Cyclistic bike-share analysis case study - How does a bike-share navigate speedy success?

Google Analytics_Capstone 1

Author: Glicerio Vergara

Date: 2023-08-01

Background for this activity

This is the solution for the Google data case study number 1. The project implements the data analysis process: ask, prepare, process, analyze, share, and act. Below is the background information required to understand the project and data.

A startup called Cyclistic rents out bicycles and offers both casual riding alternatives and yearly subscription programs. The future success of Cyclistic, in the opinion of the company's marketing director, depends on increasing the number of annual subscriptions. The marketing team wants to obtain a thorough understanding of how annual members and casual riders use Cyclistic bikes differently in order to accomplish this goal. The team plans to create a customized marketing plan to turn infrequent riders into devoted annual members by learning more about their unique usage patterns and preferences.

The outcome of the project will be a detailed report and data-driven recommendations for the marketing strategy.These recommendations will include answers that explain: - How annual members and casual riders differ - Why casual riders would buy a membership - How digital media could affect their marketing tactics

The data has been made available by Motivate International Inc. under a license. All the file has been downloaded as `cvs` and will be imported into `R` later on.

The data source is available via the following link: - https://divvy-tripdata.s3.amazonaws.com/index.html

The projects takes the data from 2020/04 to 2023/05.

Step 1: Packages loading

In order to set up the environment for starting the process, loading pre-existing packages that will help to prepare, process and analyze the data need to be done in `R`.

The following are the data packages used for developing the analysis.

If the packages are not already installed, use `install.packages()` with the name of the package in "" as shown e.g. install.packages("dplyr").

Step 2: Data preparation

As the data is separated in multiple `cvs` documents, there is a need to combine the multiple files in order to create a master data set. In this case the combined set will be named `Combine_df`.

The following code unites all the `cvs` files in an specific folder:

```
# Set the path to the folder containing CSV files
folder_path <- "Data/Trip data"

# Get a list of all CSV files in the folder
csv_files <- list.files(path = folder_path, pattern = "\\.csv$", full.names = TRUE)

# Read and combine all CSV files into a single data frame
Combined_df <- do.call(rbind, lapply(csv_files, read.csv))
```

Step 2: Data processing

After all the documents are combined into the master data, There are some new columns that need to be created in order to prepare for the analysis.

The new columns are as follows: - Ride_length: Ride time in seconds - Day_of_week: Day of the week for the ride

```
# Add new column from existing column that represent the time of used per ride
Combined_df$Ride_length <- difftime( Combined_df$ended_at, Combined_df$started_at, units='secs')

# Add new column from existing column that represent the day of the weeks
Combined_df$Day_of_week <- wday( Combined_df$ended_at, label = TRUE)

# Filter dataset so that the Ride_length column is positive
Combined_df <- Combined_df %>% filter ( Ride_length > 0)
```

Step 2: Descriptive data analysis

After the data is stored appropriately and has been prepared for analysis, it is possible to start answering the initial questions of the project.

The following is a snapshot of the final data set.

```
# Display the first row of the data set
head(Combined_df)%>% as.data.frame()
```

