Project: How Does a Bike-Share Navigate Speedy Success?

"A Case Study on Bike-Share Company"

Lihan Vicky Tu

Date: October,2023

Tool Used: R/R Studio

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.

Question:

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

Data

Data Source: Divvy Trip Data that are available to the public

Data URL: https://divvy-tripdata.s3.amazonaws.com/index.html

Data Date Range: January, 2023 to September, 2023

Data Quality: The data has already been inspected processed by Divvy to remove trips that are taken by staff during inspections as well as trips below 60 seconds because trips below 60 seconds in lengths tend to be a result of false docking practices by the users.

Data Privacy: Per Divvy's Data License Agreement, I am allowed to access, analyze, copy, modify and distribute this data while only using data as source material in analysis, reports or studies for non-commercial purposes.

The data is organized into the following columns: ride id, rideable type, time started at, time ended at, start station name, start station id, end station name, end station id, start latitutde, end longitude, end latitude, end longitude, member type.

Data Processing:

Now to load the tidyverse library to continue processing the data and proceed with the study:

```
library('tidyverse')
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
```

```
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Load data for processing:

```
Bikeshare2023 <- read.csv('/Users/Vicky/Desktop/BikeShare2023.csv')
summary(Bikeshare2023)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:4596173     Length:4596173     Length:4596173     Length:4596173
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:4596173     Length:4596173     Length:4596173     Length:4596173
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.63   Min.   :-87.94   Min.   : 0.00   Min.   :-88.16
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  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.   :-87.46   Max.   :42.18   Max.   :  0.00
##                                   NA's   :5809    NA's   :5809
##  member_casual
##  Length:4596173
##  Class :character
##  Mode  :character
##
##
##
##
```

Now I will convert the started_at and ended_at columns into date-time format and find ride_length as a new column for later analysis:

```
Bikeshare2023[['ended_at']] <- strptime(Bikeshare2023[['ended_at']], format = "%Y-%m-%d %H:%M:%S")
Bikeshare2023[['started_at']] <- strptime(Bikeshare2023[['started_at']], format = "%Y-%m-%d %H:%M:%S")
Bikeshare2023$ride_length <- difftime(Bikeshare2023$ended_at, Bikeshare2023$started_at)
#Bikeshare2023$ride_length <- seconds_to_period(Bikeshare2023[['ride_length']])
summary(Bikeshare2023)
```

```
##    ride_id          rideable_type        started_at
##  Length:4596173     Length:4596173     Min.   :2023-01-01 00:01:58.00
##  Class :character   Class :character   1st Qu.:2023-05-06 11:30:19.75
```

```
##  Mode  :character   Mode  :character   Median :2023-06-26 16:56:29.00
##                                        Mean   :2023-06-18 23:27:39.83
##                                        3rd Qu.:2023-08-12 11:12:20.75
##                                        Max.   :2023-09-30 23:59:57.00
##                                        NA's   :17
##     ended_at                      start_station_name start_station_id
##  Min.   :2023-01-01 00:02:41.00   Length:4596173     Length:4596173
##  1st Qu.:2023-05-06 11:49:32.50   Class :character   Class :character
##  Median :2023-06-26 17:10:53.50   Mode  :character   Mode  :character
##  Mean   :2023-06-18 23:46:27.21
##  3rd Qu.:2023-08-12 11:38:08.25
##  Max.   :2023-10-10 04:56:16.00
##  NA's   :19
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:4596173     Length:4596173     Min.   :41.63   Min.   :-87.94
##  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.   :-87.46
##
##     end_lat         end_lng       member_casual      ride_length
##  Min.   : 0.00   Min.   :-88.16   Length:4596173     Length:4596173
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   Class :difftime
##  Median :41.90   Median :-87.64   Mode  :character   Mode  :numeric
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.18   Max.   :  0.00
##  NA's   :5809    NA's   :5809
```

I will also create a new column indicating which day of the week the rides take place:

