case_study

March 19, 2022

1 Case study

2 How Does cyclistic Bike-Share Navigate Speedy Success?

2.1 Stage 1

2.2 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

2.3 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, Moreno believes there is a very good chance to convert casual riders into members.

Breaking down business task into problum statement for further understading

```
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?
```

2.3.1 Problem assigned us to solve

How do annual members and casual riders use Cyclistic bikes differently?

2.4 Stage 2

2.5 Prepare data for exploration

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

Key task of prepare stage

1.Download data and store it appropriately.

2.Identify how its organized.

3.Short and filter data

4.Determine the credibility of data

2.5.1 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>

Data has been stored properly on respective path
C:\Users\sachi\OneDrive\Desktop\Case Study_Bike_Share\Bike_Share_12_month_data\Original_data_

2.5.2 Identify how data is orgenized need to import data sets to R studio

Installing tidyverse pakage which is esensial for data analysis in R

[2]:
```
install.packages("tidyverse")
```

```
package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages
```

[ ]:
```
library(tidyverse)
```

2.5.3 Importing cvs files

[4]:
```
df1=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-04-divvy-tripdata.csv")

df2=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-05-divvy-tripdata.csv")

df3=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-06-divvy-tripdata.csv")

df4=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-07-divvy-tripdata.csv")

df5=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-08-divvy-tripdata.csv")
```

```
df6=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-09-divvy-tripdata.csv")

df7=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-10-divvy-tripdata.csv")

df8=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-11-divvy-tripdata.csv")

df9=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-12-divvy-tripdata.csv")

df10=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2022-01-divvy-tripdata.csv")

df11=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2022-02-divvy-tripdata.csv")

df12=read.csv("C:/Users/sachi/OneDrive/Desktop/Case Study_ Bike_Share/
 ↪Bike_Share_12_month_data/Original_data_files/2021-03-divvy-tripdata.csv")
```

[322]:
```
head(df1)
```

ride_id	rideable_type	started_at	ended_at	start_station_name
6C992BD37A98A63F	classic_bike	2021-04-12 18:25:36	2021-04-12 18:56:55	State St & Pearson St
1E0145613A209000	docked_bike	2021-04-27 17:27:11	2021-04-27 18:31:29	Dorchester Ave & 49th St
E498E15508A80BAD	docked_bike	2021-04-03 12:42:45	2021-04-07 11:40:24	Loomis Blvd & 84th St
1887262AD101C604	classic_bike	2021-04-17 09:17:42	2021-04-17 09:42:48	Honore St & Division St
C123548CAB2A32A5	docked_bike	2021-04-03 12:42:25	2021-04-03 14:13:42	Loomis Blvd & 84th St
097E76F3651B1AC1	classic_bike	2021-04-25 18:43:18	2021-04-25 18:43:59	Clinton St & Polk St

2.5.4 Understanding dataset

[6]:
```
colnames(df1)
```

1. 'ride_id' 2. 'rideable_type' 3. 'started_at' 4. 'ended_at' 5. 'start_station_name' 6. 'start_station_id' 7. 'end_station_name' 8. 'end_station_id' 9. 'start_lat' 10. 'start_lng' 11. 'end_lat' 12. 'end_lng' 13. 'member_casual'

[7]:
```
colnames(df11)
```

1. 'ride_id' 2. 'rideable_type' 3. 'started_at' 4. 'ended_at' 5. 'start_station_name' 6. 'start_station_id' 7. 'end_station_name' 8. 'end_station_id' 9. 'start_lat' 10. 'start_lng' 11. 'end_lat' 12. 'end_lng' 13. 'member_casual'

2.5.5 Comparing column name of datasets to combine all datasets

isntalling required package

[8]:
```
install.packages("janitor")
```

```
package 'janitor' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages
```