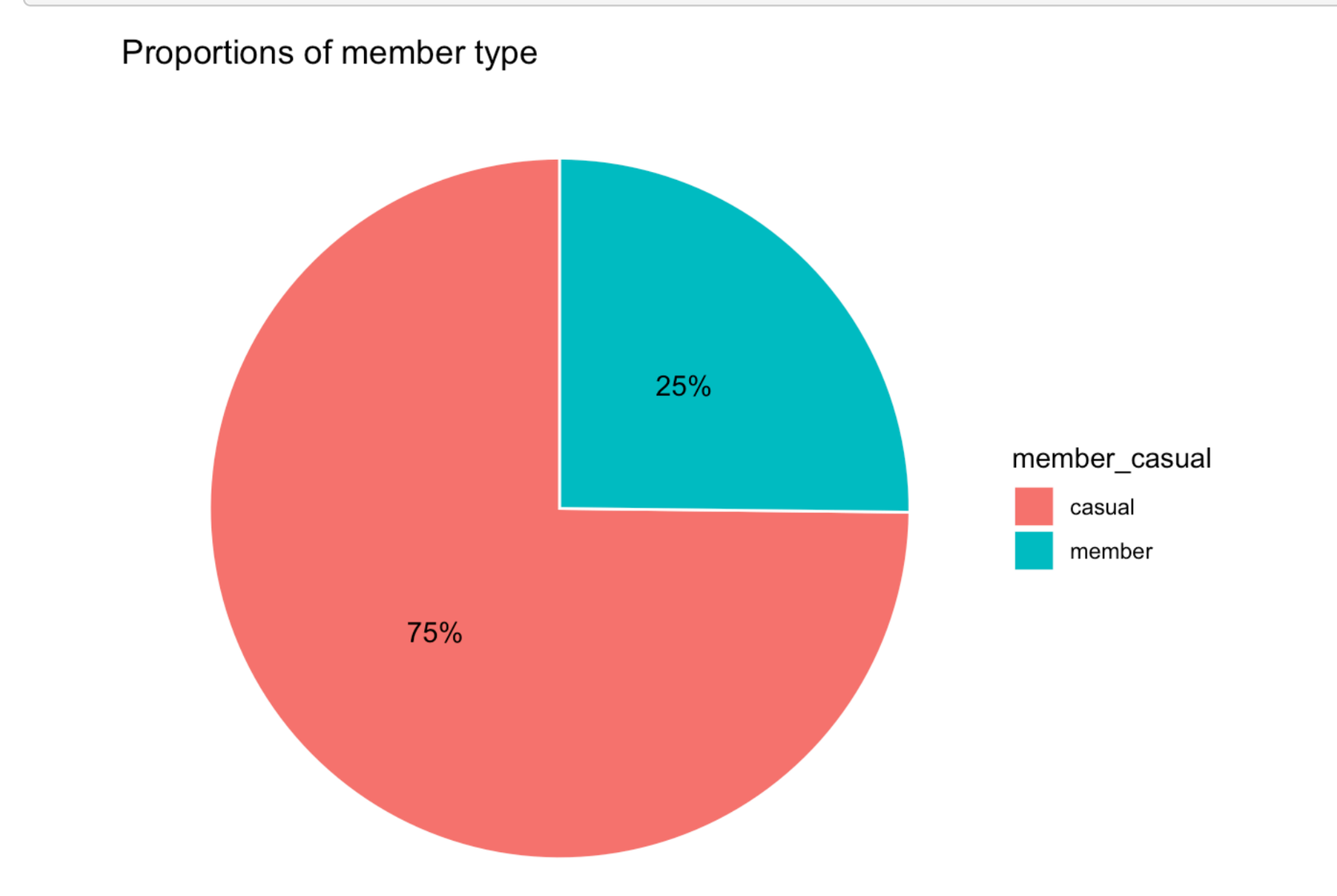
```
##            ride_id rideable_type          started_at            ended_at
## 1 D13A74BDC5BECD4D   docked_bike 2020-04-26 15:52:04 2020-04-27 13:02:59
## 2 0027472D3F011D6C   docked_bike 2020-04-18 14:57:36 2020-04-19 13:23:10
## 3 9A6511C5E73D0037   docked_bike 2020-04-19 23:48:13 2020-04-20 00:08:00
## 4 DBE13117234A5743   docked_bike 2020-04-12 17:02:52 2020-04-18 14:40:13
## 5 D1FC3C4D239F430A   docked_bike 2020-04-20 23:57:32 2020-04-21 00:02:54
## 6 9EC5672E419457FB   docked_bike 2020-04-06 14:07:57 2020-04-11 10:34:52
##                               start_station_name start_station_id
## 1                      Blue Island Ave & 18th St              129
## 2                         Damen Ave & Pierce Ave               69
## 3 Burling St (Halsted) & Diversey Pkwy (Temp)               332
## 4                     Lake Shore Dr & North Blvd              268
## 5                    Cottage Grove Ave & 67th St              429
## 6                   Archer (Damen) Ave & 37th St              645
##                                 end_station_name end_station_id start_lat
## 1                      Blue Island Ave & 18th St            129   41.8576
## 2                       Lincoln Ave & Addison St            330   41.9094
## 3 Burling St (Halsted) & Diversey Pkwy (Temp)             332   41.9331
## 4                    Winthrop Ave & Lawrence Ave            253   41.9117
## 5                            MLK Jr Dr & 63rd St            430   41.7737
## 6                   Archer (Damen) Ave & 37th St            645   41.8267
##   start_lng end_lat  end_lng member_casual Ride_length Day_of_week
## 1  -87.6615 41.8576 -87.6615        casual  86400 secs         Mon
## 2  -87.6777 41.9462 -87.6733        member  86400 secs         Sun
## 3  -87.6478 41.9331 -87.6478        casual  86400 secs         Mon
## 4  -87.6268 41.9688 -87.6577        casual 518400 secs         Sat
## 5  -87.6056 41.7801 -87.6159        member  86400 secs         Tue
## 6  -87.6831 41.8267 -87.6831        casual 432000 secs         Sat
```