```
Bikeshare2023$day_of_week <- wday(Bikeshare2023$started_at)
summary(Bikeshare2023)
```

```
##    ride_id          rideable_type        started_at
##  Length:4596173     Length:4596173     Min.   :2023-01-01 00:01:58.00
##  Class :character   Class :character   1st Qu.:2023-05-06 11:30:19.75
##  Mode  :character   Mode  :character   Median :2023-06-26 16:56:29.00
##                                        Mean   :2023-06-18 23:27:39.83
##                                        3rd Qu.:2023-08-12 11:12:20.75
##                                        Max.   :2023-09-30 23:59:57.00
##                                        NA's   :17
##     ended_at                      start_station_name start_station_id
##  Min.   :2023-01-01 00:02:41.00   Length:4596173     Length:4596173
##  1st Qu.:2023-05-06 11:49:32.50   Class :character   Class :character
##  Median :2023-06-26 17:10:53.50   Mode  :character   Mode  :character
##  Mean   :2023-06-18 23:46:27.21
##  3rd Qu.:2023-08-12 11:38:08.25
##  Max.   :2023-10-10 04:56:16.00
##  NA's   :19
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:4596173     Length:4596173     Min.   :41.63   Min.   :-87.94
##  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.   :-87.46
##
##     end_lat         end_lng       member_casual      ride_length
##  Min.   : 0.00   Min.   :-88.16   Length:4596173     Length:4596173
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   Class :difftime
##  Median :41.90   Median :-87.64   Mode  :character   Mode  :numeric
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.18   Max.   :  0.00
##  NA's   :5809    NA's   :5809
##   day_of_week
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :4.000
##  Mean   :4.157
##  3rd Qu.:6.000
##  Max.   :7.000
##  NA's   :17
```

Analyzing Data:

Now to find the following descriptive statistics needed for the entire dataset:
1. mean ride_length
2. max ride_length
3. mode of day_of_week

To find mean of ride_length:

```
AverageRideLength <- seconds_to_period(mean(Bikeshare2023$ride_length, na.rm = TRUE))
```

To find max of ride_length:

```
MaxRideLength <- seconds_to_period(max(Bikeshare2023$ride_length, na.rm = TRUE))
```

To find mode of day_of_week:

```
# Create mode() function to calculate mode
mode <- function(x, na.rm = FALSE) {

  if(na.rm){ #if na.rm is TRUE, remove NA values from input x
    x = x[!is.na(x)]
  }

  val <- unique(x)
  return(val[which.max(tabulate(match(x, val)))])
}
ModeTimeofWeek <- mode(Bikeshare2023$day_of_week, na.rm = TRUE)
```

Output the Statictics:

```
print("Mean of ride_length: ")
```

```
## [1] "Mean of ride_length: "
```

```
AverageRideLength
```

```
## [1] "19M 3.19484754511313S"
```

```
print("Max of ride_length:")
```

```
## [1] "Max of ride_length:"
```

```
MaxRideLength
```

```
## [1] "68d 9H 29M 4S"
```

```
print("Mode of ride_length:")
```

```
## [1] "Mode of ride_length:"
```

```
ModeTimeofWeek
```

```
## [1] 7
```

Now I will summary the data based on member_casual and day_of_week. I will first find the average ride_length and number of rides per member_casual:

```
Bikeshare2023%>%
  group_by(member_casual) %>%
  summarize(AverageRideLength = seconds_to_period(mean(ride_length, na.rm = TRUE)))
```

```
## # A tibble: 2 x 2
##   member_casual AverageRideLength
##   <chr>         <Period>
## 1 casual        29M 31.0525130096767S
## 2 member        12M 43.5071133808793S
```

```
Bikeshare2023%>%
  group_by(member_casual) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   member_casual [2]
##   member_casual       n
##   <chr>           <int>
## 1 casual        1732044
## 2 member        2864129
```

I will then find the average ride_length and number of rides per day of the week:

```
Bikeshare2023%>%
  group_by(day_of_week) %>%
  summarize(AverageRideLength = seconds_to_period(mean(ride_length, na.rm = TRUE)))
```

