Cyclist_Data_Analysis_220f

Afzi

2025-02-22

Introduction

This is a Capstone Project for Google Data Analytics Certificate, the Dataset we are using refers to a company called Cyclistic. This is a bike-share company in Chicago, the director believes that future of the company depends on converting the casual memberships into annual memberships because annual memberships are more profitable. Our task is to understand various the behaviour of both membership types so we can help the marketing department in converting the casual members.

DATA Preparation

First We load all the libraries in our Environment. (Note: You may need to install these packages if not already installed)

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── 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 to become errors
```

```
library(ggplot2)
library(dplyr)
library(lubridate)
library(geosphere)
library(knitr)
library(rmarkdown)
```

We then read the csv files from our downloaded folder and merge them in 1 dataframe so it will be much easier to analyze and we wont have to repeat any cleaning and preparation process.

```
current_path <- getwd()
csv_path <- paste0(current_path, "/cyclist_data_all_csv")
csv_files <- list.files(path = csv_path, pattern = "*.csv", full.names = TRUE)
data_list <- lapply(csv_files, read.csv)
combined_data <- bind_rows(data_list)
```

We will view the data to get an overview.

```
head(combined_data)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 FCB05EB1758F85E8  classic_bike 2024-02-03 14:14:18 2024-02-03 14:21:00
## 2 7FB986AD5D3DE9D6  classic_bike 2024-02-05 21:10:06 2024-02-05 21:15:44
## 3 40CA13E15B5B470D electric_bike 2024-02-05 15:10:44 2024-02-05 15:12:32
## 4 D47A1660919E8861  classic_bike 2024-02-15 12:40:34 2024-02-15 12:44:24
## 5 4CD173D11BA019F8  classic_bike 2024-02-14 12:28:36 2024-02-14 12:36:59
## 6 DA5032C0CA737AF5 electric_bike 2024-02-16 00:54:48 2024-02-16 01:01:47
##             start_station_name start_station_id             end_station_name
## 1        Clark St & Newport St              632 Southport Ave & Waveland Ave
## 2 Michigan Ave & Washington St            13001       Wabash Ave & Grand Ave
## 3    Leavitt St & Armitage Ave     TA1309000029 Milwaukee Ave & Wabansia Ave
## 4 Southport Ave & Waveland Ave            13235  Southport Ave & Belmont Ave
## 5      Wentworth Ave & 35th St     KA1503000005        Shields Ave & 31st St
## 6   Sheridan Rd & Lawrence Ave     TA1309000041        Clark St & Newport St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1          13235  41.94454 -87.65468 41.94815 -87.66394        member
## 2   TA1307000117  41.88398 -87.62468 41.89147 -87.62676        member
## 3          13243  41.91760 -87.68250 41.91262 -87.68139        member
## 4          13229  41.94815 -87.66394 41.93948 -87.66375        member
## 5   KA1503000038  41.83078 -87.63250 41.83846 -87.63541        casual
## 6            632  41.96942 -87.65479 41.94454 -87.65468        member
```

```
str(combined_data)
```

```
## 'data.frame':    5854384 obs. of  13 variables:
##  $ ride_id           : chr  "FCB05EB1758F85E8" "7FB986AD5D3DE9D6" "40CA13E15B5B470D" "D47A1660919E8861" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2024-02-03 14:14:18" "2024-02-05 21:10:06" "2024-02-05 15:10:44" "2024-02-15 12:40:34" ...
##  $ ended_at          : chr  "2024-02-03 14:21:00" "2024-02-05 21:15:44" "2024-02-05 15:12:32" "2024-02-15 12:44:24" ...
##  $ start_station_name: chr  "Clark St & Newport St" "Michigan Ave & Washington St" "Leavitt St & Armitage Ave" "Southport
Ave & Waveland Ave" ...
##  $ start_station_id  : chr  "632" "13001" "TA1309000029" "13235" ...
##  $ end_station_name  : chr  "Southport Ave & Waveland Ave" "Wabash Ave & Grand Ave" "Milwaukee Ave & Wabansia Ave" "South
port Ave & Belmont Ave" ...
##  $ end_station_id    : chr  "13235" "TA1307000117" "13243" "13229" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ start_lng         : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
glimpse(combined_data)
```

```
## Rows: 5,854,384
## Columns: 13
## $ ride_id            <chr> "FCB05EB1758F85E8", "7FB986AD5D3DE9D6", "40CA13E15B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "electric_bike", "c…
## $ started_at         <chr> "2024-02-03 14:14:18", "2024-02-05 21:10:06", "2024…
## $ ended_at           <chr> "2024-02-03 14:21:00", "2024-02-05 21:15:44", "2024…
## $ start_station_name <chr> "Clark St & Newport St", "Michigan Ave & Washington…
## $ start_station_id   <chr> "632", "13001", "TA1309000029", "13235", "KA1503000…
## $ end_station_name   <chr> "Southport Ave & Waveland Ave", "Wabash Ave & Grand…
## $ end_station_id     <chr> "13235", "TA1307000117", "13243", "13229", "KA15030…
## $ start_lat          <dbl> 41.94454, 41.88398, 41.91760, 41.94815, 41.83078, 4…
## $ start_lng          <dbl> -87.65468, -87.62468, -87.68250, -87.66394, -87.632…
## $ end_lat            <dbl> 41.94815, 41.89147, 41.91262, 41.93948, 41.83846, 4…
## $ end_lng            <dbl> -87.66394, -87.62676, -87.68139, -87.66375, -87.635…
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "…
```

After we have a basic understanding of the data we are working with we can move on the data Cleaning and Preparetion steps.

```
na_counts <- sapply(combined_data, function(x) sum(is.na(x)))
print(na_counts)
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               7005               7005
##      member_casual
##                  0
```

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

combined_data$started_at <- as_datetime(combined_data$started_at)
combined_data$ended_at <- as_datetime(combined_data$ended_at)
combined_data$ride_length <- combined_data$ended_at - combined_data$started_at
combined_data$weekday_started_at <- weekdays(combined_data$started_at)
combined_data$weekday_ended_at <- weekdays(combined_data$ended_at)
combined_data$hour_started_at <- hour(combined_data$started_at)
combined_data$hour_ended_at <- hour(combined_data$ended_at)
combined_data$month_started_at <- month(combined_data$started_at)
combined_data$journey <- paste0(combined_data$start_station_name, "_", combined_data$end_station_name)
```

DATA Analysis

Now that data

```
mean(combined_data$ride_length)
```

```
## Time difference of 926.5758 secs
```

```
max(combined_data$ride_length)
```

```
## Time difference of 90562 secs
```

```
mode_function <- function(x) {
  uniq_x <- unique(x)
  uniq_x[which.max(tabulate(match(x, uniq_x)))]
}

mode_function(combined_data$weekday_started_at)
```

```
## [1] "Saturday"
```

```
mode_function(combined_data$weekday_ended_at)
```

```
## [1] "Saturday"
```

Now that we have seen basic mean, max and mode of the data we will move forward with our analysis and In this Capstone I will be using majorly visualization so that it is easy to understand for all people.