[9]:
```
library(janitor)
```

```
Warning message:
"package 'janitor' was built under R version 3.6.3"
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

2.5.6 Comparing dataset columns

[10]:
```
compare_df_cols(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
```

column_name	df1	df2	df3	df4	df5	df6	df7	df8	df9
end_lat	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric	nume
end_lng	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric	nume
end_station_id	factor	factor	factor	factor	factor	factor	factor	factor	factor
end_station_name	factor	factor	factor	factor	factor	factor	factor	factor	factor
ended_at	factor	factor	factor	factor	factor	factor	factor	factor	factor
member_casual	factor	factor	factor	factor	factor	factor	factor	factor	factor
ride_id	factor	factor	factor	factor	factor	factor	factor	factor	factor
rideable_type	factor	factor	factor	factor	factor	factor	factor	factor	factor
start_lat	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric	nume
start_lng	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric	nume
start_station_id	factor	factor	factor	factor	factor	factor	factor	factor	factor
start_station_name	factor	factor	factor	factor	factor	factor	factor	factor	factor
started_at	factor	factor	factor	factor	factor	factor	factor	factor	factor

Each dataset has similiral colums and similar data type so its appropriate to combine all ataset

2.5.7 Combining datasests

[11]:
```
union_df=rbind(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
```

[12]:
```
head(union_df)
```

ride_id	rideable_type	started_at	ended_at	start_station_name
6C992BD37A98A63F	classic_bike	2021-04-12 18:25:36	2021-04-12 18:56:55	State St & Pearson St
1E0145613A209000	docked_bike	2021-04-27 17:27:11	2021-04-27 18:31:29	Dorchester Ave & 49th St
E498E15508A80BAD	docked_bike	2021-04-03 12:42:45	2021-04-07 11:40:24	Loomis Blvd & 84th St
1887262AD101C604	classic_bike	2021-04-17 09:17:42	2021-04-17 09:42:48	Honore St & Division St
C123548CAB2A32A5	docked_bike	2021-04-03 12:42:25	2021-04-03 14:13:42	Loomis Blvd & 84th St
097E76F3651B1AC1	classic_bike	2021-04-25 18:43:18	2021-04-25 18:43:59	Clinton St & Polk St

[13]:
```
tail(union_df)
```

	ride_id	rideable_type	started_at	ended_at	start_station_
5667981	081549DEA616CA22	electric_bike	2021-03-14 01:59:38	2021-03-14 03:13:09	Larrabee St &
5667982	9397BDD14798A1BA	docked_bike	2021-03-20 14:58:56	2021-03-20 17:22:47	Michigan Ave
5667983	BBBEB8D51AAD40DA	classic_bike	2021-03-02 11:35:10	2021-03-02 11:43:37	Kingsbury St
5667984	637FF754DA0BD9E1	classic_bike	2021-03-09 11:07:36	2021-03-09 11:49:11	Michigan Ave
5667985	F8F43A0B978A7A35	classic_bike	2021-03-01 18:11:57	2021-03-01 18:18:37	Kingsbury St
5667986	3AE64EA5BF43CF72	electric_bike	2021-03-26 17:58:14	2021-03-26 18:06:43	

2.5.8 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

2.6 Stage 3

2.7 Process

Process stage is very important in data analytics because here data will get cleaned and transpormed for analysis stage clean and transpormed data is key for accurate analysis

2.7.1 Key tasks of process stage

```
1. Check for errors in data

2.Check for duplicate data

3.Treat null values

4.Orgenize and format data

5.perform calculations

6.Derived metrics or new metrics
```

2.7.2 Installing packages for data cleaning and data overview

[14]:
```
install.packages("readr")
install.packages("dplyr")
library(readr)
library(dplyr)
```

```
also installing the dependencies 'glue', 'cli', 'vroom'


  There are binary versions available but the source versions are later:
      binary source needs_compilation
glue   1.4.2  1.6.2              TRUE
cli    2.5.0  3.2.0              TRUE
vroom  1.4.0  1.5.7              TRUE
readr  1.4.0  2.1.2              TRUE

  Binaries will be installed

Warning message:
"package 'readr' is in use and will not be installed"

package 'glue' successfully unpacked and MD5 sums checked

Warning message:
"cannot remove prior installation of package 'glue'"Warning message in
file.copy(savedcopy, lib, recursive = TRUE):
"problem copying
C:\Users\sachi\anaconda3\envs\r\Lib\R\library\00LOCK\glue\libs\x64\glue.dll to
C:\Users\sachi\anaconda3\envs\r\Lib\R\library\glue\libs\x64\glue.dll: Permission
denied"Warning message:
"restored 'glue'"

package 'cli' successfully unpacked and MD5 sums checked
package 'vroom' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages

also installing the dependency 'rlang'


  There are binary versions available but the source versions are later:
      binary source needs_compilation
rlang 0.4.11  1.0.2              TRUE
dplyr  1.0.6  1.0.8              TRUE

  Binaries will be installed
```

```
Warning message:
"package 'dplyr' is in use and will not be installed"
```

```
package 'rlang' successfully unpacked and MD5 sums checked
```

```
Warning message:
"cannot remove prior installation of package 'rlang'"Warning message in
file.copy(savedcopy, lib, recursive = TRUE):
"problem copying
C:\Users\sachi\anaconda3\envs\r\Lib\R\library\00LOCK\rlang\libs\x64\rlang.dll to
C:\Users\sachi\anaconda3\envs\r\Lib\R\library\rlang\libs\x64\rlang.dll:
Permission denied"Warning message:
"restored 'rlang'"