To understand the differences between the memberships, it is important to understand their split. The split will highlight which of the groups brings the highest revenue to the company.

```
# Table of percentage per costumer type
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  group_by(member_casual) %>%
  summarise(count_members = n()) %>%
  mutate(percent_member = (count_members / sum(count_members) * 100))%>%
  as.data.frame()
```

```
##   member_casual count_members percent_member
## 1        casual         83872       74.83226
## 2        member         28208       25.16774
```

```
# Pie chart of the results
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  group_by(member_casual) %>%
  summarise(count = n()) %>%
  mutate(percent = count / sum(count) * 100) %>%
  ggplot(aes(x = "", y = count, fill = member_casual)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  geom_text(aes(label = paste0(round(percent), "%")), position = position_stack(vjust = 0.5)) +
  labs(title = "Proportions of member type") +
  coord_polar("y", start = 0) +
  theme_void()
```



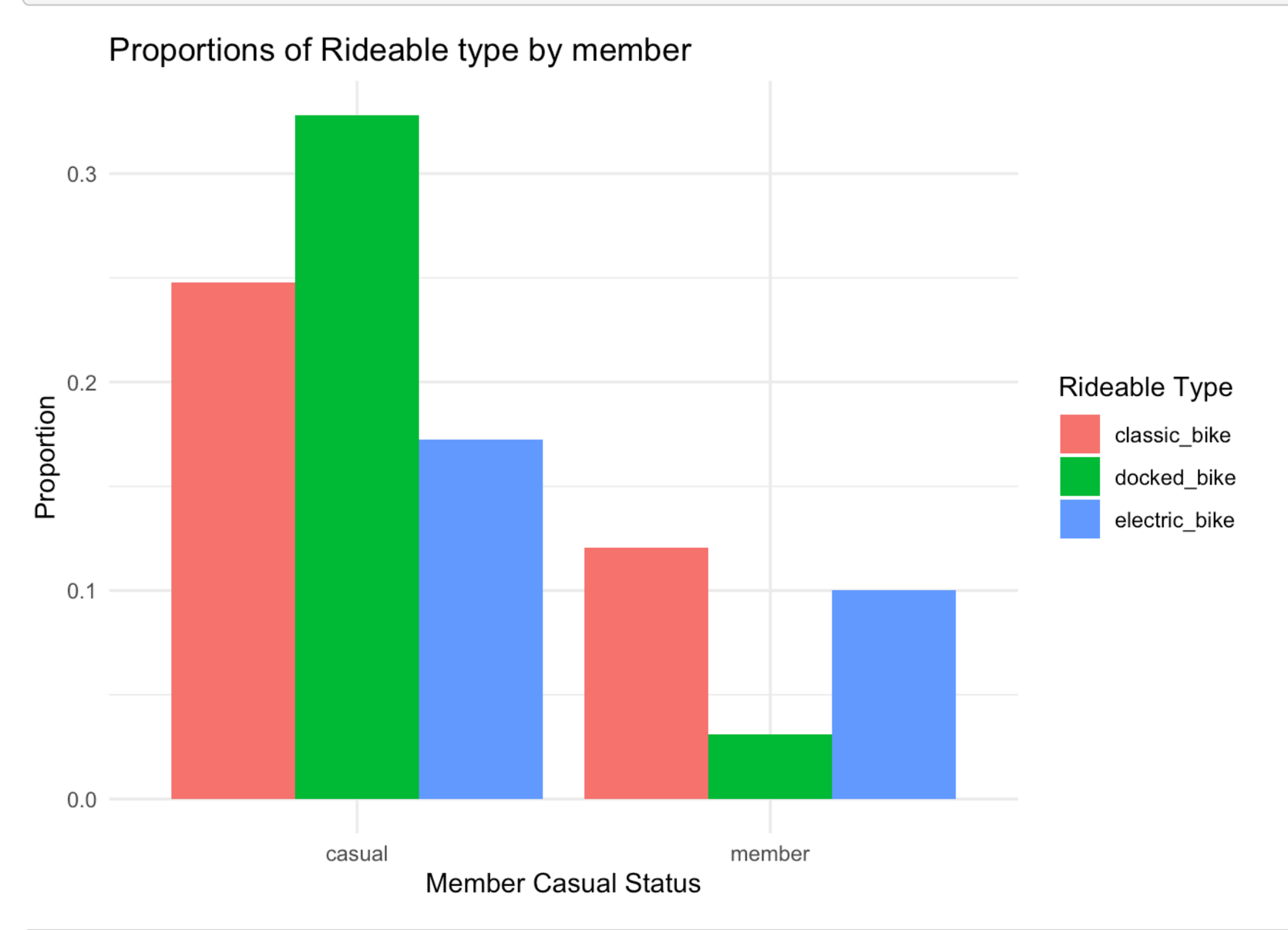

```
# Bar chart of the results
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  group_by(member_casual) %>%
  summarise(count_members = n()) %>%
  ggplot(aes(x = member_casual, y = count_members, fill = member_casual)) +
  geom_col(width = 0.5) +
  labs(title = "Number of members by type",
       x = "Member type",
       y = "Number of members") +
  theme_minimal()
```

