Cyclistic bike-share analysis

Yan Shao

2023-07-26

Introduction

Scenario

You are 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, your team wants to understand **how casual riders and annual members use Cyclistic bikes differently**. From these insights, your 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.

About the company

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: **single-ride passes, full-day passes, and 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.

Data source

I downloaded 12 months of Cyclistic's historical trip data(July 2022 - June 2023) from here.The data has been made available by Motivate International Inc. under this license.

Tool

- **Rstudio**
- **Tableau**
- **Github**

Business Task

- Understand how annual members and casual riders differ.

- Convert casual riders into annual members.

Process

```
##install.packages("DT") ## show datatable
## loading the packages
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts t
o become errors
```

```
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
```

```
library(lubridate)
library(DT)
```

1. Importing individual CSV files

I have stored the previous 12 months of Cyclistic trip data in the folder "original_data_202207_2023_06". When I read the data file, I need add a folder name before the file name as writing codes.

```
jul202207_df<-read_csv("0original_data_202207_2023_06/202207-divvy-tripdata.csv")
aug202208_df<-read_csv("0original_data_202207_2023_06/202208-divvy-tripdata.csv")
sep202209_df<-read_csv("0original_data_202207_2023_06/202209-divvy-tripdata.csv")
oct202210_df<-read_csv("0original_data_202207_2023_06/202210-divvy-tripdata.csv")
nov202211_df<-read_csv("0original_data_202207_2023_06/202211-divvy-tripdata.csv")
dec202212_df<-read_csv("0original_data_202207_2023_06/202212-divvy-tripdata.csv")
```

```
jan202301_df<-read_csv("0original_data_202207_2023_06/202301-divvy-tripdata.csv")
feb202302_df<-read_csv("0original_data_202207_2023_06/202302-divvy-tripdata.csv")
mar202303_df<-read_csv("0original_data_202207_2023_06/202303-divvy-tripdata.csv")
apr202304_df<-read_csv("0original_data_202207_2023_06/202304-divvy-tripdata.csv")
may202305_df<-read_csv("0original_data_202207_2023_06/202305-divvy-tripdata.csv")
jun202306_df<-read_csv("0original_data_202207_2023_06/202306-divvy-tripdata.csv")
```

2. Merging into one-year dataset

```
cyclistic_year_df<-bind_rows(jul202207_df,aug202208_df,sep202209_df,oct202210_df,nov20221
1_df,dec202212_df,jan202301_df,feb202302_df,mar202303_df,apr202304_df,may202305_df,jun202
306_df)

summary(cyclistic_year_df)

##    ride_id          rideable_type        started_at
##  Length:5779444     Length:5779444     Min.   :2022-07-01 00:00:01.00
##  Class :character   Class :character   1st Qu.:2022-08-25 15:58:58.75
##  Mode  :character   Mode  :character   Median :2022-11-02 06:55:44.00
##                                        Mean   :2022-12-13 12:55:20.82
##                                        3rd Qu.:2023-04-21 14:21:37.25
##                                        Max.   :2023-06-30 23:59:56.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2022-07-01 00:06:23.00   Length:5779444     Length:5779444
##  1st Qu.:2022-08-25 16:14:35.00   Class :character   Class :character
##  Median :2022-11-02 07:06:49.50   Mode  :character   Mode  :character
##  Mean   :2022-12-13 13:13:41.24
##  3rd Qu.:2023-04-21 14:39:04.00
##  Max.   :2023-07-10 20:26:44.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5779444     Length:5779444     Min.   :41.64   Min.   :-87.87
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng       member_casual
##  Min.   : 0.00   Min.   :-88.16   Length:5779444
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.37   Max.   :  0.00
##  NA's   :5795    NA's   :5795
```