The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages
```

[17]:
```
install.packages("skimr")
install.packages("here")
library(skimr)
library(here)
```

```
package 'skimr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages
package 'here' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sachi\AppData\Local\Temp\Rtmp6xJb7e\downloaded_packages
Warning message:
"package 'skimr' was built under R version 3.6.3"Warning message:
"package 'here' was built under R version 3.6.3"here() starts at C:/Users/sachi
```

[18]:
```
library(lubridate)
```

```
Warning message:
"package 'lubridate' was built under R version 3.6.3"
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

Checking for error , duplicate and treat null values

[19]:
```
skim_without_charts(union_df)
```

```
Warning message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."

-- Data Summary ------------------------
                           Values
Name                       union_df
Number of rows             5667986
Number of columns          13
-----------------------
Column type frequency:
  factor                   9
  numeric                  4
-----------------------
Group variables            None

-- Variable type: factor -------------------------------------------------------
# A tibble: 9 x 6
  skim_variable      n_missing complete_rate ordered n_unique
* <chr>                  <int>         <dbl> <lgl>      <int>
1 ride_id                    0             1 FALSE    5667986
2 rideable_type              0             1 FALSE          3
3 started_at                 0             1 FALSE    4747127
4 ended_at                   0             1 FALSE    4740417
5 start_station_name         0             1 FALSE        854
6 start_station_id           0             1 FALSE        845
7 end_station_name           0             1 FALSE        855
8 end_station_id             0             1 FALSE        847
9 member_casual              0             1 FALSE          2
  top_counts
* <chr>
1 000: 1, 000: 1, 000: 1, 000: 1
2 cla: 3268797, ele: 2087901, doc: 311288
3 202: 7, 202: 7, 202: 7, 202: 7
4 202: 17, 202: 16, 202: 15, 202: 14
5 emp: 712978, Str: 82954, Mic: 44409, Wel: 43969
6 emp: 712975, 130: 82954, LF-: 47856, 133: 46176
7 emp: 761817, Str: 83648, Mic: 44913, Wel: 44149
8 emp: 761817, 130: 83648, LF-: 53932, 130: 44913
9 mem: 3127293, cas: 2540693

-- Variable type: numeric ------------------------------------------------------
# A tibble: 4 x 10
```

```
  skim_variable n_missing complete_rate  mean     sd    p0   p25   p50   p75
* <chr>              <int>         <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 start_lat              0         1      41.9 0.0463  41.6  41.9  41.9  41.9
2 start_lng              0         1     -87.6 0.0295 -87.8 -87.7 -87.6 -87.6
3 end_lat             4617         0.999  41.9 0.0463  41.4  41.9  41.9  41.9
4 end_lng             4617         0.999 -87.6 0.0291 -89.0 -87.7 -87.6 -87.6
   p100
* <dbl>
1  45.6
2 -73.8
3  42.2
4 -87.5
```

[20]:
```
union_df%>%
    distinct(.keep_all = TRUE) %>%
    skim_without_charts()
```

```
Warning message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."Warning
message in sorted_count(x):
"Variable contains value(s) of "" that have been converted to "empty"."

-- Data Summary ------------------------
                           Values
Name                       Piped data
Number of rows             5667986
Number of columns          13

-----------------------
Column type frequency:
  factor                   9
  numeric                  4

------------------------
Group variables            None

-- Variable type: factor -------------------------------------------------------
# A tibble: 9 x 6
  skim_variable      n_missing complete_rate ordered n_unique
* <chr>                  <int>         <dbl> <lgl>      <int>
1 ride_id                    0             1 FALSE    5667986
2 rideable_type              0             1 FALSE          3
3 started_at                 0             1 FALSE    4747127
4 ended_at                   0             1 FALSE    4740417
5 start_station_name         0             1 FALSE        854
6 start_station_id           0             1 FALSE        845
```

```
7 end_station_name           0             1 FALSE        855
8 end_station_id             0             1 FALSE        847
9 member_casual              0             1 FALSE          2
  top_counts
* <chr>
1 000: 1, 000: 1, 000: 1, 000: 1
2 cla: 3268797, ele: 2087901, doc: 311288
3 202: 7, 202: 7, 202: 7, 202: 7
4 202: 17, 202: 16, 202: 15, 202: 14
5 emp: 712978, Str: 82954, Mic: 44409, Wel: 43969
6 emp: 712975, 130: 82954, LF-: 47856, 133: 46176
7 emp: 761817, Str: 83648, Mic: 44913, Wel: 44149
8 emp: 761817, 130: 83648, LF-: 53932, 130: 44913
9 mem: 3127293, cas: 2540693

-- Variable type: numeric -------------------------------------------------------
# A tibble: 4 x 10
  skim_variable n_missing complete_rate  mean     sd    p0   p25   p50   p75
* <chr>             <int>         <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 start_lat             0         1      41.9 0.0463  41.6  41.9  41.9  41.9
2 start_lng             0         1     -87.6 0.0295 -87.8 -87.7 -87.6 -87.6
3 end_lat            4617         0.999  41.9 0.0463  41.4  41.9  41.9  41.9
4 end_lng            4617         0.999 -87.6 0.0291 -89.0 -87.7 -87.6 -87.6
   p100
* <dbl>
1  45.6
2 -73.8
3  42.2
4 -87.5
```