After knowing the percentage per type of costumer, it is important to understand which type of ride they used most of the time. To do that, it is required to display the split per type of member and type of ride used.

```
# Proportion table of the members and type of ride
prop.table(table(Combined_df$member_casual, Combined_df$rideable_type))%>%
  as.data.frame()
```

```
##     Var1          Var2       Freq
## 1 casual  classic_bike 0.24788544
## 2 member  classic_bike 0.12043183
## 3 casual   docked_bike 0.32805139
## 4 member   docked_bike 0.03100464
## 5 casual electric_bike 0.17238580
## 6 member electric_bike 0.10024090
```

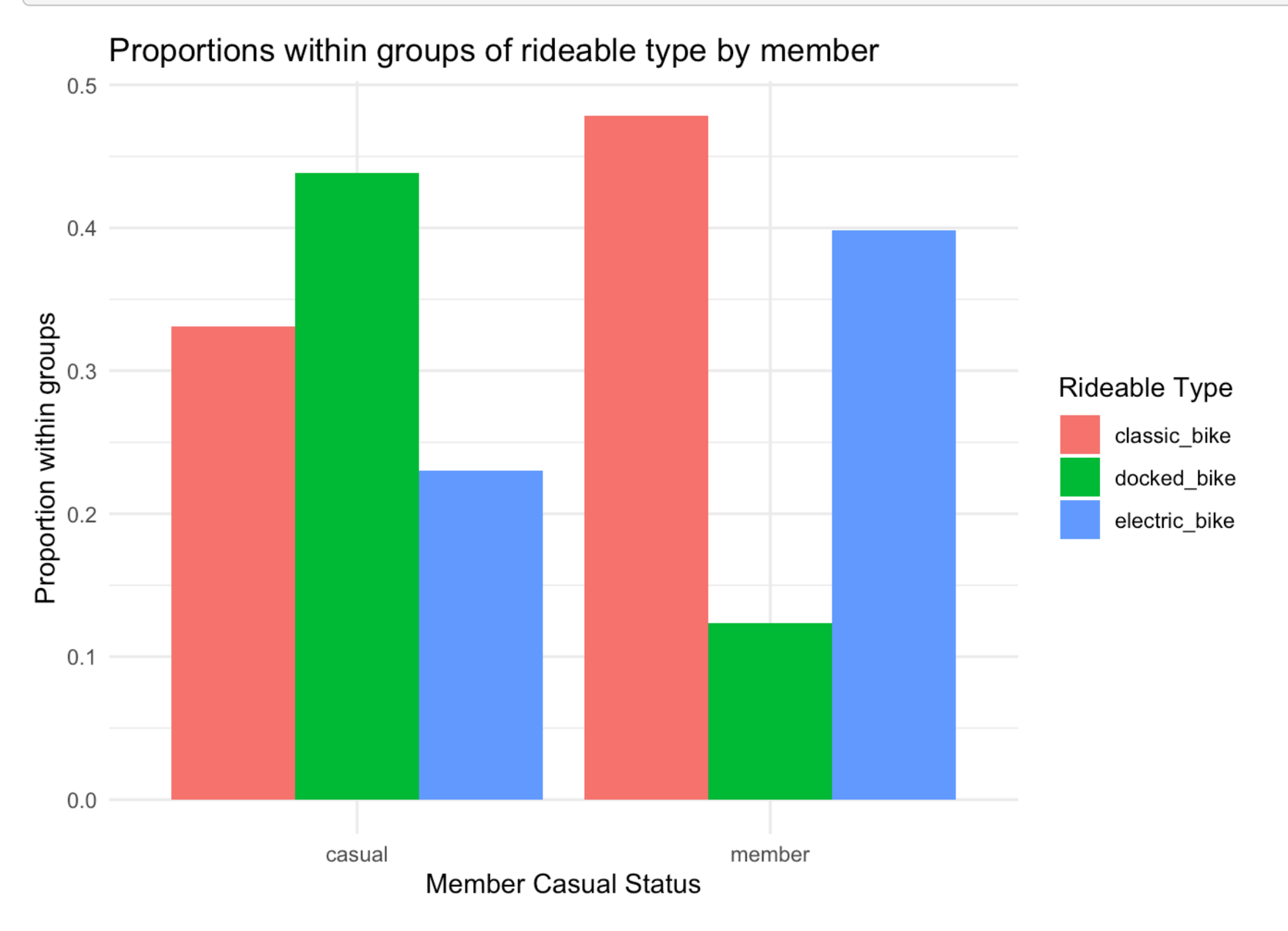
```
# Bar chart of the split per member and type of ride
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  count(member_casual, rideable_type) %>%
  mutate(proportion = n / sum(n)) %>%
  group_by(member_casual) %>%
  ggplot(aes(x = member_casual, y = proportion, fill = rideable_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportions of Rideable type by member",
       x = "Member Casual Status",
       y = "Proportion") +
  scale_fill_discrete(name = "Rideable Type") +
  theme_minimal()
```




```
# Proportion table within member groups and type of ride
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  count(member_casual, rideable_type) %>%
  group_by(member_casual) %>%
  mutate(proportion = n / sum(n))
```

```