We will start our analysis by checking the average ride length for each membership type.

```
avg_ride_length <- combined_data %>%
  group_by(member_casual) %>%
  summarize(avg_length = mean(ride_length, na.rm = TRUE))
```

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

The above chart indicates that the casual members have significantly more ride lengths on average as compared to annual members

Now we will see the Average ride lengths by Weekdays Membership wise.

```
avg_ride_length_by_day <- combined_data %>%
  group_by(member_casual, weekday_started_at) %>%
  summarize(avg_length = mean(ride_length, na.rm = TRUE)) %>% arrange(member_casual, weekday_started_at)
```

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

```
weekday_order <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

# Arrange data frame by weekday order using factor and arrange
avg_ride_length_by_day <- avg_ride_length_by_day %>%
  mutate(day_of_week = factor(weekday_started_at, levels = weekday_order)) %>%
  arrange(day_of_week)
```

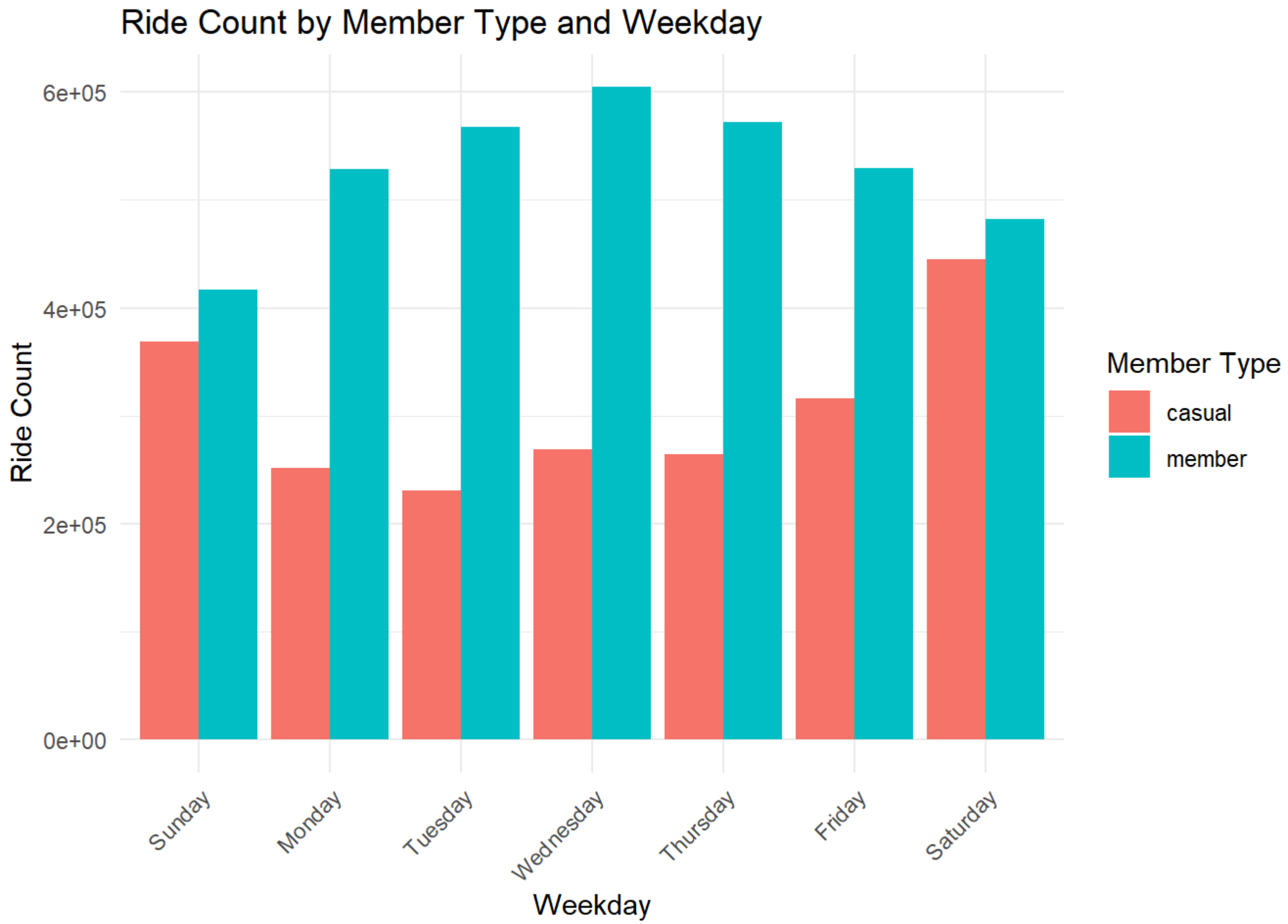
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

The Above chart indicates that we have an increase in ride length on Weekends specifically from Casual members.

We will see the count of rides used by both membership types now.

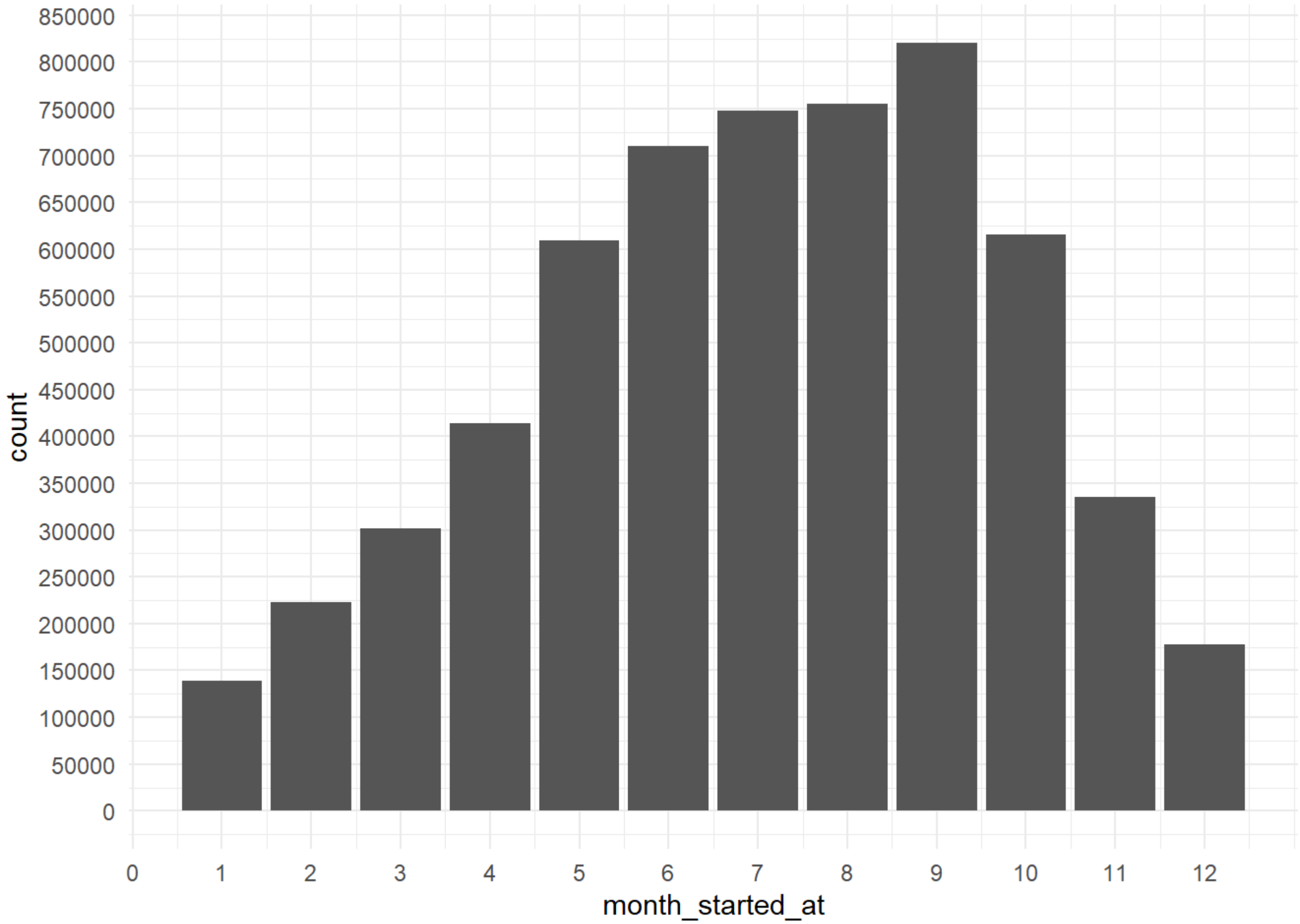
```
ride_count_by_day <- combined_data %>%
  group_by(member_casual, weekday_started_at) %>%
  summarise(ride_count = n()) %>%
  ungroup() %>%
  mutate(weekday_started_at = factor(weekday_started_at, levels = weekday_order))
```