3. Cleaning

a. selecting some useful columns and creating a new data frame to protect original data

```
cyclistic_df <- cyclistic_year_df %>%
  select(member_casual,rideable_type,started_at,ended_at,start_lat,start_lng,end_lat,end_
lng)
```

b. removing rows without values

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

c. separating data time to calculate ride length(minutes)

```
cyclistic_df <- cyclistic_df %>%
  mutate(started_date = as.Date(started_at),weekday = wday(started_date,label=TRUE), ride
_length_mins = as.numeric(difftime(ended_at,started_at,units="mins")))
```

d.filtering out error value

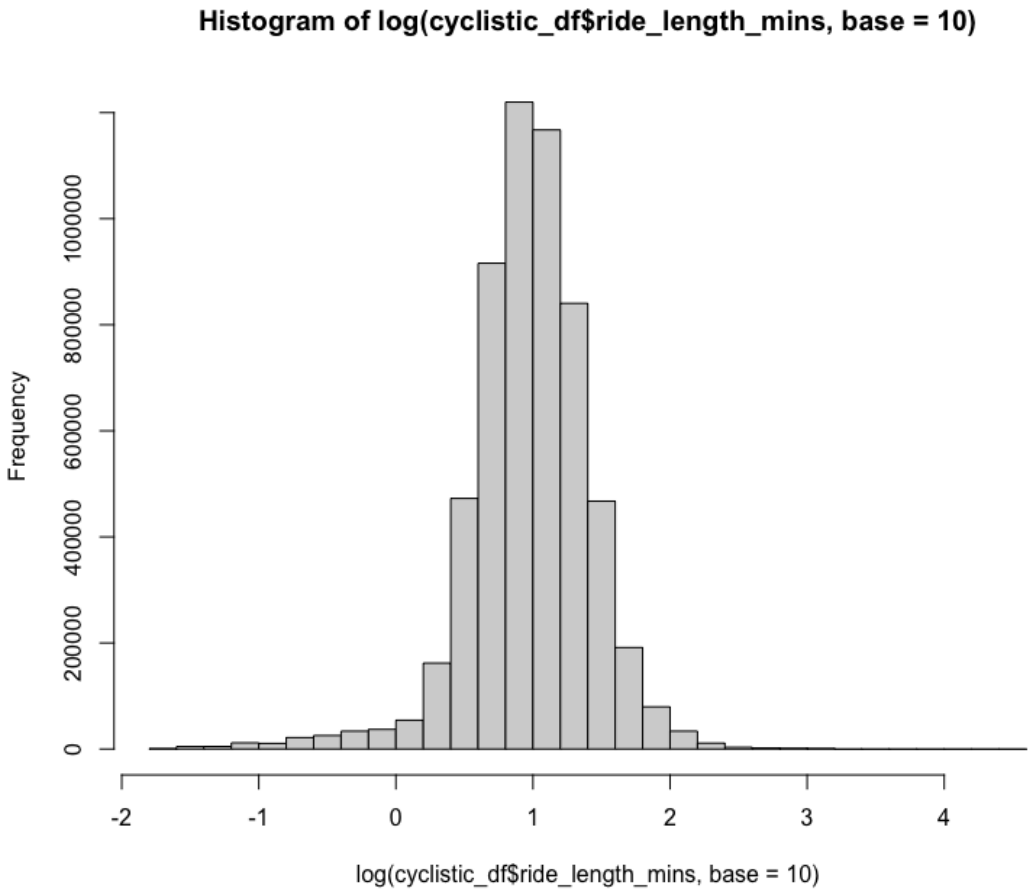
such as negative value in the ride length column and zero in the end latitude column.

```
cyclistic_df <- cyclistic_df %>% filter(ride_length_mins>0 & end_lat != 0)

head(cyclistic_df)
```

```
## # A tibble: 6 × 11
##   member_casual rideable_type started_at          ended_at            start_lat
##   <chr>         <chr>         <dttm>              <dttm>                  <dbl>
## 1 member        classic_bike  2022-07-05 08:12:47 2022-07-05 08:24:32      41.9
## 2 casual        classic_bike  2022-07-26 12:53:38 2022-07-26 12:55:31      41.9
## 3 casual        classic_bike  2022-07-03 13:58:49 2022-07-03 14:06:32      41.9
## 4 casual        classic_bike  2022-07-31 17:44:21 2022-07-31 18:42:50      41.9
## 5 member        classic_bike  2022-07-13 19:49:06 2022-07-13 20:15:24      41.9
## 6 member        electric_bike 2022-07-01 17:04:35 2022-07-01 17:13:18      41.9
## # ℹ 6 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   started_date <date>, weekday <ord>, ride_length_mins <dbl>
```

e. further removing outliers



code:hist(log(cyclistic_df$ride_length_mins,base=10))

