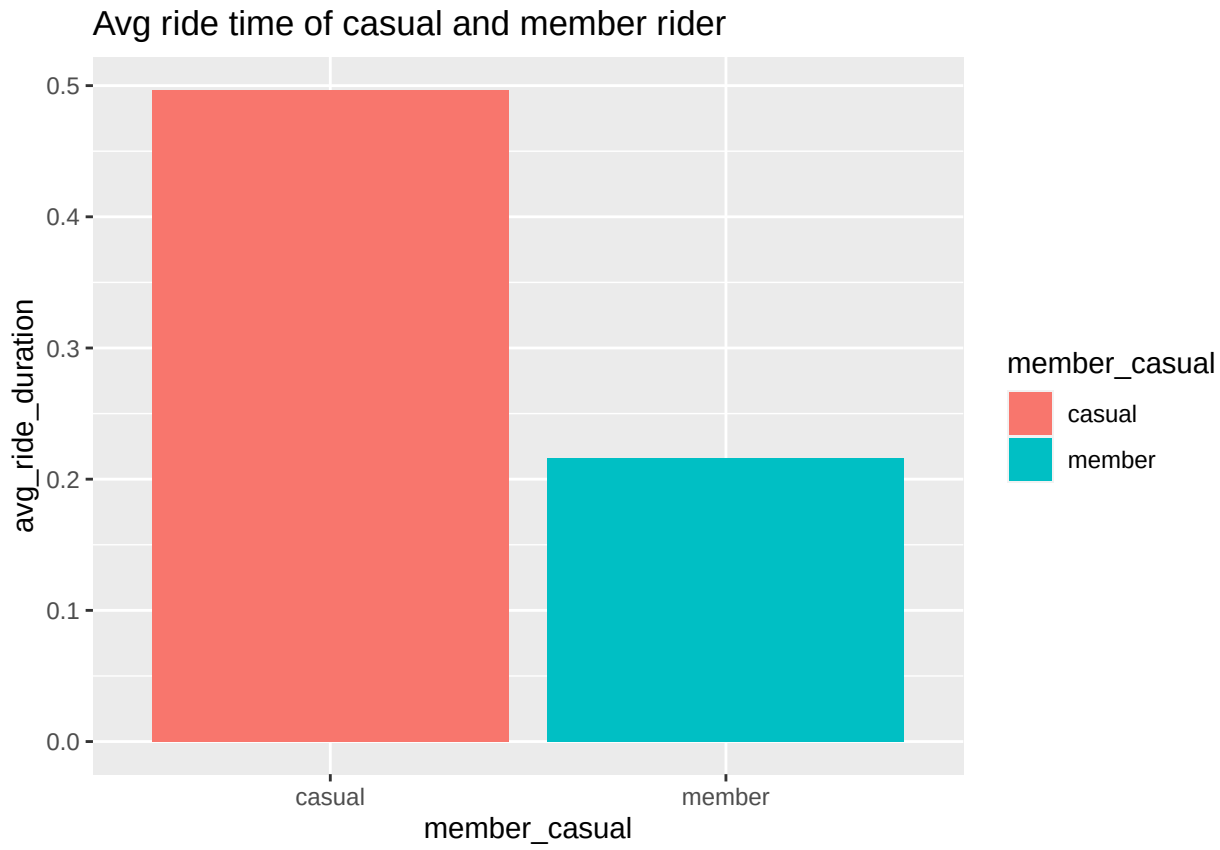
```
Yeardata %>%
group_by(member_casual) %>%
summarise(avg_ride_duration=mean(duration_hr))
```

```
## # A tibble: 2 x 2
##   member_casual avg_ride_duration
##   <chr>         <drtn>
## 1 casual        0.4969708 hours
## 2 member        0.2163583 hours
```

```
options(repr.plot.width = 5, repr.plot.height = 2.1)
Yeardata %>%
group_by(member_casual) %>%