When we execute both skim_without_charts(union_df) and union_df %>% distinct(.keep_all = TRUE) %>% kim_without_charts() rows count remain same 5667986 so we finaliesed there is no duplicate rows in dataset and missing values in end_lat and end_lang kept on.

2.7.3 New metrics creation and calculations

Dealing with dates

```
we have already loaded lubridate library to deal with dates
```

Assigning datafrme to new varible to prevent it from crash during transformation

[21]:
```
new_df=union_df
```

Checking data type of date column

[22]:
```
class(new_df$started_at)
class(new_df$ended_at)
```

'factor'

'factor'

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

[23]:
```
new_df$started_at=ymd_hms(new_df$started_at)
new_df$ended_at=ymd_hms(new_df$ended_at)
```

[24]:
```
class(new_df$started_at)
class(new_df$ended_at)
```

1. 'POSIXct' 2. 'POSIXt'

1. 'POSIXct' 2. 'POSIXt'

Now its ready to extract date and time related information from this columns

Extraction of day name into new coumns weekday as per our bussiness task we might require weekday from started_at column

[25]:
```
new_df$weekday=weekdays(new_df$started_at)
```

Month name extaction month column

[26]:
```
new_df$month=months(new_df$started_at)
```

[27]:
```
month=months(new_df$started_at)
```

[28]:
```
str(new_df)

# weekday and month has been added to dataframe
```

```
'data.frame':   5667986 obs. of  15 variables:
 $ ride_id           : Factor w/ 5667986 levels "00001A81D056B01B",..: 143082
39569 300958 32373 254792 12428 110411 280871 145203 5400 …
 $ rideable_type     : Factor w/ 3 levels "classic_bike",..: 1 2 2 1 2 1 1 3 1 1
…
 $ started_at        : POSIXct, format: "2021-04-12 18:25:36" "2021-04-27
17:27:11" …
 $ ended_at          : POSIXct, format: "2021-04-12 18:56:55" "2021-04-27
18:31:29" …
 $ start_station_name: Factor w/ 854 levels "","2112 W Peterson Ave",..: 579 221
402 309 402 160 19 221 19 221 …
 $ start_station_id  : Factor w/ 845 levels "","13001","13006",..: 572 438 242
541 242 148 210 438 210 438 …
 $ end_station_name  : Factor w/ 855 levels "","2112 W Peterson Ave",..: 559 219
400 559 400 158 18 219 18 219 …
 $ end_station_id    : Factor w/ 847 levels "","13001","13006",..: 81 438 242 81
242 148 210 438 210 438 …
```

```
 $ start_lat          : num  41.9 41.8 41.7 41.9 41.7 …
 $ start_lng          : num  -87.6 -87.6 -87.7 -87.7 -87.7 …
 $ end_lat            : num  41.9 41.8 41.7 41.9 41.7 …
 $ end_lng            : num  -87.7 -87.6 -87.7 -87.7 -87.7 …
 $ member_casual      : Factor w/ 2 levels "casual","member": 2 1 1 2 1 1 1 1 1 1
…
 $ weekday            : chr  "Monday" "Tuesday" "Saturday" "Saturday" …
 $ month              : chr  "April" "April" "April" "April" …
```

Creating new colnmm named weekend_weekday based on weekday column

[29]:
```
new_df$weekend_weekday=ifelse(new_df$weekday==c("Saturday","Sunday"),"weekend","weekday")
```