## # A tibble: 6 × 4
## # Groups:   member_casual [2]
##   member_casual rideable_type     n proportion
##   <fct>         <chr>         <int>      <dbl>
## 1 casual        classic_bike  27783      0.331
## 2 casual        docked_bike   36768      0.438
## 3 casual        electric_bike 19321      0.230
## 4 member        classic_bike  13498      0.479
## 5 member        docked_bike    3475      0.123
## 6 member        electric_bike 11235      0.398
```

```
# Bar chart of proportion table within member groups and type of ride
Combined_df %>%
  mutate(member_casual = as.factor(member_casual)) %>%
  count(member_casual, rideable_type) %>%
  group_by(member_casual) %>%
  mutate(proportion = n / sum(n)) %>%
  ggplot(aes(x = member_casual, y = proportion, fill = rideable_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportions within groups of rideable type by member",
       x = "Member Casual Status",
       y = "Proportion within groups") +
  scale_fill_discrete(name = "Rideable Type") +
  theme_minimal()
```



To give a sense of the average ride length and the maximum ride time per costumer, the split per member type and rideable type need to be displayed. This result will give more granularity to the type of ride preferred per costumer type. The highest mean and media will show which group has the highest average ride length per type of vehicle. The highest ride length time will give an understanding on which group of costumers used the ride the most.

```
Combined_df %>%
  group_by( member_casual, rideable_type) %>%
  summarise(mean_ride_length = mean(Ride_length),
            media_ride_length = median(Ride_length),
            max_ride_length = max(Ride_length))%>%
  as.data.frame()
```