summarise(avg_ride_duration=mean(duration_hr))%>%
ggplot()+
```

```
labs(title = "Avg ride time of casual and member rider" )+
geom_col(mapping =aes(x=member_casual,y=avg_ride_duration,fill=member_casual))
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

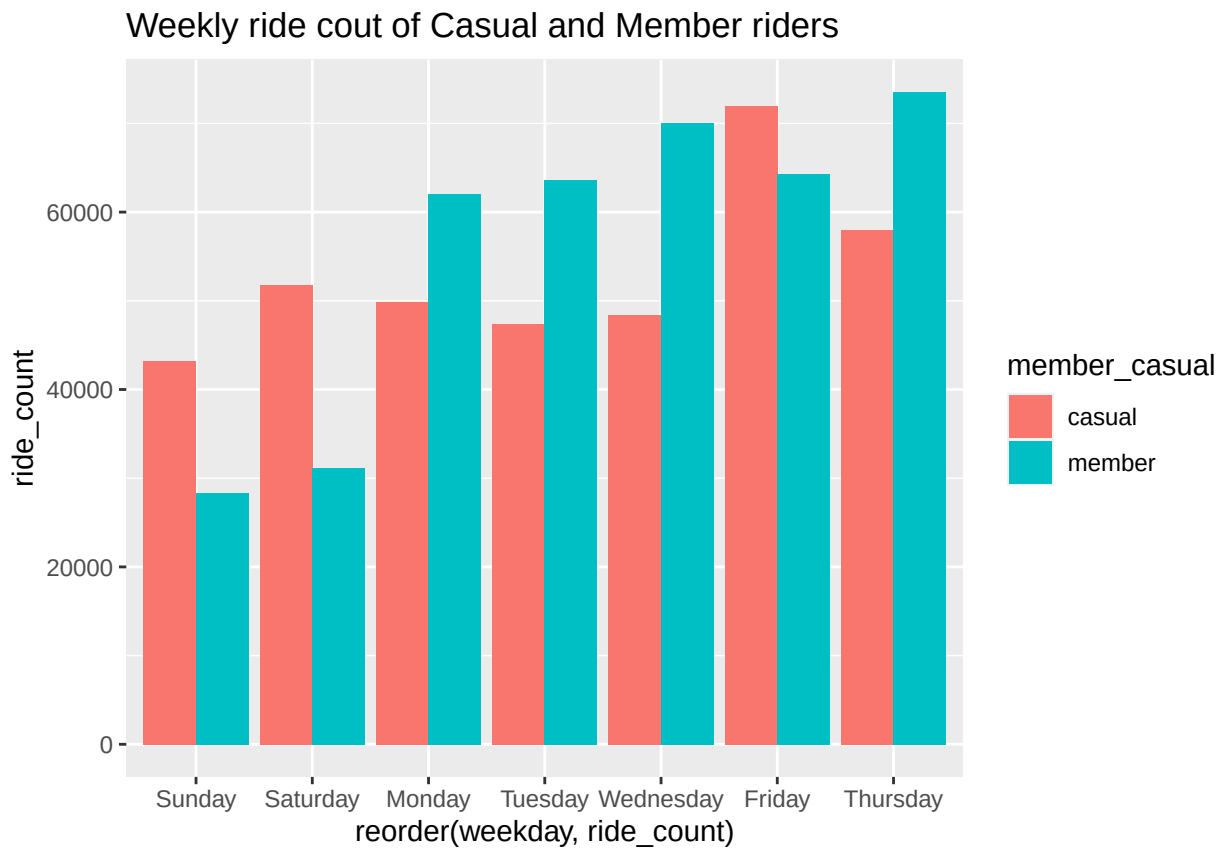


As per the average time callculation casual riders used more time for riding cycle.