Depending on the plot, data mainly fall in the range between 0 and 2. So I cut off the value less than 1 minute. But I thought that some values that are greater than 100 minutes may be meaningful. I used the

filter function to browse the current data set.

```
## so filter ride_length_mins >3000;>2000;>1000 to determine the final range
## ride length more than 3000 -> only casual
cnt_morethan3000<-cyclistic_df %>%
  group_by(member_casual) %>%
  filter(ride_length_mins>3000)

## ride length more than 2000 -> only casual
cnt_morethan2000<-cyclistic_df %>%
  group_by(member_casual) %>%
  filter(ride_length_mins>2000)

## ride length more than 1000 -> contain both
cnt_morethan1000<-cyclistic_df %>%
  group_by(member_casual) %>%
  filter(ride_length_mins>1000)

## filter only member -> the maximum value < 1500
cntonly_member<-cyclistic_df %>%
  filter(member_casual == "member")
```

The data showed that only casual riders provided ride length that is more than 1500 minutes. So these values could not help me understand the difference between casual rides and members.

```
cyclistic_df <- cyclistic_df %>% filter(ride_length_mins<1500 & ride_length_mins > 1)

summary(cyclistic_df)

##  member_casual      rideable_type        started_at
##  Length:5622843     Length:5622843     Min.   :2022-07-01 00:00:01.00
##  Class :character   Class :character   1st Qu.:2022-08-25 10:57:05.50
##  Mode  :character   Mode  :character   Median :2022-11-01 16:02:19.00
##                                        Mean   :2022-12-13 02:34:09.42
##                                        3rd Qu.:2023-04-21 09:51:47.50
##                                        Max.   :2023-06-30 23:59:56.00
##
##     ended_at                       start_lat       start_lng
##  Min.   :2022-07-01 00:06:23.00   Min.   :41.64   Min.   :-87.87
##  1st Qu.:2022-08-25 11:09:49.50   1st Qu.:41.88   1st Qu.:-87.66
##  Median :2022-11-01 16:15:26.00   Median :41.90   Median :-87.64
##  Mean   :2022-12-13 02:49:51.81   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:2023-04-21 10:03:40.50   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :2023-07-01 18:26:01.00   Max.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng        started_date        weekday
##  Min.   :41.55   Min.   :-88.16   Min.   :2022-07-01   Sun:718840
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:2022-08-25   Mon:711285
##  Median :41.90   Median :-87.64   Median :2022-11-01   Tue:784626
##  Mean   :41.90   Mean   :-87.65   Mean   :2022-12-12   Wed:826507
```

```
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:2023-04-21   Thu:840381
##  Max.   :42.37   Max.   :-87.30   Max.   :2023-06-30   Fri:842282
##                                                       Sat:898922
##  ride_length_mins
##  Min.   :   1.017
##  1st Qu.:   5.800
##  Median :   9.967
##  Mean   :  15.707
##  3rd Qu.:  17.600
##  Max.   :1499.933
##
```

f. exporting the dataframe as a .CSV file

write.csv(cyclistic_df,"/Users/shaoyan/Cyclistic bike-share analysis/cyclistic_data_cleaned.csv")

4. Analysis

Descriptive analysis on ride_length

```
mean(cyclistic_df$ride_length_mins)
```

```
## [1] 15.70676
```

```
median(cyclistic_df$ride_length_mins)
```

```
## [1] 9.966667
```

```
max(cyclistic_df$ride_length_mins)
```

```
## [1] 1499.933
```

```
min(cyclistic_df$ride_length_mins)
```

```
## [1] 1.016667
```

Compare members and casual users

```
aggregate(cyclistic_df$ride_length_mins ~ cyclistic_df$member_casual, FUN = mean)
```

```
##   cyclistic_df$member_casual cyclistic_df$ride_length_mins
## 1                     casual                      20.93532
## 2                     member                      12.38656
```

```
aggregate(cyclistic_df$ride_length_mins ~ cyclistic_df$member_casual, FUN = median)
```

```
##   cyclistic_df$member_casual cyclistic_df$ride_length_mins
## 1                     casual                          12.3
## 2                     member                           8.8
```

```
aggregate(cyclistic_df$ride_length_mins ~ cyclistic_df$member_casual, FUN = max)
```

```
##   cyclistic_df$member_casual cyclistic_df$ride_length_mins
## 1                     casual                      1499.917
## 2                     member                      1499.933

aggregate(cyclistic_df$ride_length_mins ~ cyclistic_df$member_casual, FUN = min)

##   cyclistic_df$member_casual cyclistic_df$ride_length_mins
## 1                     casual                      1.016667
## 2                     member                      1.016667
```