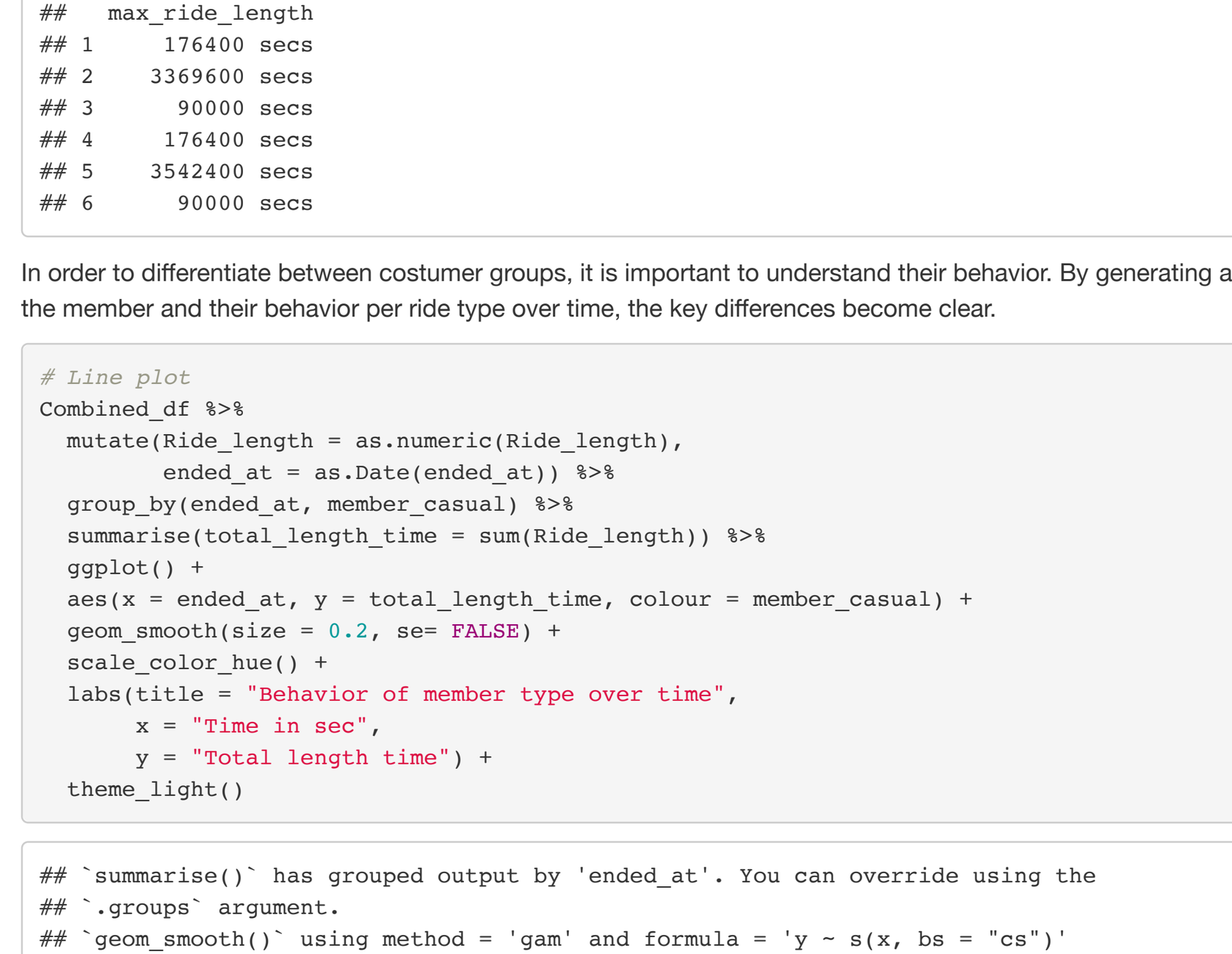
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
##   member_casual rideable_type mean_ride_length media_ride_length
## 1        casual  classic_bike    87091.29 secs        86400 secs
## 2        casual   docked_bike   133568.54 secs        86400 secs
## 3        casual electric_bike    86403.35 secs        86400 secs
## 4        member  classic_bike    86765.92 secs        86400 secs
## 5        member   docked_bike    95949.58 secs        86400 secs
## 6        member electric_bike    86403.20 secs        86400 secs
##   max_ride_length
## 1     176400 secs
## 2    3369600 secs
## 3      90000 secs
## 4     176400 secs
## 5    3542400 secs
## 6      90000 secs
```