4.4 ride count of casul and member riders during the week

```
Yeardata %>%
group_by(member_casual,weekday,weekend_weekday,month) %>% summarise(ride_count=n(),ride_duration=sum(du
ggplot(mapping =aes(x=reorder(weekday,ride_count),y=ride_count,fill=member_casual))+
labs(title = "Weekly ride cout of Casual and Member riders" )+
geom_col(position = "dodge")
```

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

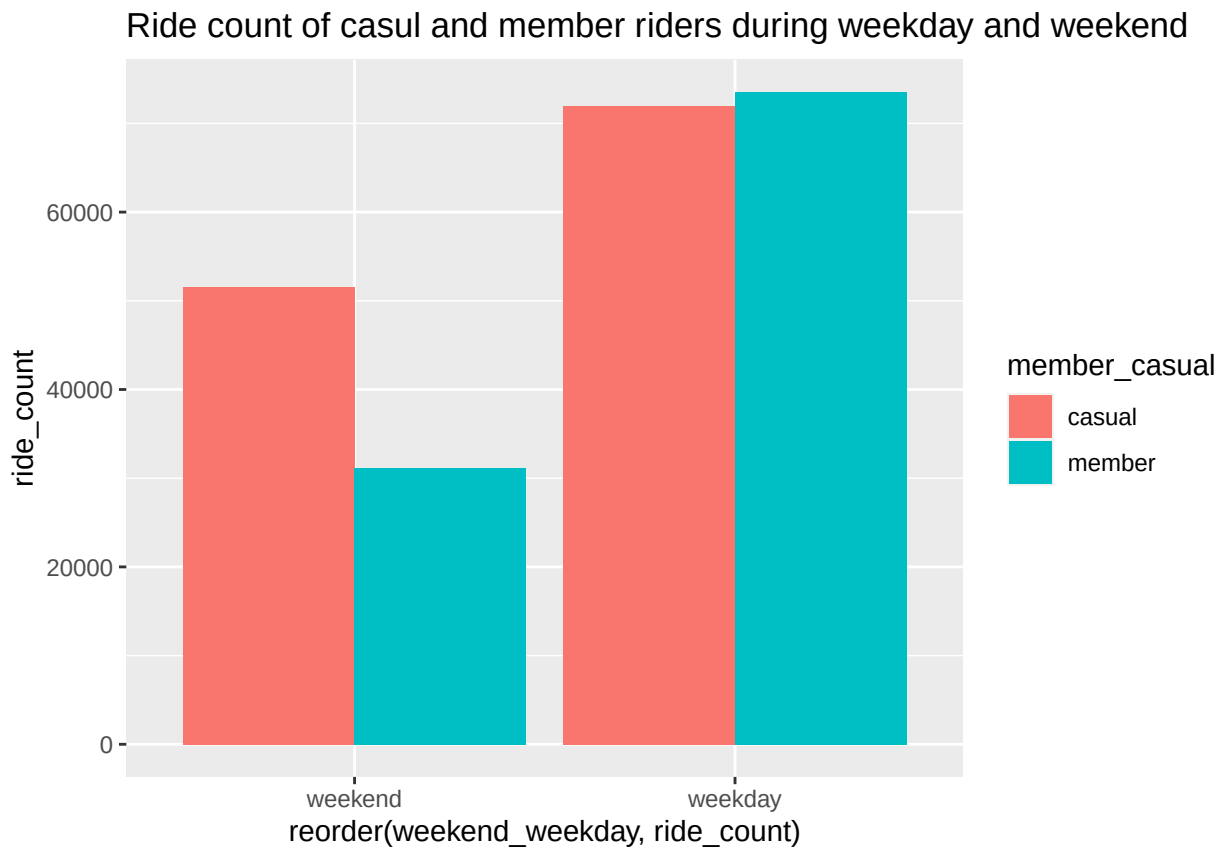


It look likes casual riders rode more on weekend compare to member riders although still finding weekend and weekdays variable.