[30]:
```
str(new_df)
```

```
'data.frame':   5667986 obs. of  16 variables:
 $ ride_id           : Factor w/ 5667986 levels "00001A81D056B01B",..: 143082
39569 300958 32373 254792 12428 110411 280871 145203 5400 …
 $ rideable_type     : Factor w/ 3 levels "classic_bike",..: 1 2 2 1 2 1 1 3 1 1
…
 $ started_at        : POSIXct, format: "2021-04-12 18:25:36" "2021-04-27
17:27:11" …
 $ ended_at          : POSIXct, format: "2021-04-12 18:56:55" "2021-04-27
18:31:29" …
 $ start_station_name: Factor w/ 854 levels "","2112 W Peterson Ave",..: 579 221
402 309 402 160 19 221 19 221 …
 $ start_station_id  : Factor w/ 845 levels "","13001","13006",..: 572 438 242
541 242 148 210 438 210 438 …
 $ end_station_name  : Factor w/ 855 levels "","2112 W Peterson Ave",..: 559 219
400 559 400 158 18 219 18 219 …
 $ end_station_id    : Factor w/ 847 levels "","13001","13006",..: 81 438 242 81
242 148 210 438 210 438 …
 $ start_lat         : num  41.9 41.8 41.7 41.9 41.7 …
 $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.7 …
 $ end_lat           : num  41.9 41.8 41.7 41.9 41.7 …
 $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.7 …
 $ member_casual     : Factor w/ 2 levels "casual","member": 2 1 1 2 1 1 1 1 1 1
…
 $ weekday           : chr  "Monday" "Tuesday" "Saturday" "Saturday" …
 $ month             : chr  "April" "April" "April" "April" …
 $ weekend_weekday   : chr  "weekday" "weekday" "weekend" "weekday" …
```

Creating New column called duration_hr substracting from ended_at from started_at

[31]:
```
new_df$duration_hr=round(difftime(new_df$ended_at,new_df$started_at,units="hours"),digits␣
↪= 2)
```

[32]:
```
str(new_df)
```

```
'data.frame':   5667986 obs. of  17 variables:
 $ ride_id           : Factor w/ 5667986 levels "00001A81D056B01B",..: 143082
39569 300958 32373 254792 12428 110411 280871 145203 5400 …
 $ rideable_type     : Factor w/ 3 levels "classic_bike",..: 1 2 2 1 2 1 1 3 1 1
…
 $ started_at        : POSIXct, format: "2021-04-12 18:25:36" "2021-04-27
17:27:11" …
 $ ended_at          : POSIXct, format: "2021-04-12 18:56:55" "2021-04-27
18:31:29" …
 $ start_station_name: Factor w/ 854 levels "","2112 W Peterson Ave",..: 579 221
402 309 402 160 19 221 19 221 …
 $ start_station_id  : Factor w/ 845 levels "","13001","13006",..: 572 438 242
541 242 148 210 438 210 438 …
 $ end_station_name  : Factor w/ 855 levels "","2112 W Peterson Ave",..: 559 219
400 559 400 158 18 219 18 219 …
 $ end_station_id    : Factor w/ 847 levels "","13001","13006",..: 81 438 242 81
242 148 210 438 210 438 …
 $ start_lat         : num  41.9 41.8 41.7 41.9 41.7 …
 $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.7 …
 $ end_lat           : num  41.9 41.8 41.7 41.9 41.7 …
 $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.7 …
 $ member_casual     : Factor w/ 2 levels "casual","member": 2 1 1 2 1 1 1 1 1 1
…
 $ weekday           : chr  "Monday" "Tuesday" "Saturday" "Saturday" …
 $ month             : chr  "April" "April" "April" "April" …
 $ weekend_weekday   : chr  "weekday" "weekday" "weekend" "weekday" …
 $ duration_hr       : 'difftime' num  0.52 1.07 94.96 0.42 …
  ..- attr(*, "units")= chr "hours"
```

[33]:
```
head(new_df)
```

ride_id	rideable_type	started_at	ended_at	start_station_name
6C992BD37A98A63F	classic_bike	2021-04-12 18:25:36	2021-04-12 18:56:55	State St & Pearson St
1E0145613A209000	docked_bike	2021-04-27 17:27:11	2021-04-27 18:31:29	Dorchester Ave & 49th St
E498E15508A80BAD	docked_bike	2021-04-03 12:42:45	2021-04-07 11:40:24	Loomis Blvd & 84th St
1887262AD101C604	classic_bike	2021-04-17 09:17:42	2021-04-17 09:42:48	Honore St & Division St
C123548CAB2A32A5	docked_bike	2021-04-03 12:42:25	2021-04-03 14:13:42	Loomis Blvd & 84th St
097E76F3651B1AC1	classic_bike	2021-04-25 18:43:18	2021-04-25 18:43:59	Clinton St & Polk St