```
## # A tibble: 8 x 2
##   day_of_week AverageRideLength
##         <dbl> <Period>
## 1           1 23M 23.7624280674052S
## 2           2 17M 54.9261301793365S
## 3           3 16M 38.275175835377S
## 4           4 16M 12.8104509512663S
## 5           5 16M 40.5557251923335S
## 6           6 18M 49.9195000003052S
## 7           7 23M 29.891951298991S
## 8          NA NA
```

```r
Bikeshare2023%>%
  group_by(day_of_week) %>%
  count()
```

```
## # A tibble: 8 x 2
## # Groups:   day_of_week [8]
##   day_of_week      n
##         <dbl>  <int>
## 1           1 594482
## 2           2 571550
## 3           3 649472
## 4           4 660096
## 5           5 685122
## 6           6 694328
## 7           7 741106
## 8          NA     17
```

Now I will find the average ride_length and number of rides based on both member_casual and day of the week:

```r
Bikeshare2023%>%
  group_by(day_of_week, member_casual) %>%
  summarize(AverageRideLength = seconds_to_period(mean(ride_length, na.rm = TRUE)))
```

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

```
## # A tibble: 16 x 3
## # Groups:   day_of_week [8]
##    day_of_week member_casual AverageRideLength
##          <dbl> <chr>         <Period>
##  1           1 casual        33M 59.2583695367885S
##  2           1 member        14M 11.3582123808686S
##  3           2 casual        29M 12.2656221920625
##  4           2 member        12M 4.86783685039018S
##  5           3 casual        26M 23.350179724288S
##  6           3 member        12M 11.5555732392112S
##  7           4 casual        25M 25.7548366988715
##  8           4 member        12M 3.10292785282229S
##  9           5 casual        25M 54.9021460328465S
## 10           5 member        12M 8.85815888182356S
## 11           6 casual        28M 28.5718203400838S
## 12           6 member        12M 43.7026132726373S
## 13           7 casual        33M 24.7433323731935
## 14           7 member        14M 15.0329386604883S
## 15          NA casual        NA
## 16          NA member        NA
```

```r
Bikeshare2023%>%
  group_by(day_of_week, member_casual) %>%
  count()
```

```
## # A tibble: 16 x 3
## # Groups:   day_of_week, member_casual [16]
##    day_of_week member_casual      n
##          <dbl> <chr>          <int>
```