rider trip data by member type and weekday

the average ride time by the days of the week for members vs casual riders

cyclistic_df$weekday <- ordered(cyclistic_df$weekday, levels=c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat","Sun")) aggregate(cyclistic_df$ride_length_mins ~ cyclistic_df$member_casual + cyclistic_df$weekday, FUN = mean)

	member_casual	weekday	ride_length_mins
1	casual	Mon	20.73156
2	member	Mon	11.80401
3	casual	Tue	18.54757
4	member	Tue	11.79966
5	casual	Wed	18.10039
6	member	Wed	11.86181
7	casual	Thu	18.41410
8	member	Thu	11.93120
9	casual	Fri	20.22454
10	member	Fri	12.28835
11	casual	Sat	23.88718
12	member	Sat	13.90373
13	casual	Sun	24.05058
14	member	Sun	13.68963

The data result tells me that most casual riders had the longer trips. Most riders use the service for short trips that last around 13 minutes. During weekend, casual riders would have longer ride length. On the other hand, members have more consistent ride length.

Share

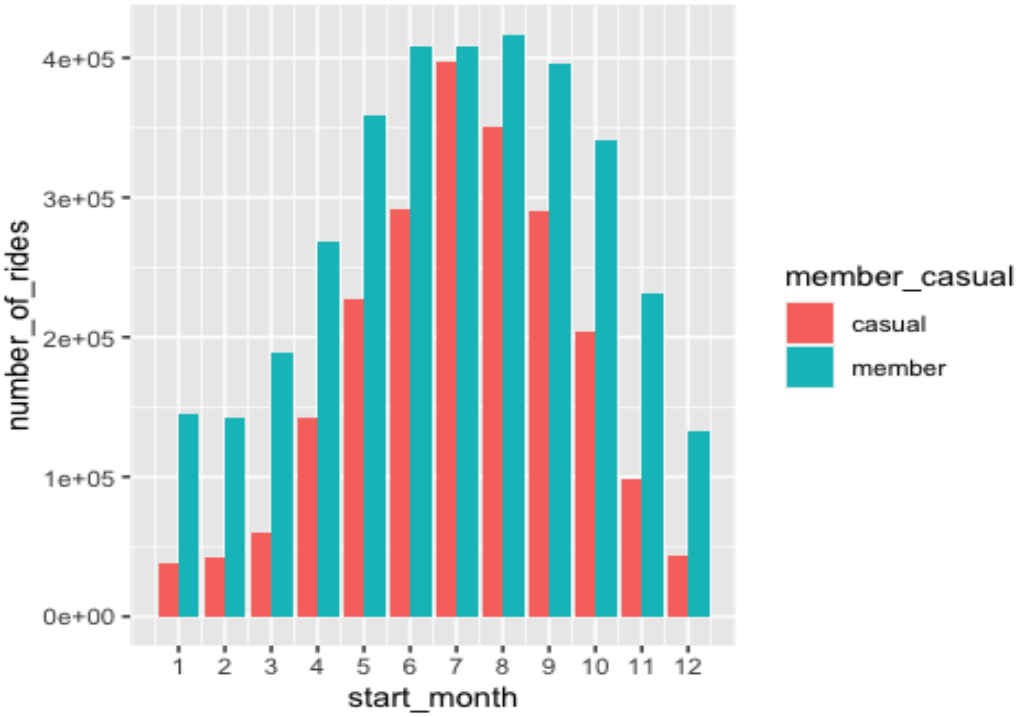
visualization

1.ggplot

the number of rides by member type by the start month

```
cyclistic_df %>% mutate(start_month = month(as.Date(started_at))) %>%
  group_by(member_casual, start_month) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length_mins)) %>%
  arrange(member_casual, start_month)  %>%
  ggplot(aes(x = start_month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+scale_x_continuous(breaks = c(1,2,3,4,5,6,7,8,9,10,11,12))
```