In order to differentiate between costumer groups, it is important to understand their behavior. By generating a plot that depicts the behavior of the member and their behavior per ride type over time, the key differences become clear.

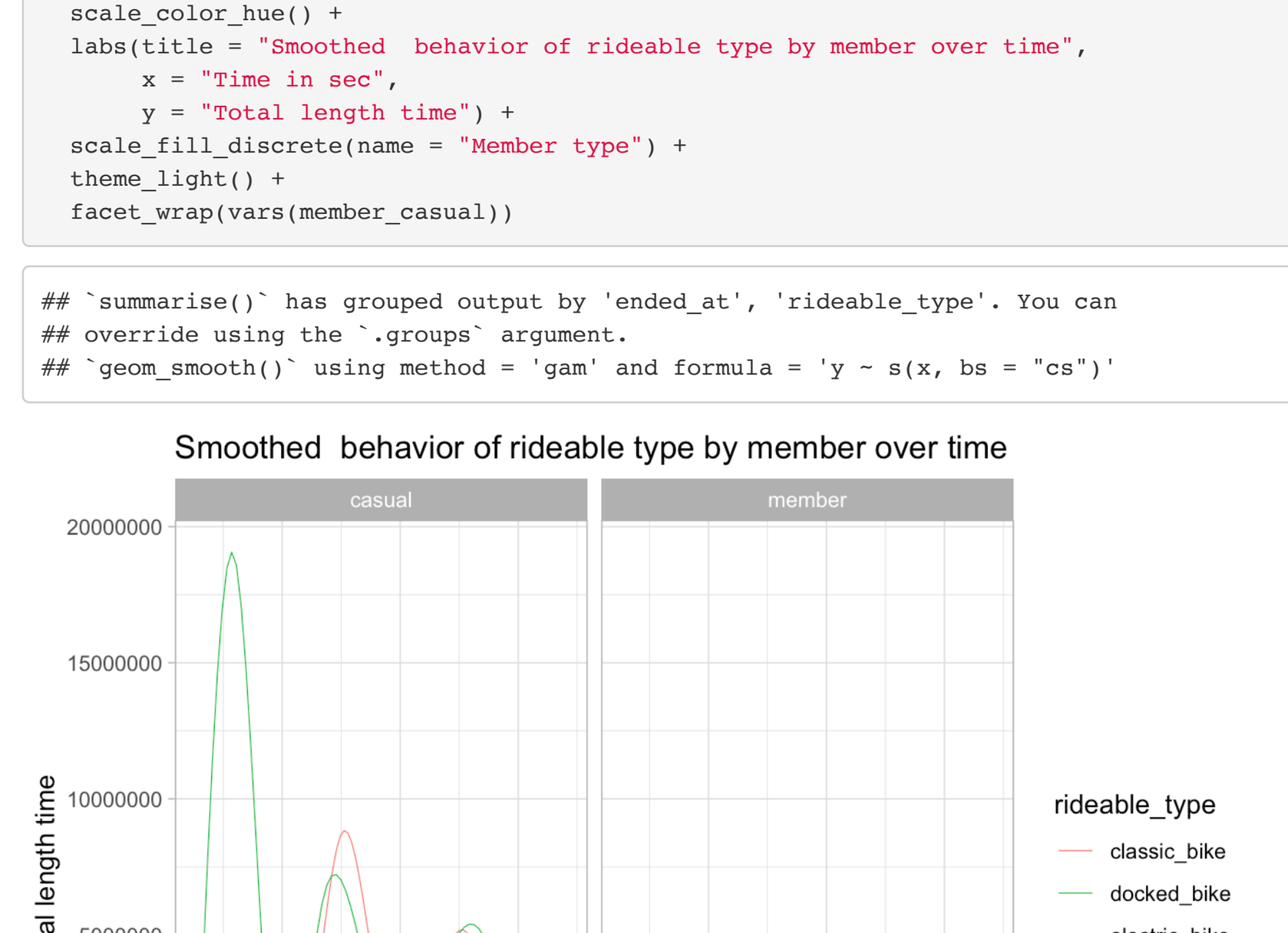
```
# Line plot
Combined_df %>%
  mutate(Ride_length = as.numeric(Ride_length),
         ended_at = as.Date(ended_at)) %>%
  group_by(ended_at, member_casual) %>%
  summarise(total_length_time = sum(Ride_length)) %>%
  ggplot() +
  aes(x = ended_at, y = total_length_time, colour = member_casual) +
  geom_smooth(size = 0.2, se= FALSE) +
  scale_color_hue() +
  labs(title = "Behavior of member type over time",
       x = "Time in sec",
       y = "Total length time") +
  theme_light()
```