```
##  1           1 casual        276449
##  2           1 member        318033
##  3           2 casual        194740
##  4           2 member        376810
##  5           3 casual        203367
##  6           3 member        446105
##  7           4 casual        205359
##  8           4 member        454737
##  9           5 casual        225347
## 10           5 member        459775
## 11           6 casual        269110
## 12           6 member        425218
## 13           7 casual        357665
## 14           7 member        383441
## 15          NA casual             7
## 16          NA member            10
```

Data Visualization

The two charts below show the average ride_length and number of rides based on member_casual

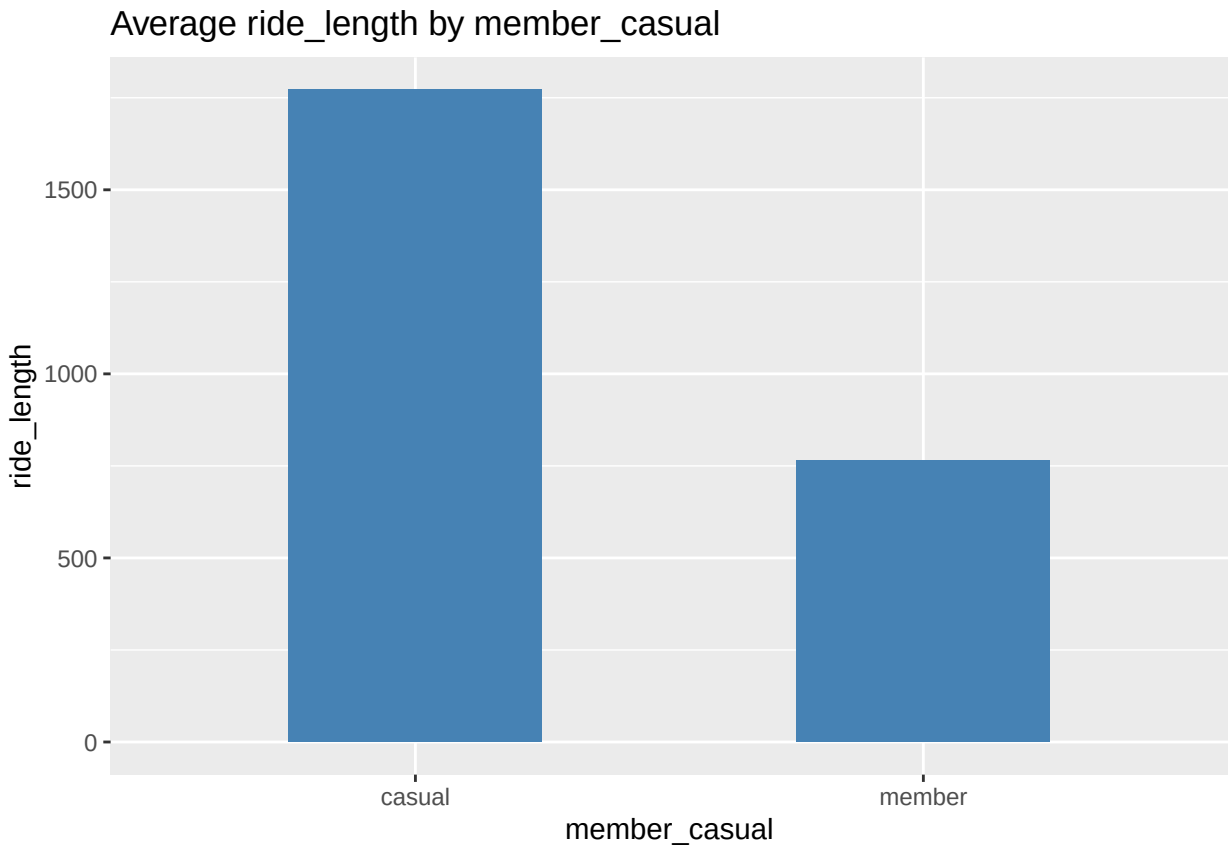
```
ggplot(data=Bikeshare2023, aes(x = member_casual, y = ride_length)) +
  geom_bar(stat="summary", fun.y='mean', fill = "steelblue", width=0.5) + ggtitle("Average ride_length
```