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



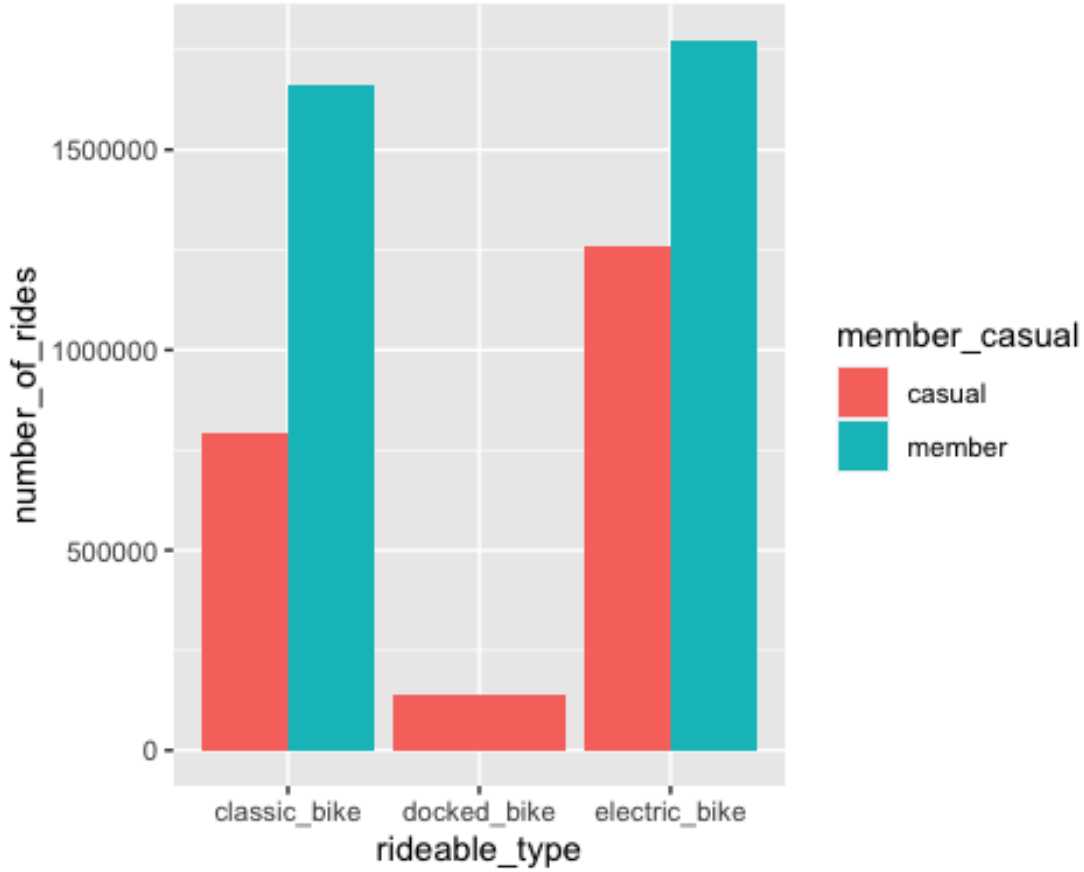
Most users prefer using bike-share service for longer periods from June to September. It might relate to warm climate during the months of the year. The number of rides of casual riders visibly dropped from November to February than members.