2.8 Stage 4

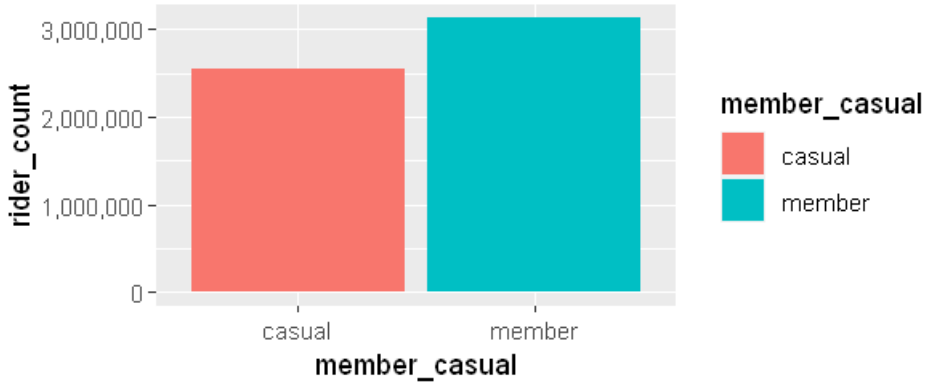
2.9 Analyze

Analyse is detective kind of task in data analysis journey. During analysis we will discover trend,pattern and relation in dataset

Our analyze step should move with considering our business task understand difference between casual rider and member rider

[200]:
```
# number of casul and menber riders
new_df %>%
      group_by(member_casual) %>%
      summarise(rider_count = n())
options(repr.plot.width = 5, repr.plot.height = 2.1)
new_df %>%
      group_by(member_casual) %>%
      summarise(rider_count = n())%>%
      ggplot()+
      geom_col(mapping = aes(x=member_casual,y=rider_count,fill=member_casual))+
      scale_y_continuous(labels = comma)
```

member_casual	rider_count
casual	2540693
member	3127293



Cyclistic has more member riders than casual riders

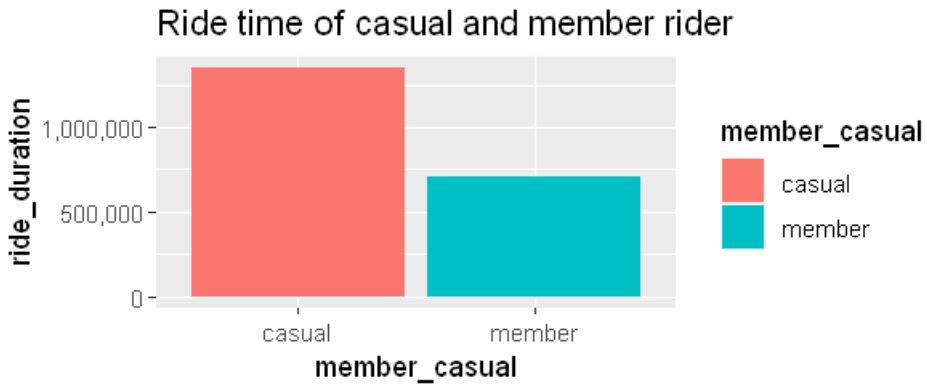
[323]:
```
#ride time of casual and member rider

new_df %>%
      group_by(member_casual) %>%
      summarise(ride_duration=sum(duration_hr))

options(repr.plot.width = 5, repr.plot.height = 2.1)
new_df %>%
      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 of casual and member rider" )+
```

```
      scale_y_continuous(labels = comma)
```

member_casual	ride_duration
casual	1351640.7 hours
member	702795.9 hours



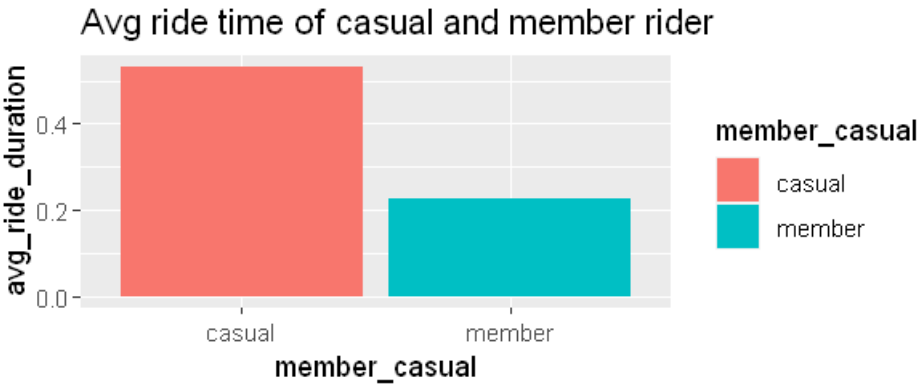
Cyclistic has less casual members but their riding duration is more than member riders