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



We can clearly see that the number of rides in weekdays is greater for annual members and on weekend there is a sharp increase in demand by casual members. We can also note that the number of rides are highest on wednesday for annual members.

Now we will see what effects months or season has on the number of rides on both membership types.

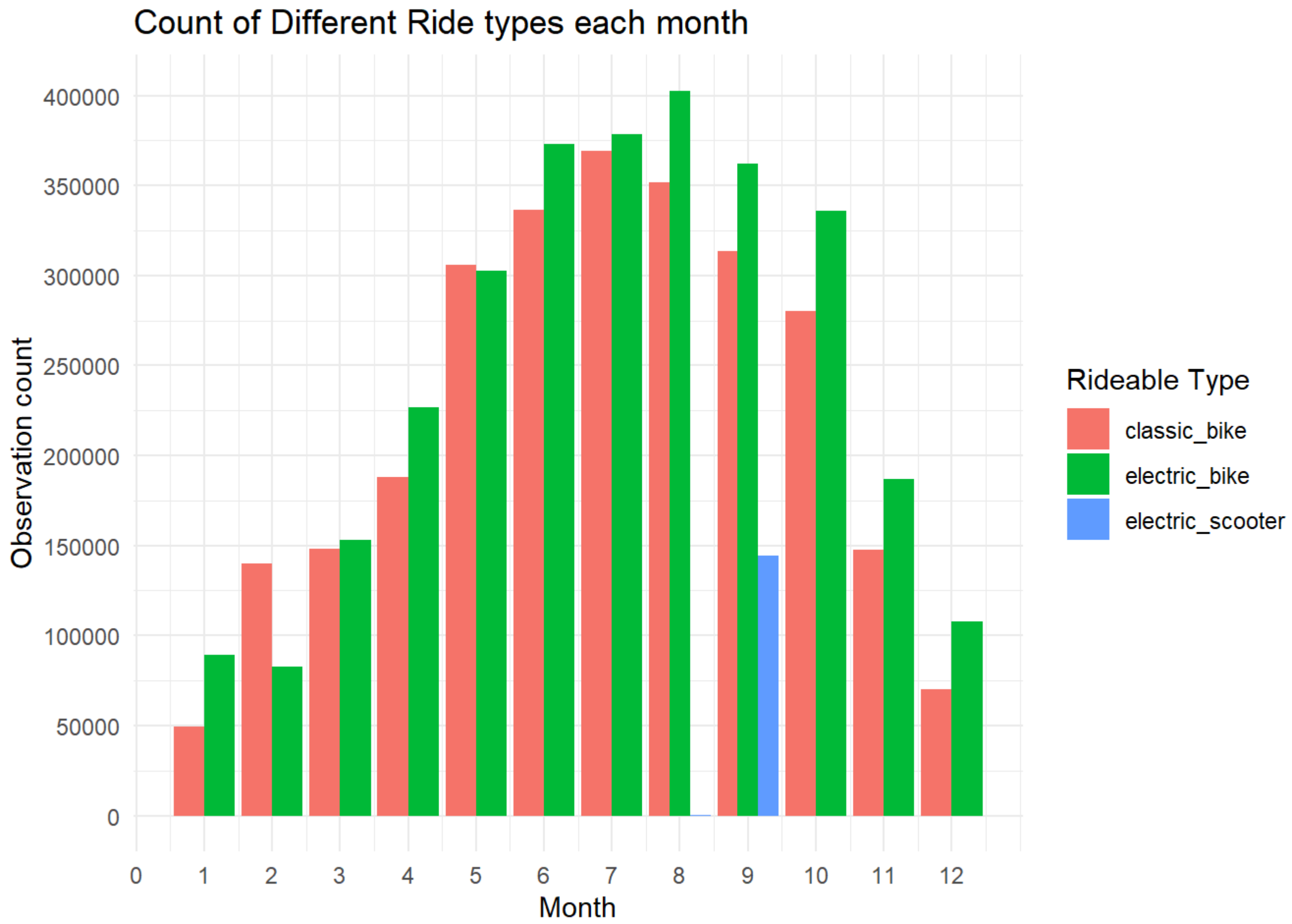


According to further research I have concluded that people are not using cyclists much in cold and when summer arrives more and more users are using cyclists.

Now we will see rideable type as per each month.

```
month_ridetype_count = combined_data %>%
  group_by(month_started_at, rideable_type) %>% summarize(observations_count = n())
```