```
## Warning in geom_bar(stat = "summary", fun.y = "mean", fill = "steelblue", :
## Ignoring unknown parameters: `fun.y`
```

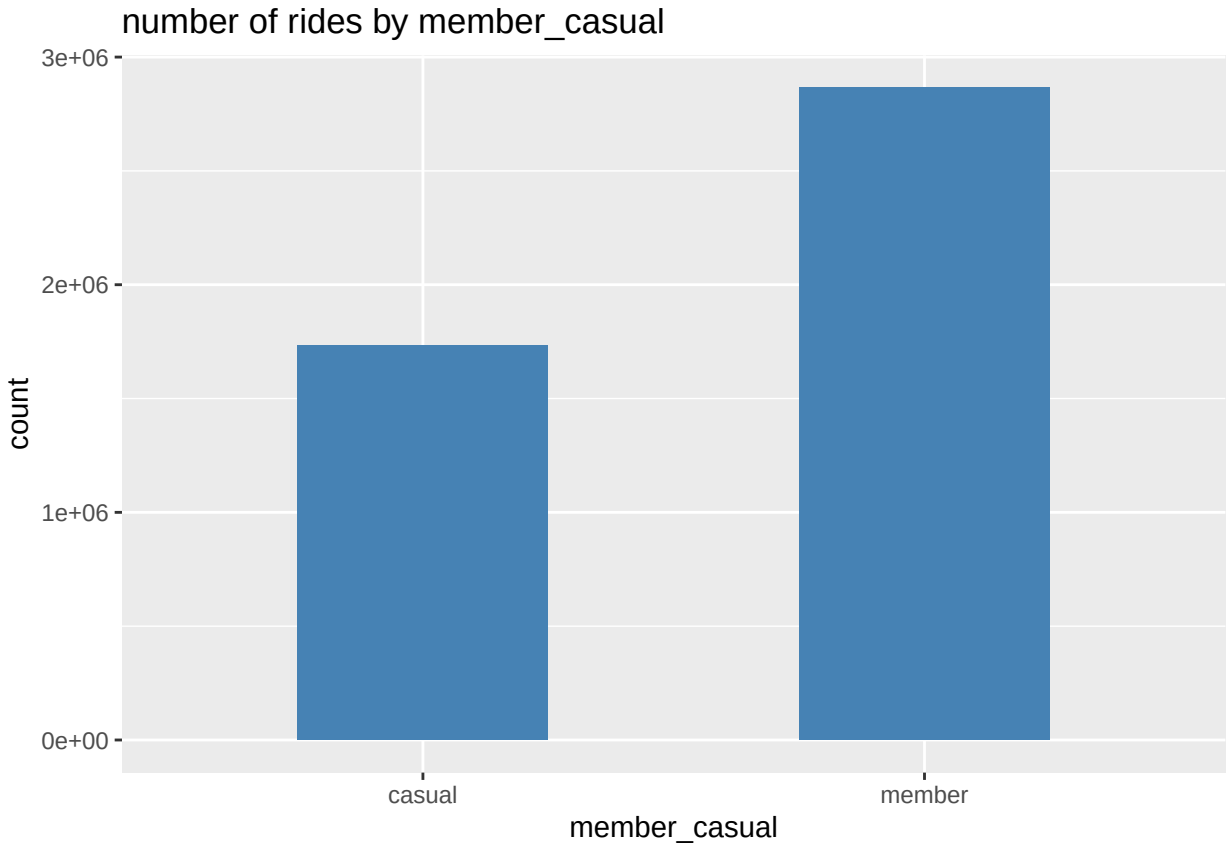
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