4.5 Ride count of casul and member riders during weekday and weekend

```
Yeardata %>%
group_by(member_casual,weekday,weekend_weekday,month) %>%summarise(ride_count=n(),ride_duration=sum(dura
ggplot(mapping =aes(x=reorder(weekend_weekday,ride_count),y=ride_count,fill=member_casual))+
labs(title = "Ride count of casul and member riders during weekday and weekend " )+
geom_col(position = "dodge")
```

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

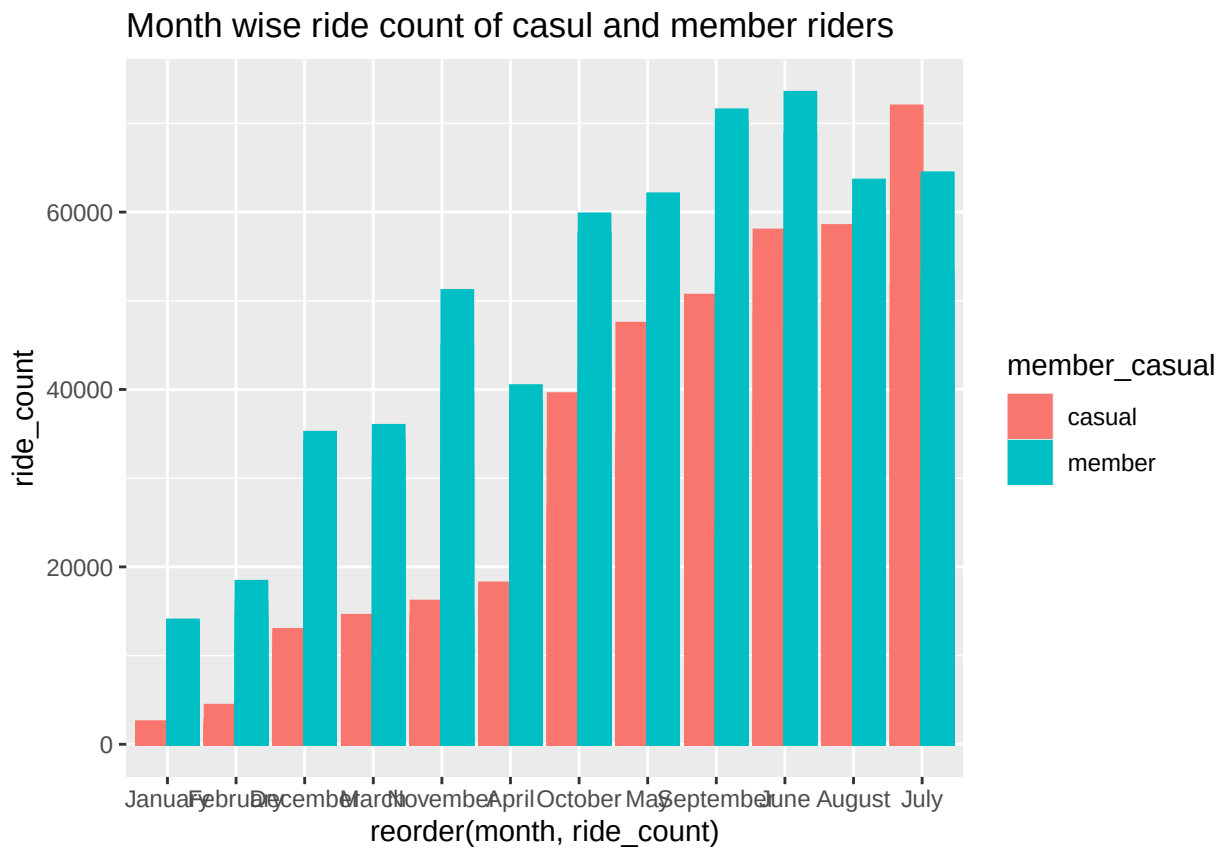


As per the above dashboard casual riders weekend count is more than member riders.