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

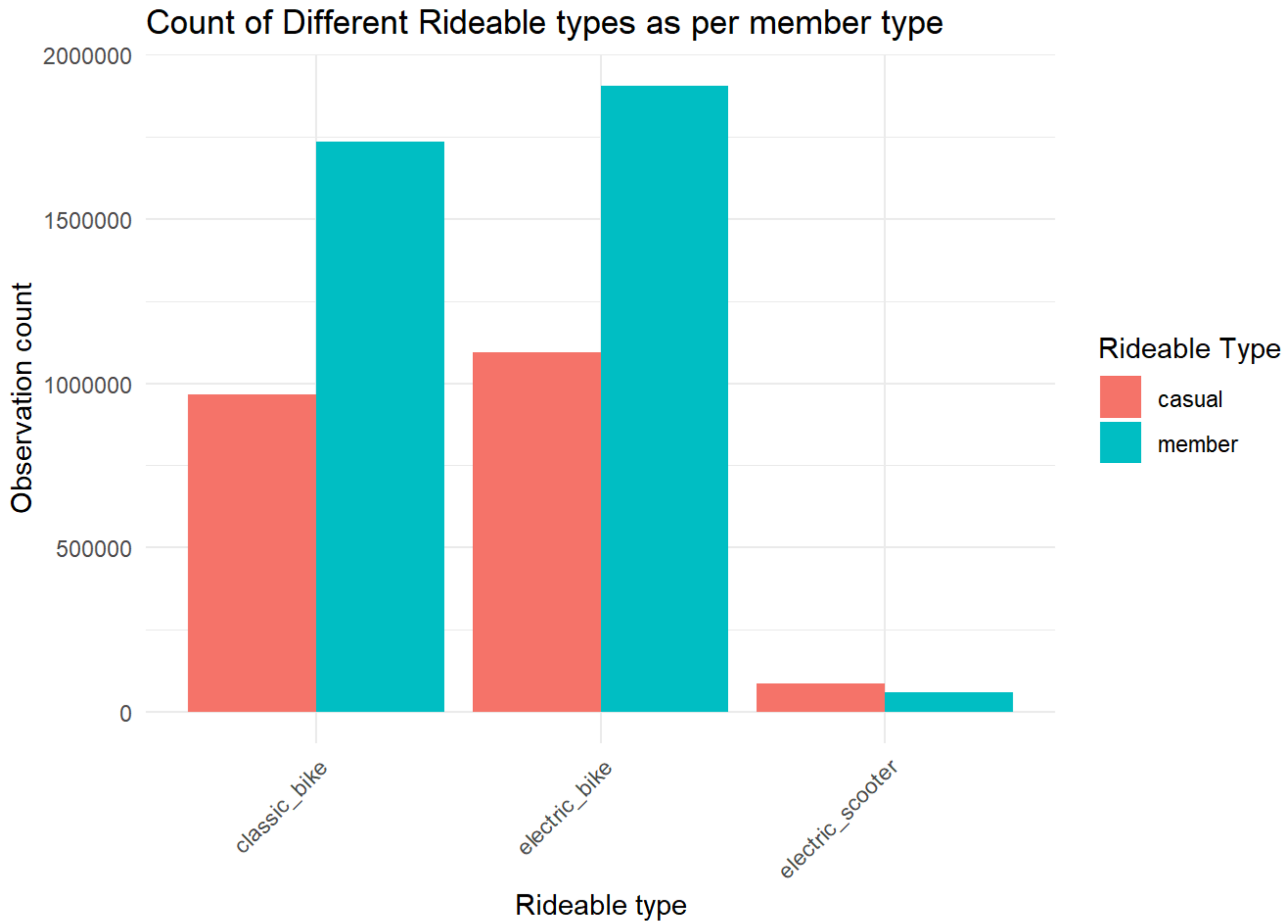


Both classic bike and electric bikes are used equally, electric bikes being slightly more prefered, also there has been electric scooter used for 1 month then closed would investigate further if possible.

Next we will check the what type of rides are used by both membership types.

```
member_ride_type <- combined_data %>% group_by(member_casual, rideable_type) %>%
  summarize(observations_count = n())
```

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

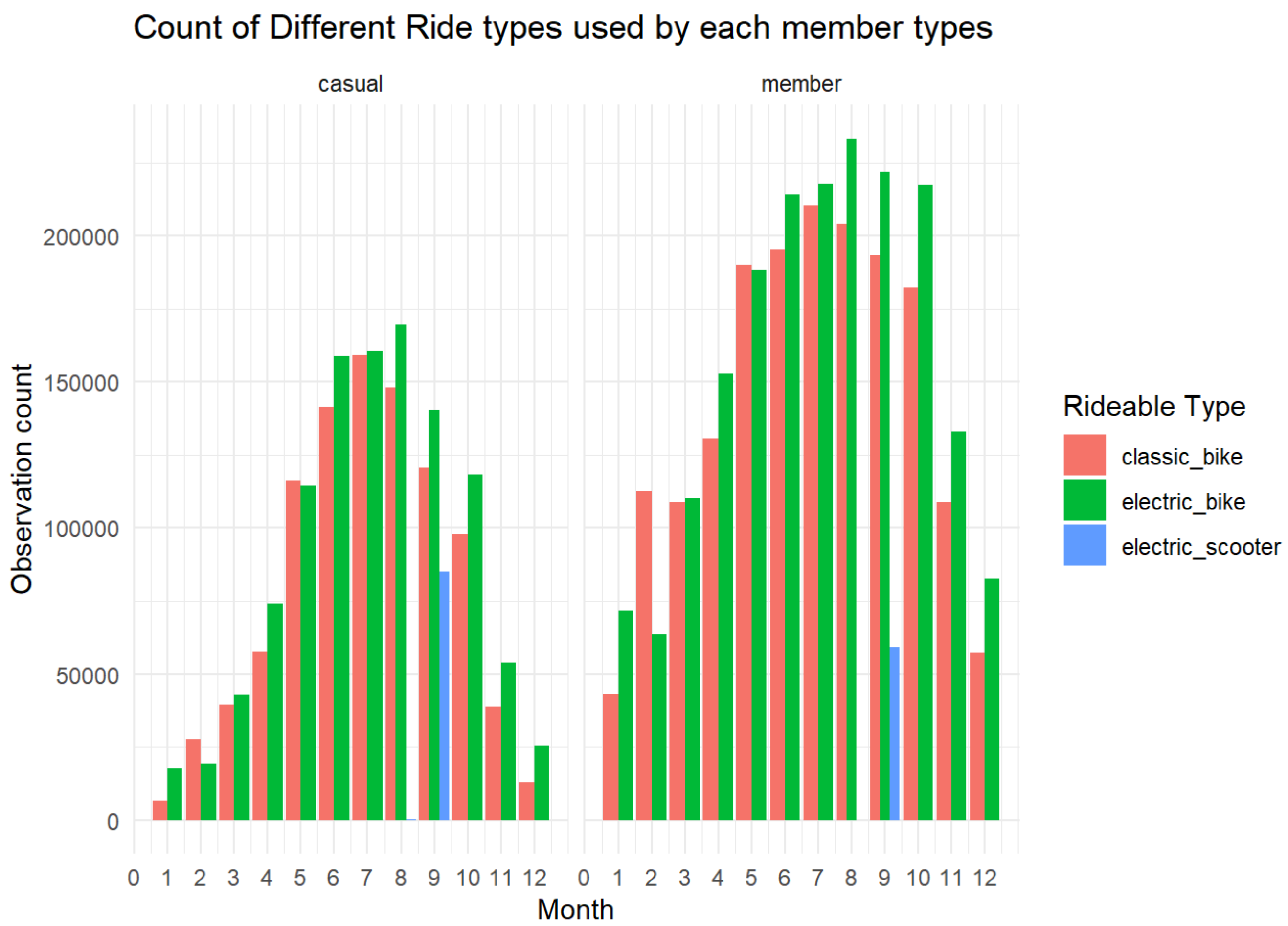


Interesting thing to note is casual members used more electric scooters than annual members even though they are proportionally less.