```
## Warning: Removed 36 rows containing non-finite values (`stat_summary()`).
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
ggplot(data=Bikeshare2023, aes(member_casual)) +
  geom_bar(fill = "steelblue", width = 0.5) + ggtitle("number of rides by member_casual")
```



The two charts below show the average ride_length and number of rides based on day_of_week

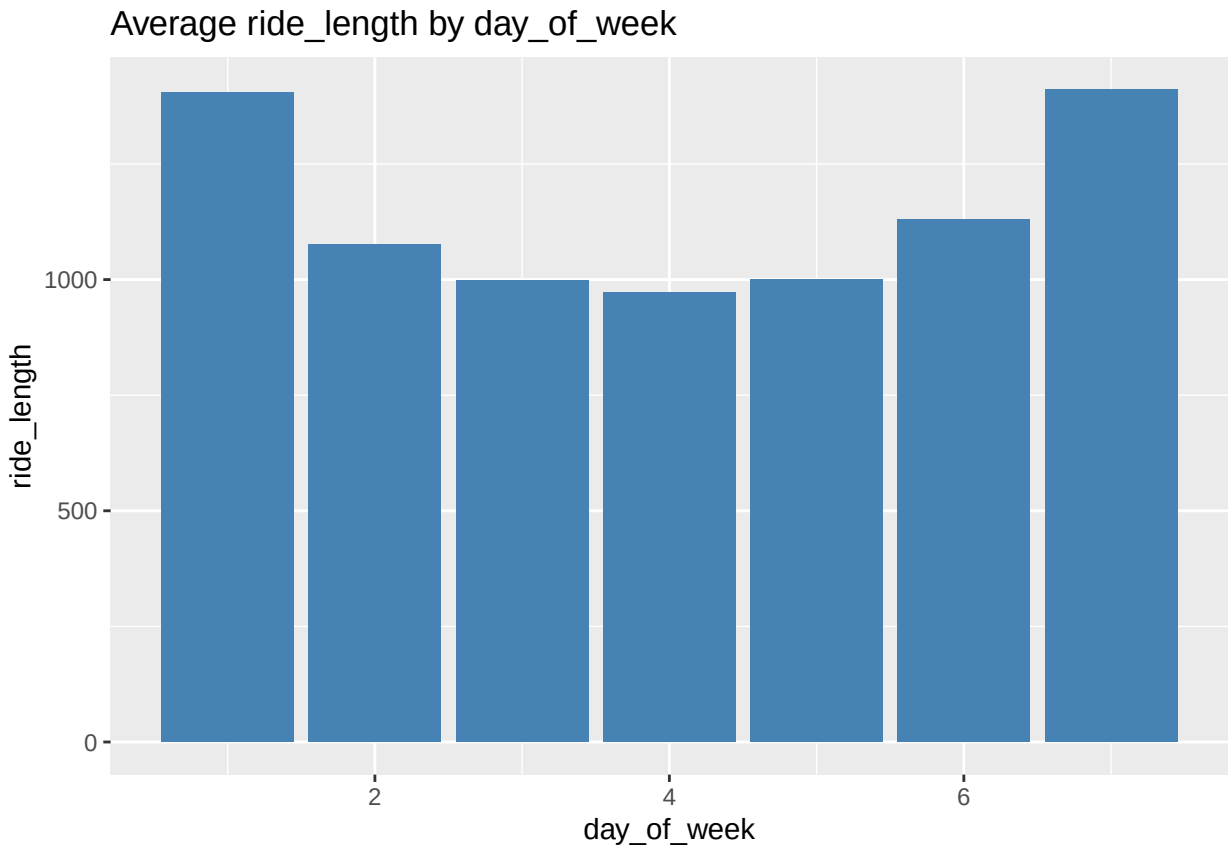
```
ggplot(data=Bikeshare2023, aes(x = day_of_week, y = ride_length)) +
  geom_bar(stat="summary", fun.y='mean', fill = "steelblue") + ggtitle("Average ride_length by day_of_w
```