```
Yeardata %>%
group_by(member_casual,weekday,weekend_weekday,month) %>% summarise(ride_count=n(),ride_duration=sum(du
ggplot(mapping =aes(x=reorder(month,ride_count),y=ride_count,fill=member_casual,color=member_casual))+
labs(title = "Month wise ride count of casul and member riders " )+
geom_col(position = "dodge")
```

4.6 Month wise ride count of casual and member riders.

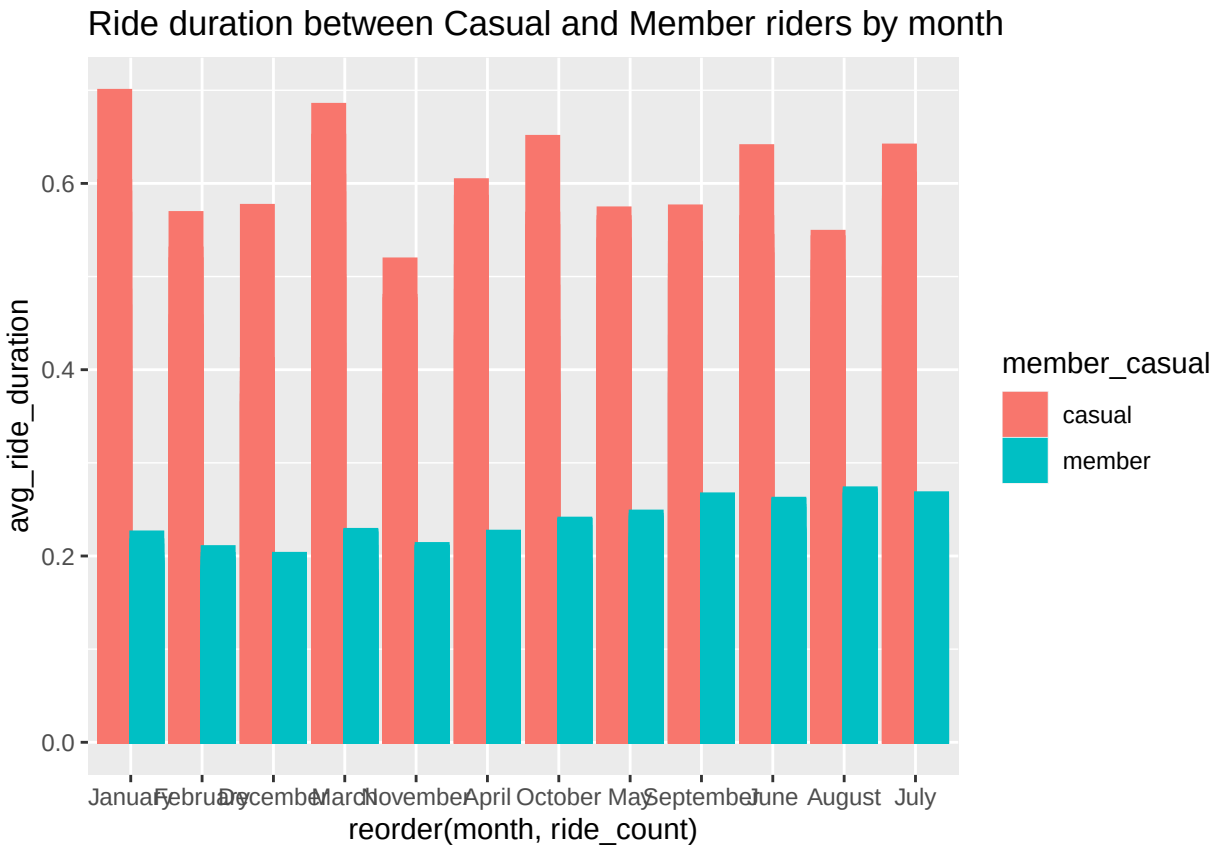
```
## `summarise()` has grouped output by 'member_casual', 'weekday',
## 'weekend_weekday'. You can override using the `.groups` argument.
```



Member riders rode more during the year expect july month. During the summer time(June, July,August, September) **Casual riders** ride the bike more so during this time is the better way to approch casual riders. #### 4.7 Ride duration between Casual and Member riders by month

```
Yeardata %>%
group_by(member_casual,weekday,weekend_weekday,month) %>%
summarise(ride_count=n(),ride_duration=sum(duration_hr),avg_ride_duration=mean(duration_hr))%>%
ggplot(mapping =aes(x=reorder(month,ride_count),y=avg_ride_duration,fill=member_casual,color=member_cas
labs(title = "Ride duration between Casual and Member riders by month" )+
geom_col(position = "dodge")
```

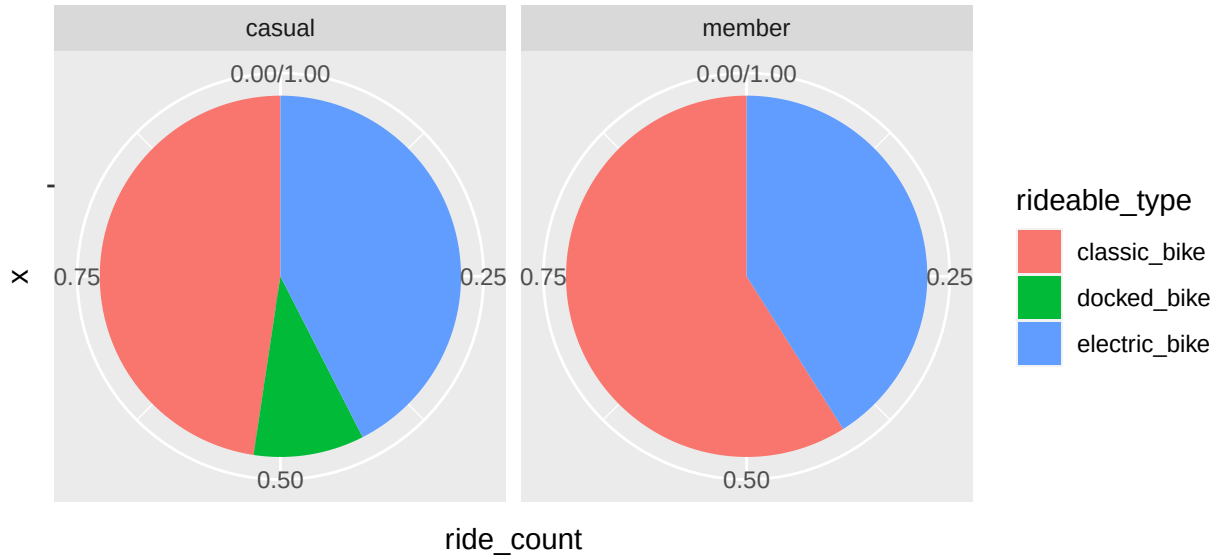
```
## `summarise()` has grouped output by 'member_casual', 'weekday',
## 'weekend_weekday'. You can override using the `.groups` argument.
## Don't know how to automatically pick scale for object of type difftime.
## Defaulting to continuous.
```




```
Yeardata %>%
group_by(member_casual,rideable_type)%>%
summarise(ride_count=n())%>%
ggplot(aes(x="",y=ride_count,fill=rideable_type))+
geom_bar(stat = "identity",width = 2,position = "fill")+
coord_polar(theta = "y")+
facet_wrap(~member_casual)
```

Casual riders ride time is higher than member riders in every month launching any strategy to support this behaviour of casual ridrer can help them retain.

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



Here is important difference between casul and member rider casual rider use docked_bike but no member rider use this

Key findings

During the analysis we found below differences between casual and member riders

1.Cyclistic has more member riders than casual riders

2.Casual riders ride duration is higher than member riders

3. During weekend(friday,saturday,sunday) casual riders are more active than member riders

4. Member riders ride more than Casual riders during the year expect july month for any promotional activity towards casual riders from June to September is better period

5. Casual rider use docked_bike but no member riders use this

Act

Act is stage of of data analytics with the help of insights we will recommend next steps

Below are the recommendation of cyclistic company from insights

1. As per data clearly says casual riders use bikes during weekend more than member riders , company can use this point to provide them yearly pass for weekend rides.

2. As casual riders ride bike for longer duration company can come up with different strategy to support this pattern.