Next we will see how many rides were done using which ride type and which membership type.

```
member_month_rideable_count <- combined_data %>%
  group_by(member_casual, month_started_at, rideable_type) %>%
  summarize(observation_count = n())
```

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



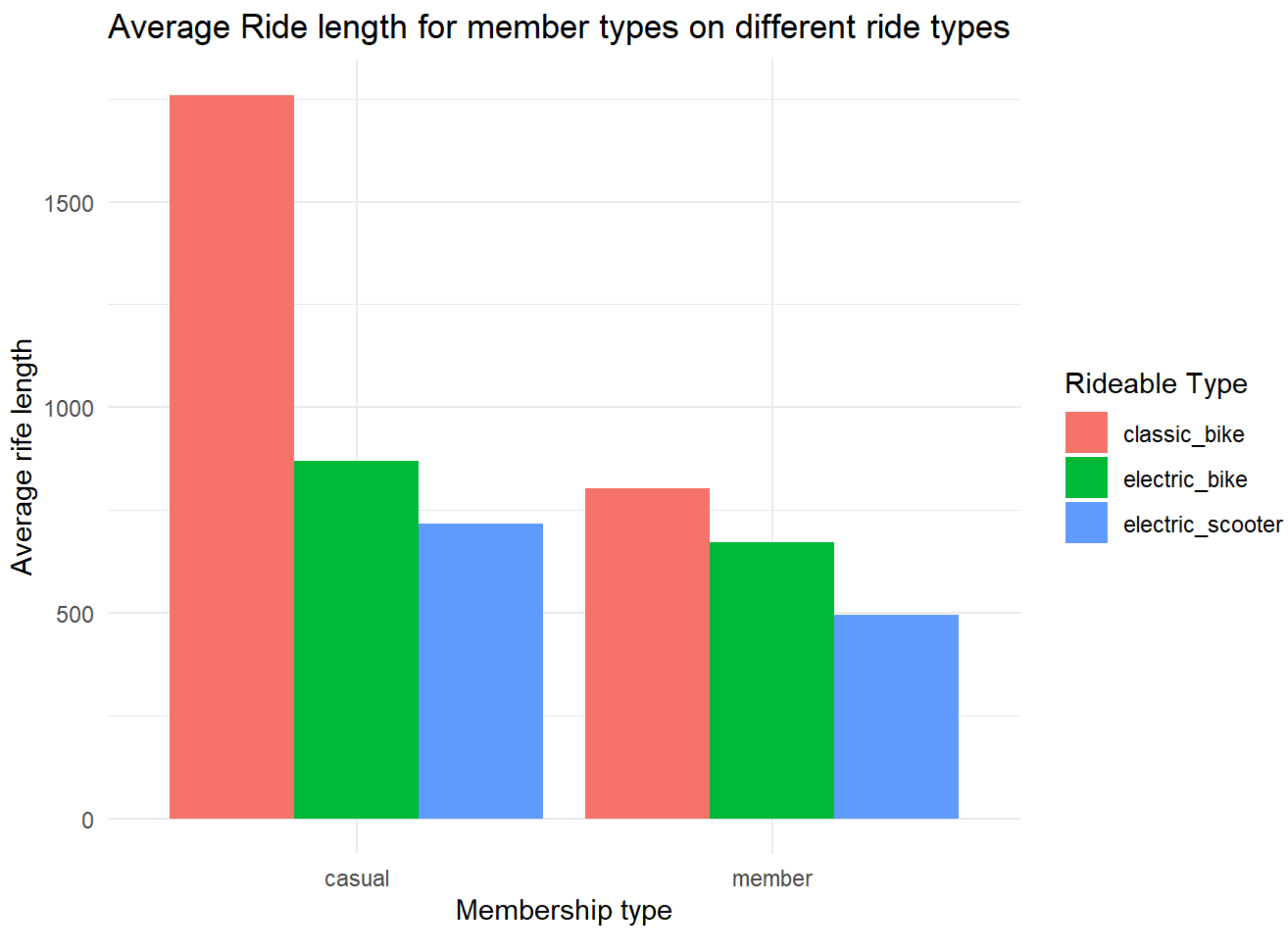
Casual users use the cyclist service very less in winter season, both users follow a uniform pattern where they use the service more in summer as compared to winter.

Next we want to see the average ride length for membership types on different ride types.

```
member_ridetype_avg <- combined_data %>% group_by(member_casual, rideable_type) %>%
  summarize(avg_ride_length = mean(ride_length))
```

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

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



It is notable that although casual members have more average ride lengths than annual users but when it comes to riding classic bikes the average ride lengths is significant and should be looked at further.

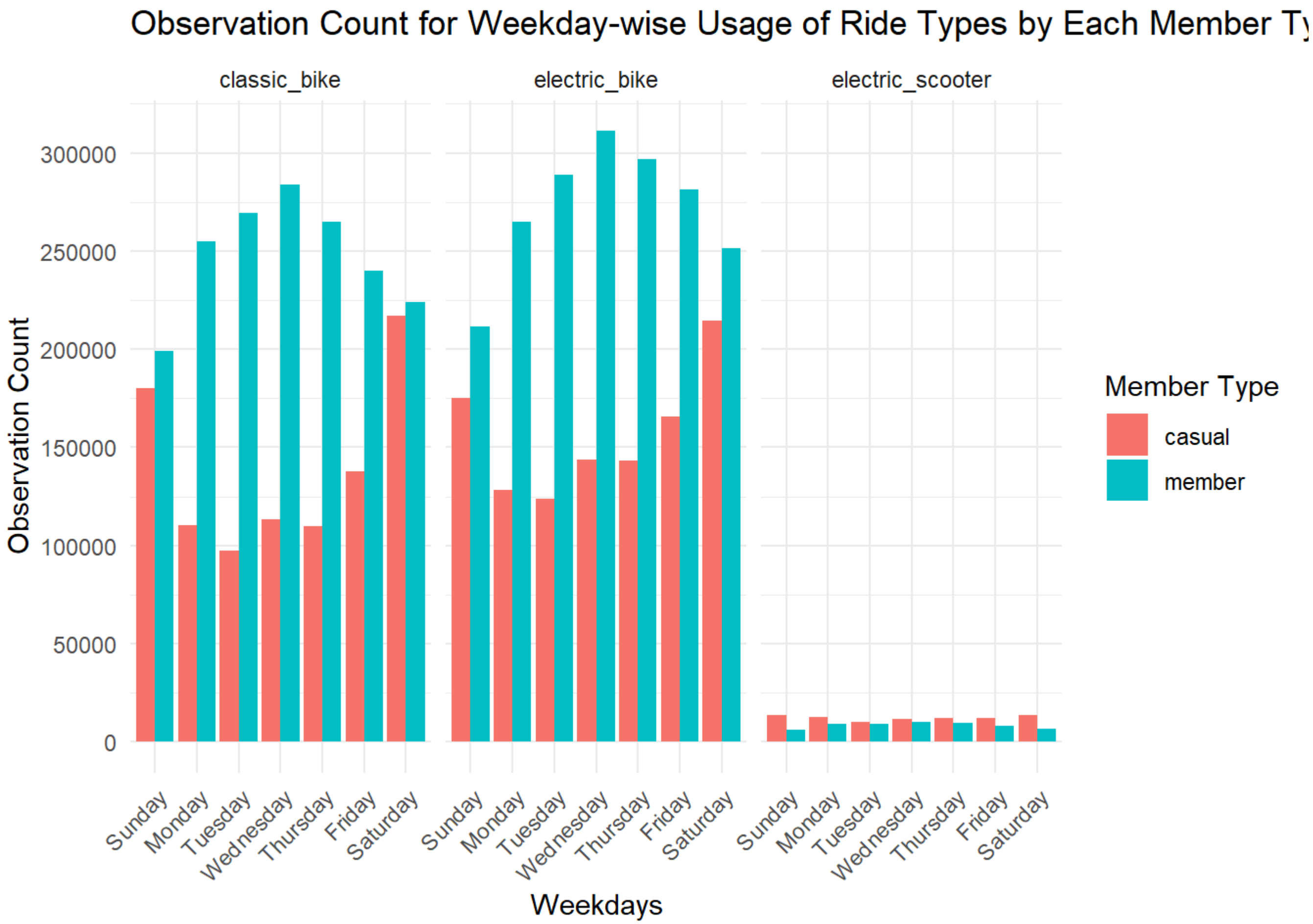
Let us see how membership type and weekday effects the ride type used by people.

```
member_weekday_eidetype_count <- combined_data %>% group_by(member_casual, weekday_started_at, rideable_type) %>%
  summarize(observation_count = n())
```