```
## Warning in geom_bar(stat = "summary", fun.y = "mean", fill = "steelblue"):
## Ignoring unknown parameters: `fun.y`
```

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

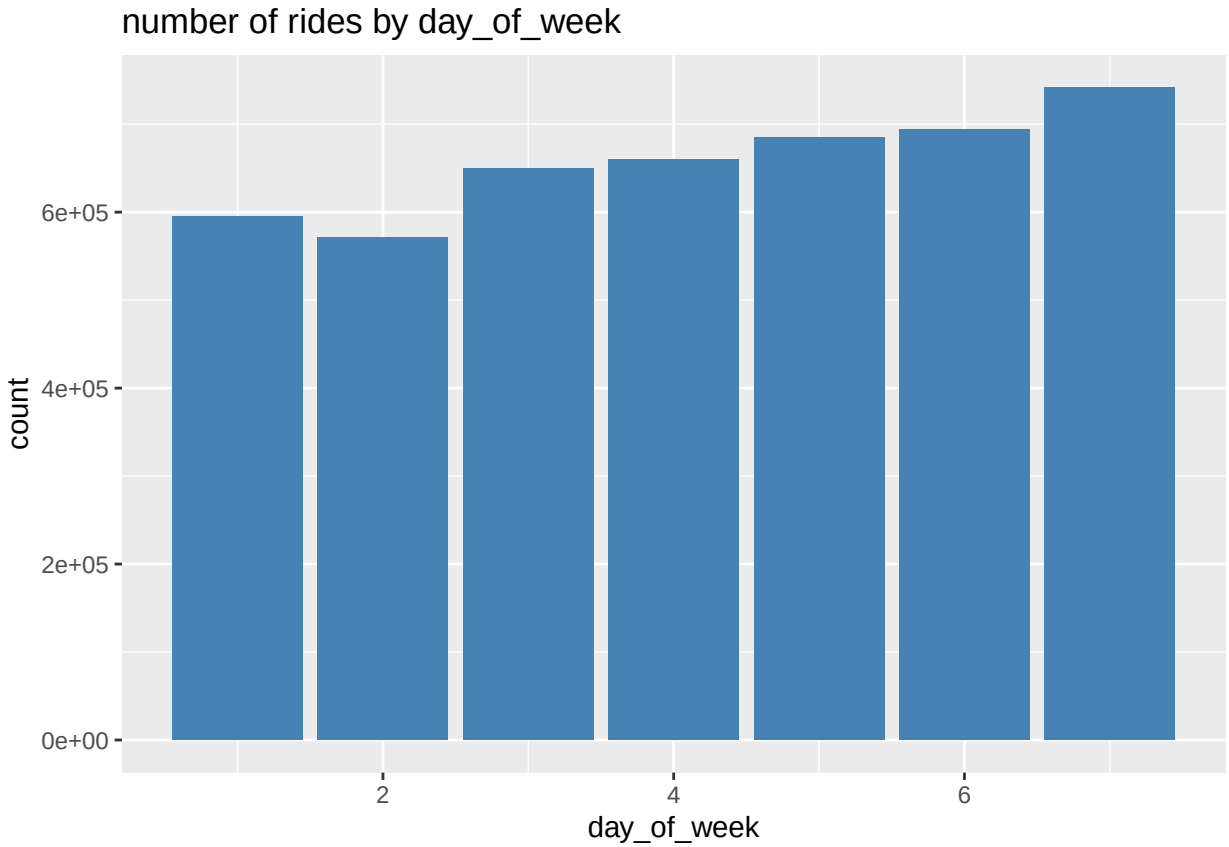
```
## Warning: Removed 36 rows containing non-finite values (`stat_summary()`).
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
ggplot(data=Bikeshare2023, aes(day_of_week)) +
  geom_bar(fill = "steelblue") + ggtitle("number of rides by day_of_week")
```