the number of rides by member type by the rideable type

```
cyclistic_df %>%
  group_by(member_casual, rideable_type) %>%
```

```
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length_mins)) %>%
  arrange(member_casual, rideable_type)  %>%
  ggplot(aes(x = rideable_type, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

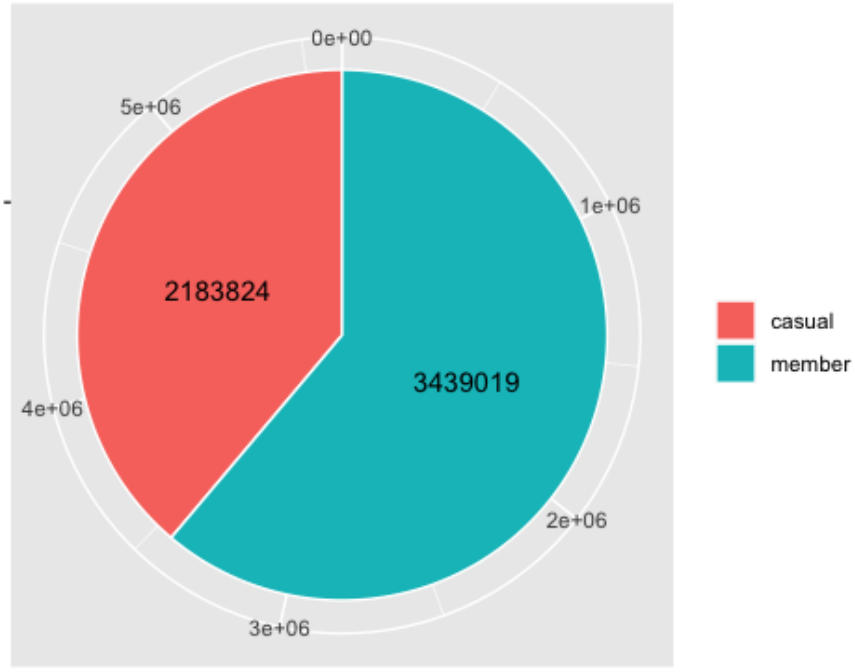
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



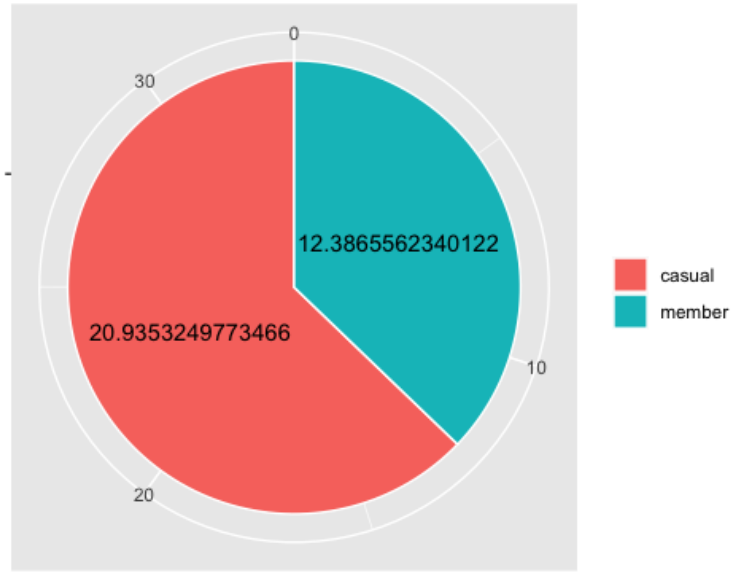
It is very interesting. The bar char shows that members during the year did not use the docked bike. For casual rides, they use more class bikes or electric bikes than docked bikes.

The number of rides by member types

```
cyclistic_df %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = "", y = number_of_rides, fill = member_casual)) +geom_bar(width = 1, sta
t = "identity", color = "white") +
  coord_polar("y", start = 0)+
  geom_text(aes(label = paste0(number_of_rides)), position = position_stack(vjust=0.5)) +
  labs(x = NULL, y = NULL, fill = NULL)
```



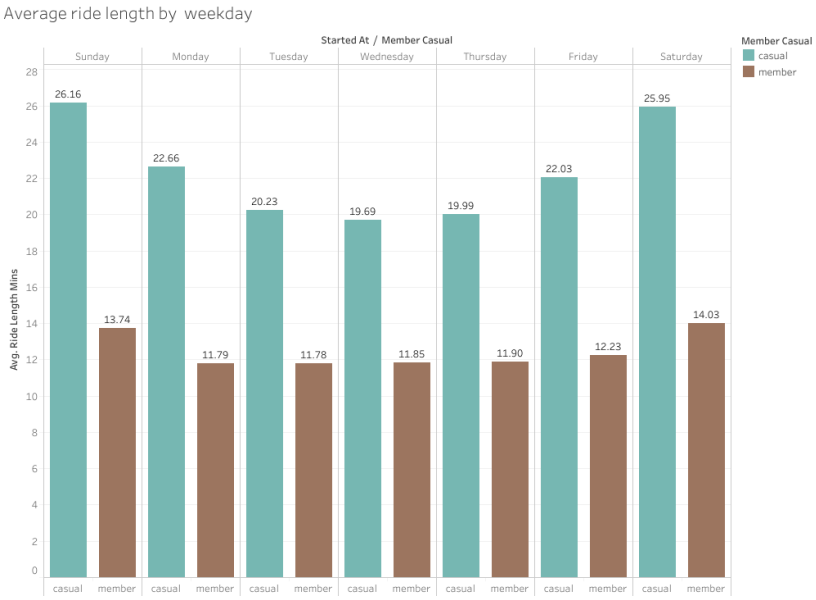