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

```
weekday_order <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

# Convert weekday_started_at to a factor with the correct order
member_weekday_eidetype_count <- member_weekday_eidetype_count %>%
  mutate(weekday_started_at = factor(weekday_started_at, levels = weekday_order))
```

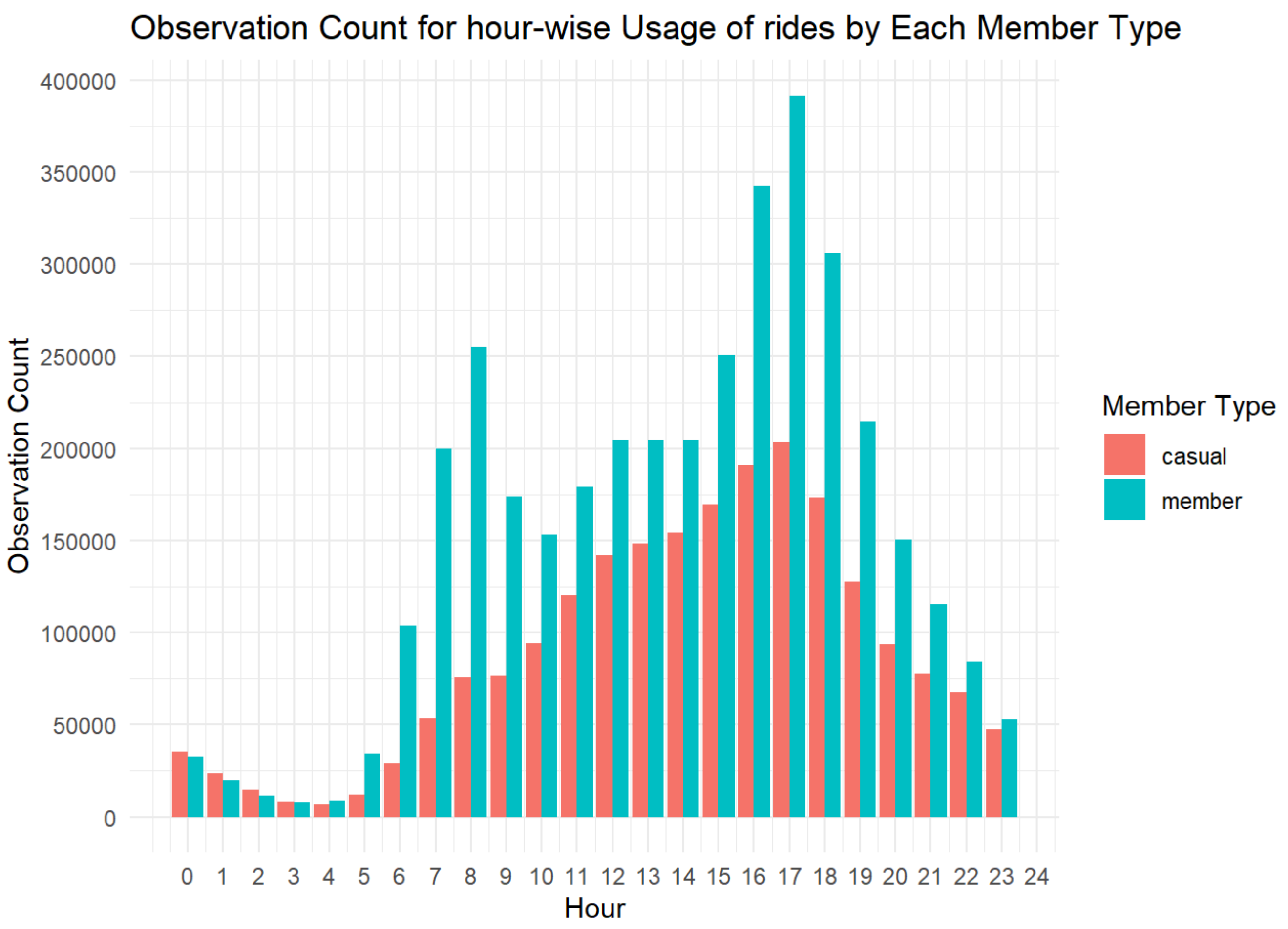


Both major types of ride follow the same pattern with both member types.

Let us now see when the different membership types have used the service in the day.

```
hour_member_count <- combined_data %>% group_by(member_casual, hour_started_at) %>%
  summarize(observation_count = n())
```