```
## Warning: Removed 17 rows containing non-finite values (`stat_count()`).
```



The chart below shows the comparison of average ride_length between member_casual by day_of_week

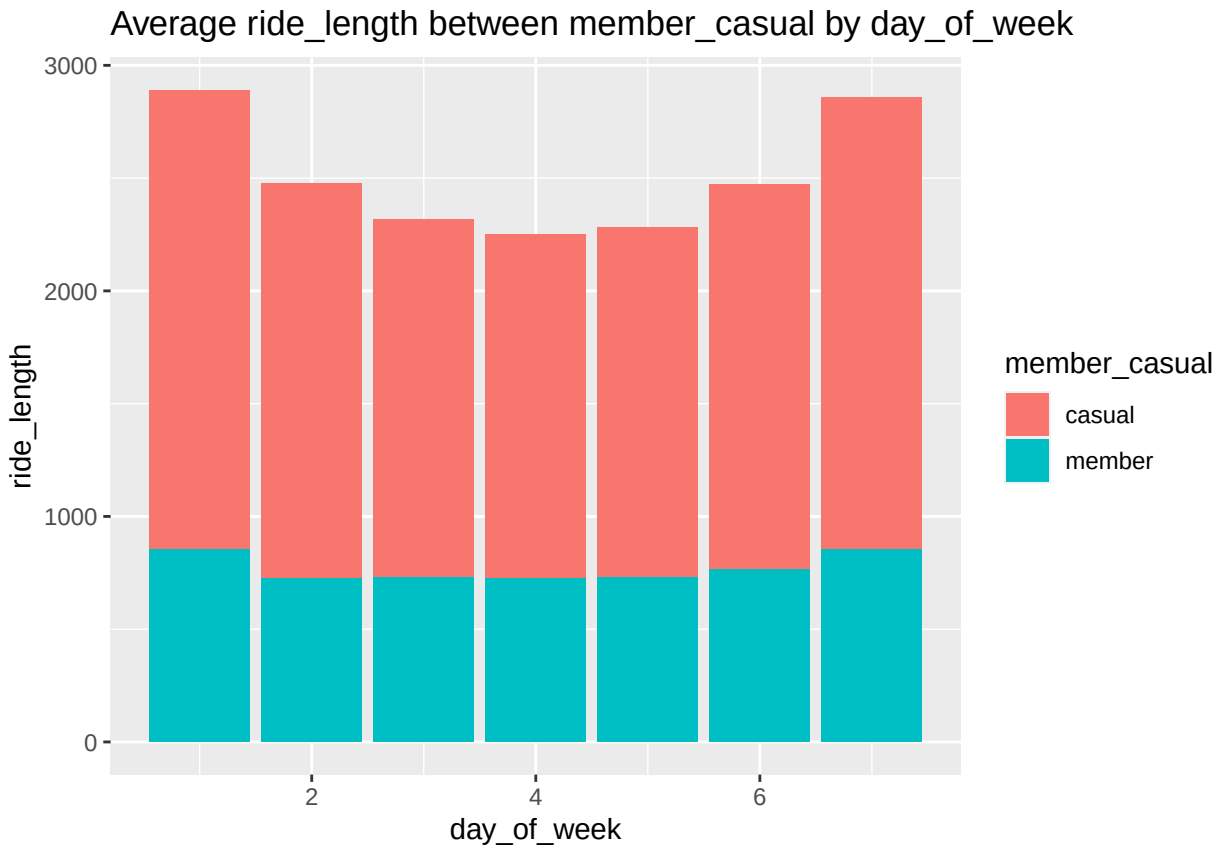
```
ggplot(Bikeshare2023, aes(x = day_of_week, y = ride_length, fill = member_casual)) +
  geom_bar(stat = "summary", fun.y='mean') + ggtitle("Average ride_length between member_casual by day_
```

```
## Warning in geom_bar(stat = "summary", fun.y = "mean"): Ignoring unknown
## parameters: `fun.y`
```

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

```
## Warning: Removed 36 rows containing non-finite values (`stat_summary()`).
```

```
## No summary function supplied, defaulting to `mean_se()`
```



The chart below shows the comparison of number of rides between member_casual by day_of_week

```
ggplot(Bikeshare2023, aes(x = day_of_week, fill = member_casual)) +
  geom_bar()+ ggtitle("Number of rides between member_casual by day_of_week")
```

```
## Warning: Removed 17 rows containing non-finite values (`stat_count()`).
```

Result and Recommendations

The data shows that even though members take many more rides than casual riders, the average ride length for casual riders is much longer than that of the members.

Both members and casual riders tend to do longer rides during the weekends. The difference by casual_member and by day of the week seems to revolve more around the number of rides:

Overall the number of rides' lowest point is on monday and steadily increasing as the week goes by and peak on saturday. The key difference here between casual riders and members is that casual riders do more rides on weekends while members do more rides on weekdays.

Based on the result, I would offer the 3 following recommendations:

1. Conduct more in-depth analysis on the bike riding tendencies for casual riders on weekends. Since they do more rides on weekends and they tend to do longer rides, it would be interesting to get more insights on potential purposes of their trips to get to know their needs better.
2. Since overall number of rides peak on Saturday, it is reasonable to consider offer a Saturday-ridership package and increase bike maintenance and services to make sure that there will be sufficient bikes for riders to access on Saturdays.
3. Since all riders tend to do shorter rides on weekdays and longer rides on weekends, I would recommend increase the the amount of time a rider can use a bike for during a single ride on weekends.

Since there is geographical location data for each ride, further more detailed analysis can be conducted to get more idea of potentially what the riders are using these bikes for to better understand their needs.