[237]:
```
### avg ride time of casual and member rider
new_df %>%
      group_by(member_casual) %>%
      summarise(avg_ride_duration=mean(duration_hr))

options(repr.plot.width = 5, repr.plot.height = 2.1)
new_df %>%
      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))
```

member_casual	avg_ride_duration
casual	0.5319969 hours
member	0.2247298 hours

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

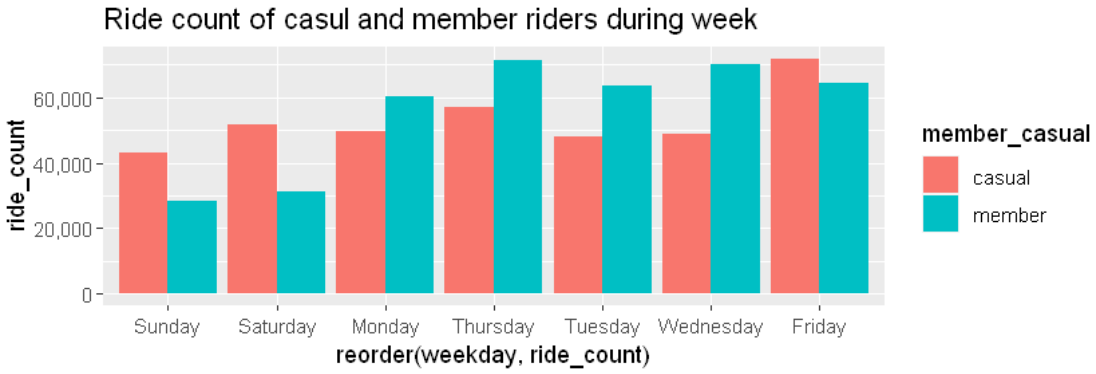


casual riders are AVG ride time winners

[229]:
```
### ride count of casul and member riders during week

options(repr.plot.width = 7, repr.plot.height = 2.4)
new_df %>%
      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(weekday,ride_count),y=ride_count,fill=member_casual))+
      labs(title = "Ride count of casul and member riders during week" )+
      geom_col(position = "dodge") +
      scale_y_continuous(labels = comma)
```

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

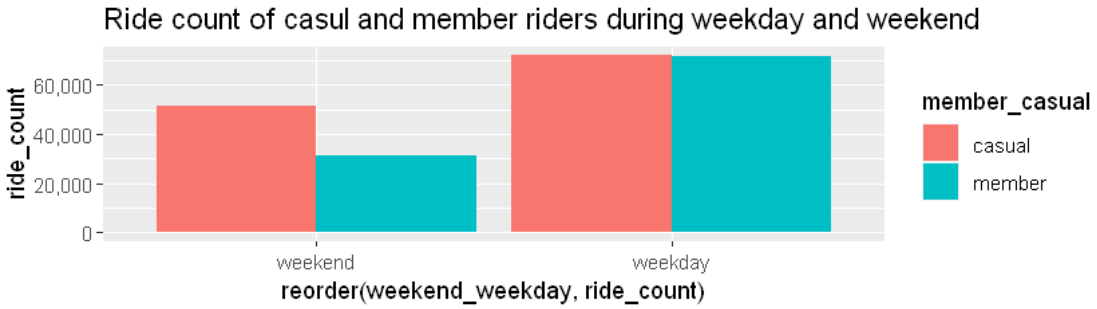


It looks casul riders weekend riders still we can confirm this by weekday_weekend variable

[266]:
```
### ride count of casul and member riders during weekday and weekend

options(repr.plot.width = 7, repr.plot.height = 2)
new_df %>%
      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(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") +
      scale_y_continuous(labels = comma)
```