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

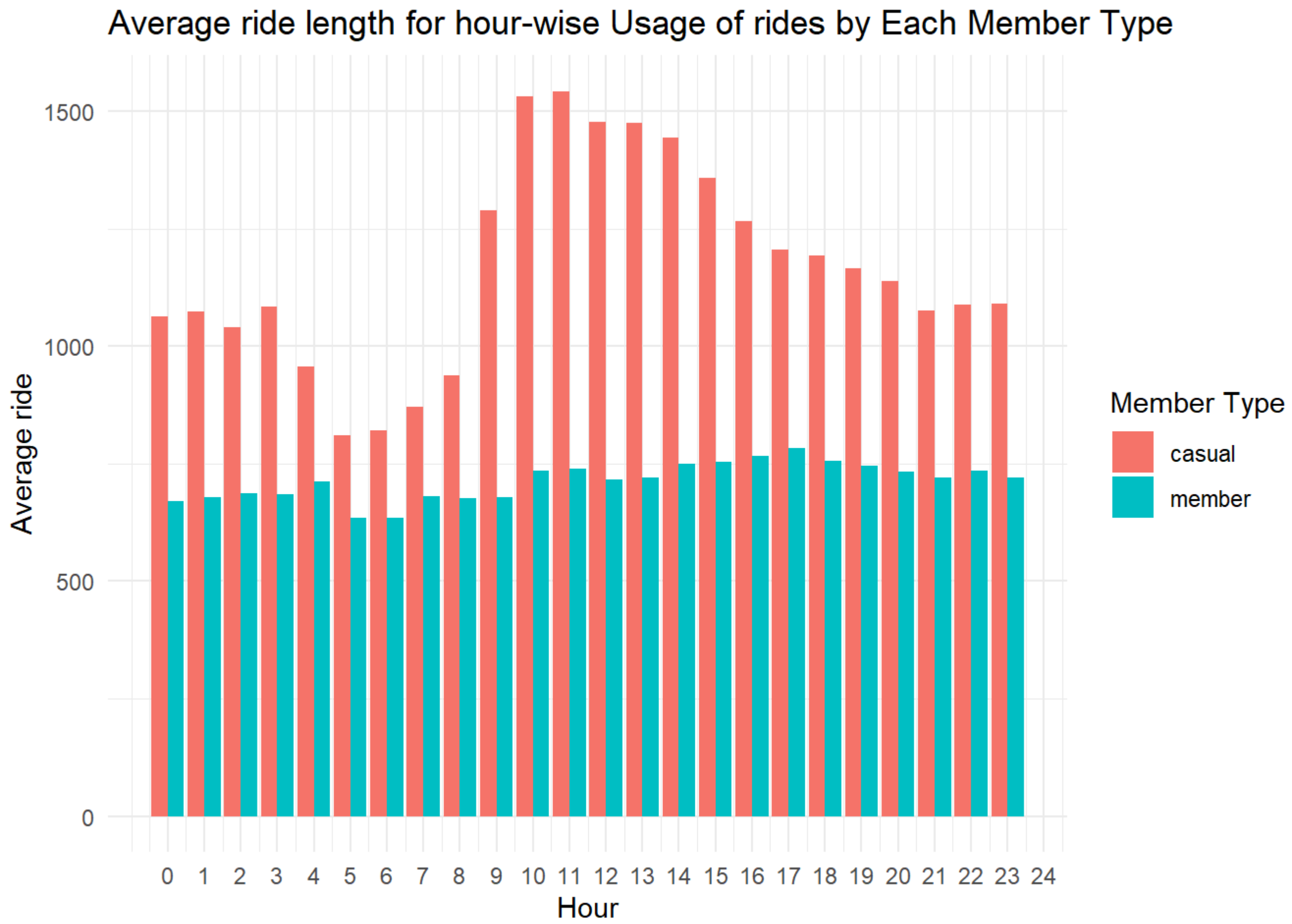


We can deduce from this chart that a huge proportion of annual members use cyclists for transport between work, college or school as it has spikes in morning and evening for casual users it follows a uniform pattern with spike in midday which indicates maybe it is not used for commute but leisure on weekends.

Next we will look at how the time of the day and membership type effects the ride length.

```
hour_member_avg <- combined_data %>% group_by(member_casual, hour_started_at) %>%
  summarize(avg_ride_length = mean(ride_length))
```

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



We can see that average ride length remains relatively constant for annual members for casual members though the ride length varies and forms a pattern where between 10 AM and 1 PM the ride lengths are highest and form a decreasing slope from there on.

Now we will see the top routes which are taken by both types of members.

```
top_routes <- combined_data %>%
  group_by(member_casual, journey) %>%
  summarize(count = n()) %>%
  arrange(member_casual, desc(count)) %>%
  group_by(member_casual) %>%
  slice_head(n = 10)
```

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

```
top_routes
```

```
## # A tibble: 20 × 3
## # Groups:   member_casual [2]
##    member_casual journey                                                   count
##    <chr>         <chr>                                                     <int>
##  1 casual        _                                                        235607
##  2 casual        Streeter Dr & Grand Ave_Streeter Dr & Grand Ave            8864
##  3 casual        DuSable Lake Shore Dr & Monroe St_DuSable Lake Shore Dr…   7246
##  4 casual        DuSable Lake Shore Dr & Monroe St_Streeter Dr & Grand A…   5274
##  5 casual        Michigan Ave & Oak St_Michigan Ave & Oak St                4622
##  6 casual        Millennium Park_Millennium Park                            3361
##  7 casual        Dusable Harbor_Dusable Harbor                              3119
##  8 casual        Streeter Dr & Grand Ave_DuSable Lake Shore Dr & Monroe …   2702
##  9 casual        Streeter Dr & Grand Ave_                                   2590
## 10 casual        Shedd Aquarium_Streeter Dr & Grand Ave                     2399
## 11 member        _                                                        291365
## 12 member        State St & 33rd St_Calumet Ave & 33rd St                   5490
## 13 member        Calumet Ave & 33rd St_State St & 33rd St                   5448
## 14 member        Ellis Ave & 60th St_Ellis Ave & 55th St                    4008
## 15 member        Ellis Ave & 60th St_University Ave & 57th St               3880
## 16 member        University Ave & 57th St_Ellis Ave & 60th St               3879
## 17 member        Ellis Ave & 55th St_Ellis Ave & 60th St                    3793
## 18 member        _Clinton St & Washington Blvd                              2911
## 19 member        _Kingsbury St & Kinzie St                                  2847
## 20 member        Kingsbury St & Kinzie St_                                  2822
```

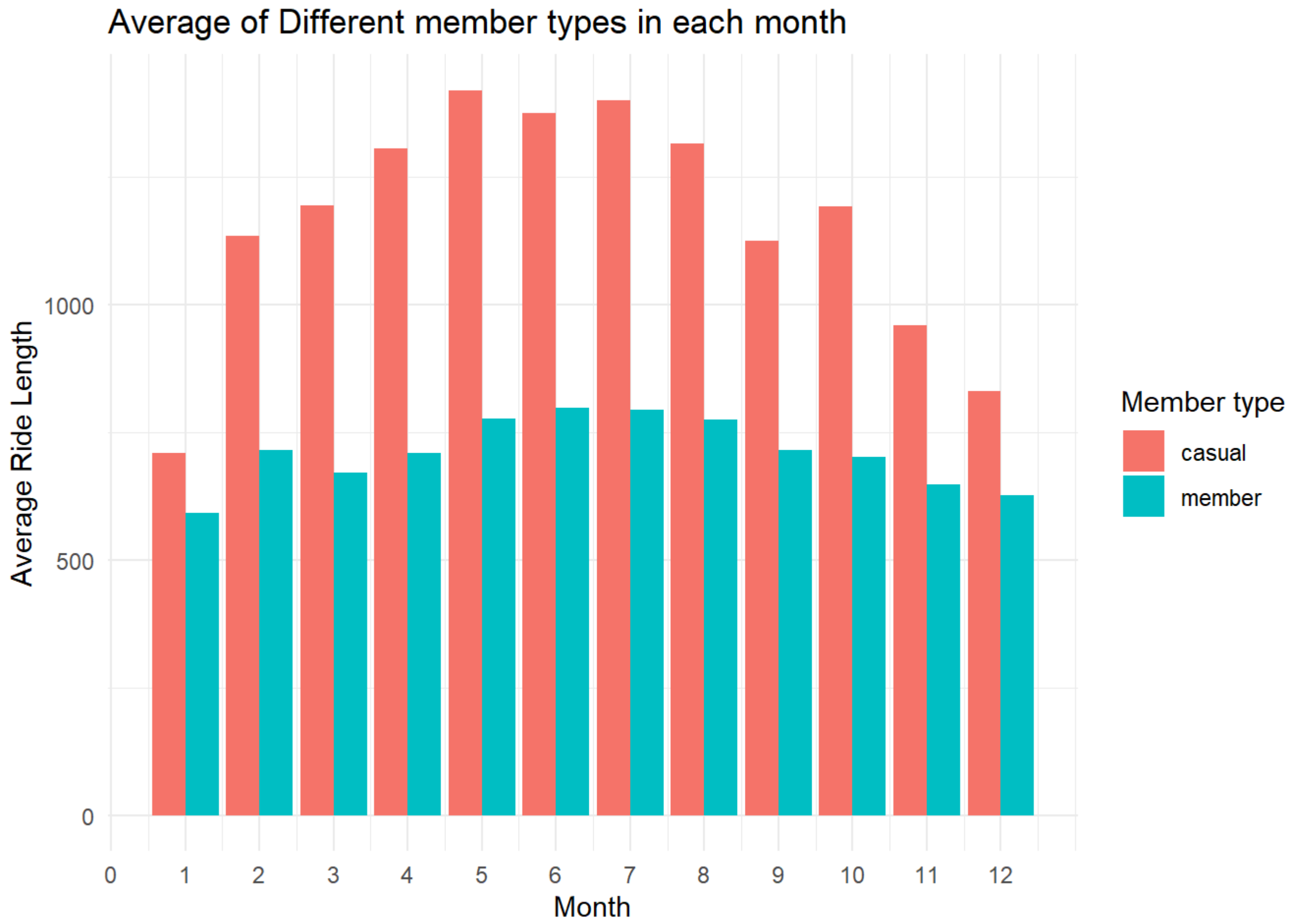
Interestingly in the top 10 routes there are no common routes between Casual and annual members