```
cyclistic_df %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length_mins)) %>%
  ggplot(aes(x = "", y = average_duration, fill = member_casual)) +geom_bar(width = 1, st
at = "identity", color = "white") +
  coord_polar("y", start = 0)+
  geom_text(aes(label = paste0(average_duration)), position = position_stack(vjust=0.5))
+
  labs(x = NULL, y = NULL, fill = NULL)
```



As shown above, member made much more trips which lasted short. On the other hand, casual riders are more likely to spend longer time in each trip.

2.Tableau

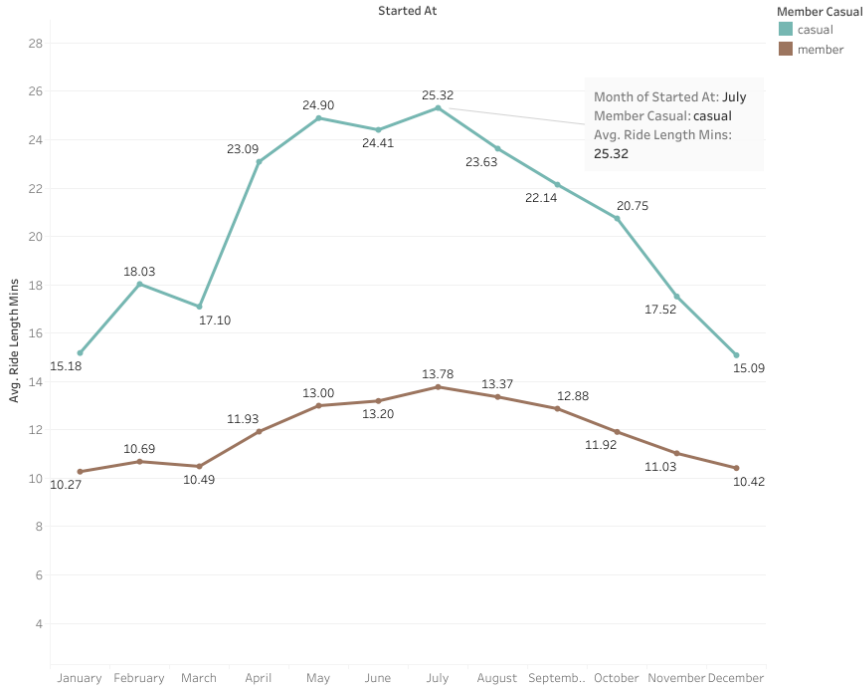
Average ride length by weekday



Here, the bar char shows that casual riders had significantly longer average trip length than members. Besides, casual rides used the service longer average trip length during weekend. On the other hand, the average trip length of members did not change obviously. It probably depends on the type of usage. For instance, casual users use it for leisure and members use it for commute.

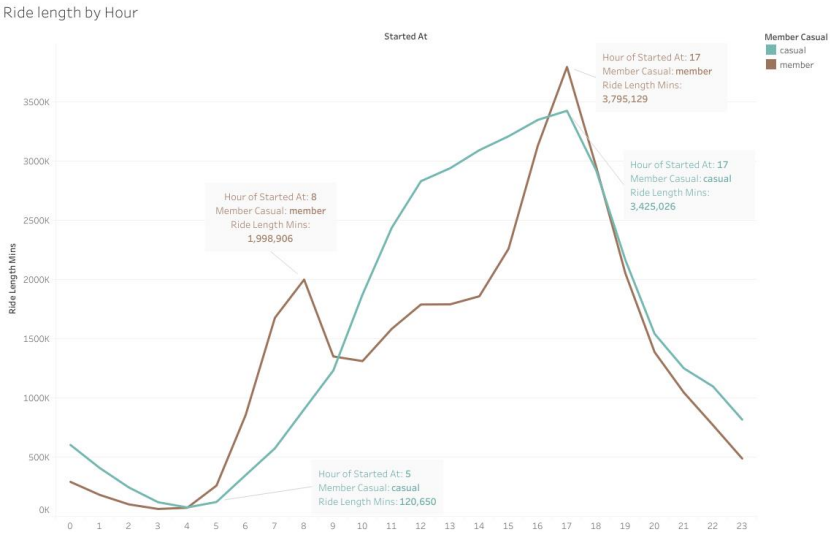
Average ride length by the start month

Average ride length by the start month



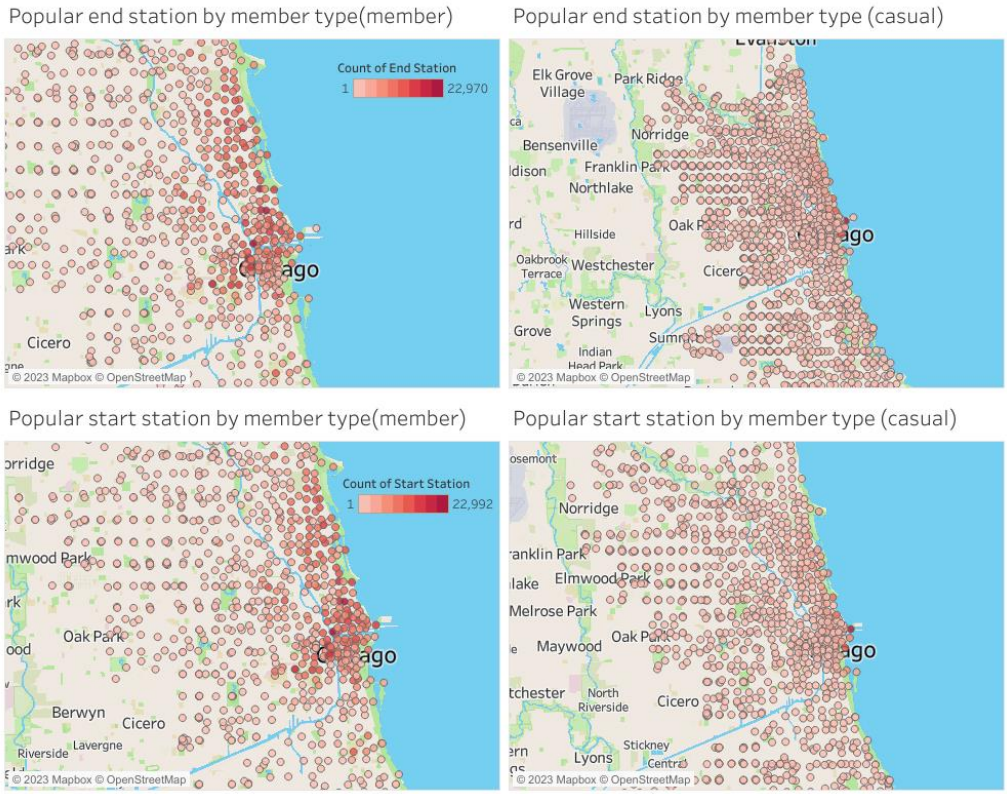
Most users prefer using bike-share service for longer periods from June to September. It might relate to warm climate during the months of the year. The number of rides of casual riders visibly dropped from November to February than members.

Ride length by Hour



Based on the ride length by hour graph, there are two peaks in 8 am and 17 pm in the member curve. It might be the timings of commute.

Popular start & end stations by member type



For members, the popular start stations (dark red dots) are distributed dispersedly. However, the popular start station of casual users merely is located near coastline. So, the casual riders might ride for leisure activities or sightseeing.

Conclusion

In the project, I mainly focused on identifying if there are the day of the week, the peak usage hour, ride length, and location, which may contribute to the difference between members and casual users. During the process, I used different tools.

- R
  - read, merge data
  - clean data
  - plot
  - make report
- Tableau

  - plot
  - create dashboard
  - share result
- Github
  - store material
  - share result

Based on these plots, I find some key points:

1. Most casual riders had longer trips on average than Members.
2. Most casual riders used the service less frequent than Members.
3. Most casual riders rode on weekends. But most Members used the service over week.
4. Most casual riders preferred using classic or electric bikes, but only casual riders used docked bikes.
5. Most members rode intensively at 8 am and 17 pm. They probably ride for commuting.
6. The popular start and end locations for casual riders are located near coastline.