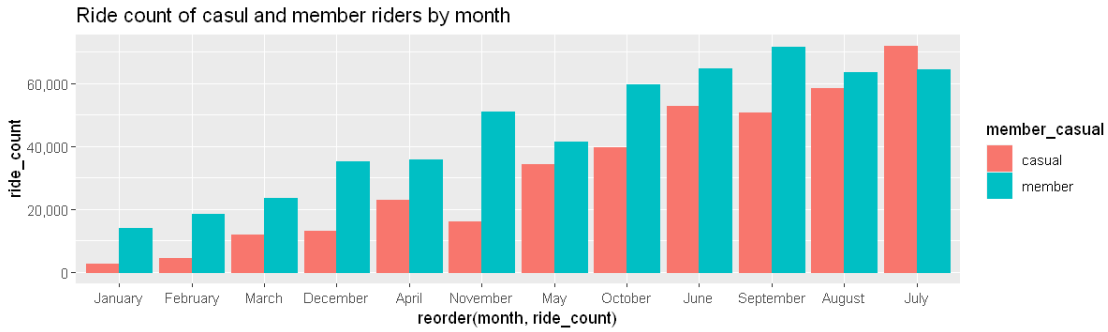
```
`summarise()` has grouped output by 'member_casual', 'weekday',
'weekend_weekday'. You can override using the `.groups` argument.
```



It clearly shows weekend ride count of casual riders is more than annual riders

[275]:
```
###  Ride count of casul and member riders by month
options(repr.plot.width = 10, repr.plot.height = 3)
new_df %>%
      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=ride_count,fill=member_casual,color=member_casual))+
      labs(title = "Ride count of casul and member riders by month" )+
      geom_col(position = "dodge") +
      scale_y_continuous(labels = comma)
```

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

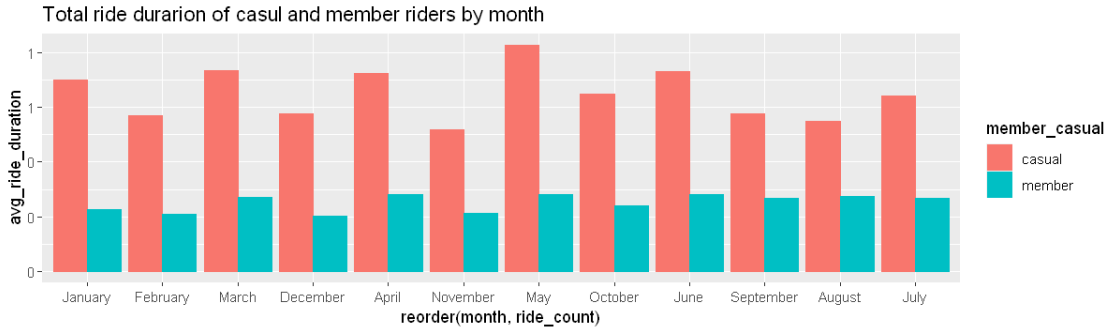


[ ]:
```
17445
```

Casual riders ride more during the year expect july month for any promotional activity towards casual riders from may to july is beter period

[274]:
```
### Total ride durarion of casul and member riders  by month
options(repr.plot.width = 10, repr.plot.height = 3)
new_df %>%
      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_casual)
      labs(title = "Total ride durarion of casul and member riders by month" )+
      geom_col(position = "dodge") +
      scale_y_continuous(labels = comma)
```

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

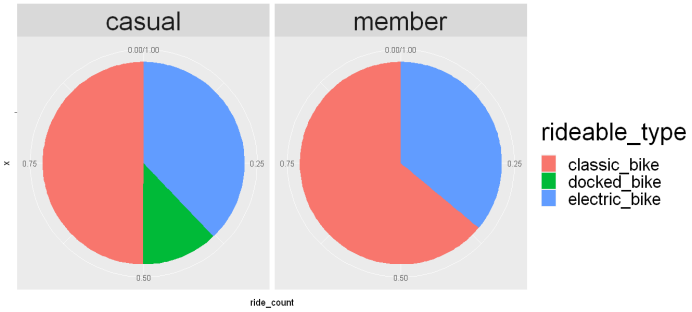


Casual riders ride time is higher than member riders in every month launching any strategy to support this behaviour of casual ridrer can help them retain.

[321]:
```
### ridable type by riders type

options(repr.plot.width = 18, repr.plot.height = 5)
new_df %>%
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)+
     theme(strip.text = element_text(size = 30),legend.title =element_text(size
 ↪=(28)),legend.text = element_text(size = 20))
```

```
`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

2.9.1 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.Casual riders ride more than annual member during the year expect jully month .for any promotional activity towards casual riders from may to july is better period

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

2.9.2 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 data clearly days 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 are riding bike for longer duration company can come up with different strategy to support this pattern

2.9.3 Is more data required?

yes, we require data demogafic data of customer to target casual rider who fall under member rider character and convert then to member