Let us now see the monthly average ride length for both member types.

```
member_month_rideable_avg <- combined_data %>%
  group_by(member_casual, month_started_at) %>%
  summarize(avg_ride_length = mean(ride_length))
```

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

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



While there is very small changes in ride length of annual members there is a huge difference in Casual members, the pattern is that they ride for longer in summer season the Ride length is highest in the month of May for casual members.

lets see if day of the week has any effects on the ride length of member types.

```
member_weekday_ride_avg <- combined_data %>%
  group_by(member_casual, weekday_started_at) %>%
  summarize(avg_ride_length = mean(ride_length))
```

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

```
weekday_order <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

# Convert weekday_started_at to a factor with the correct order
member_weekday_ride_avg <- member_weekday_ride_avg %>%
  mutate(weekday_started_at = factor(weekday_started_at, levels = weekday_order))
```

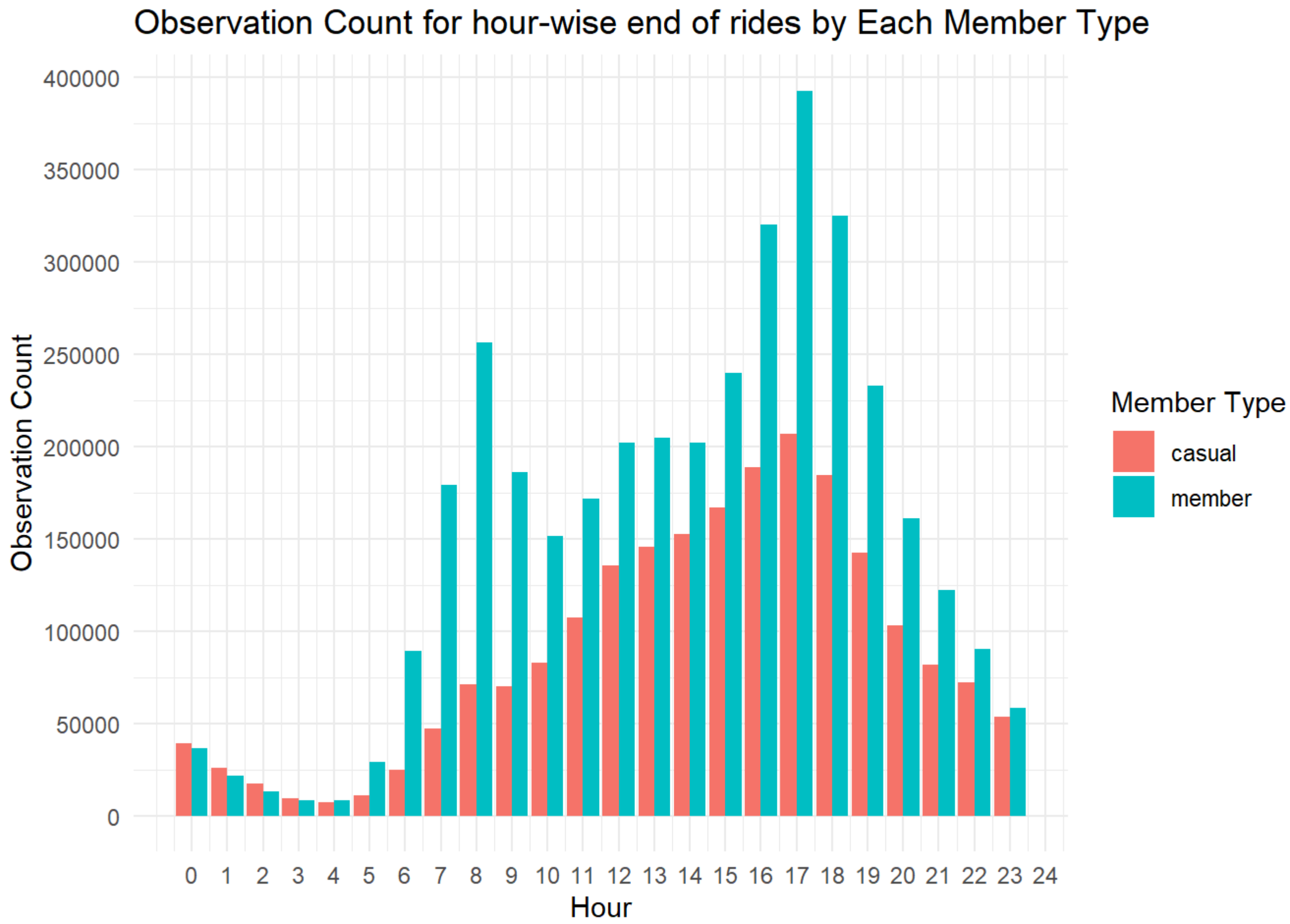
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

As we can see the Ride Length is highest on weekends for casual members and for Annual members it is constant throughout the week.

for the last plot let us see how does the ride ending time effect the number of users in both member types.

```
hour_member_count_end <- combined_data %>% group_by(member_casual, hour_ended_at) %>%
  summarize(observation_count = n())
```

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



It follows the same pattern as the chart we saw before on hours started at.

Conclusion

The analysis of Cyclistic's data reveals distinct usage patterns between annual members and casual users. Annual members primarily use the service on weekdays for commuting, showing consistent usage throughout the year. In contrast, casual users prefer weekends and exhibit higher activity in summer, with a significant decrease in winter.

Electric bikes are slightly more favored by both groups, while ride durations are notably longer for casual users, especially with classic bikes. This suggests leisure-oriented usage for casual users, compared to the time-bound rides of annual members.

Understanding these patterns can help Cyclistic tailor its services and marketing strategies. By addressing the specific needs of each user segment, Cyclistic can enhance user satisfaction and optimize operations, ensuring it remains a competitive urban mobility solution.