```
## `summarise()` has grouped output by 'ended_at'. You can override using the
## `.groups` argument.
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



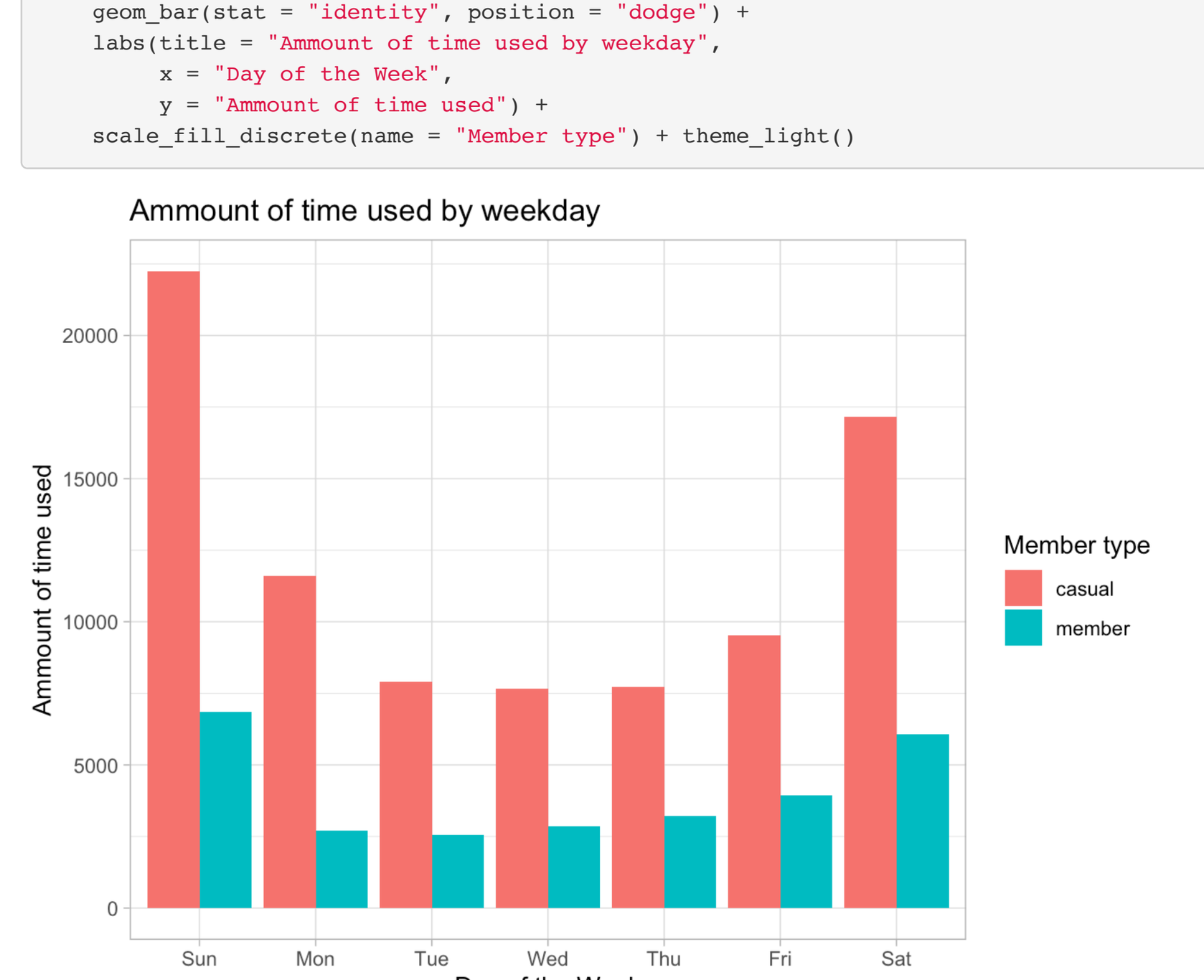

```
# Line plot
Combined_df %>%
  mutate(Ride_length = as.numeric(Ride_length),
         ended_at = as.Date(ended_at)) %>%
  group_by(ended_at, rideable_type, member_casual) %>%
  summarise(total_length_time = sum(Ride_length)) %>%
  ggplot() +
  aes(x = ended_at, y = total_length_time, colour = rideable_type) +
  geom_smooth(size = 0.2, se= FALSE) +
  scale_color_hue() +
  labs(title = "Smoothed  behavior of rideable type by member over time",
       x = "Time in sec",
       y = "Total length time") +
  scale_fill_discrete(name = "Member type") +
  theme_light() +
  facet_wrap(vars(member_casual))
```

```
## `summarise()` has grouped output by 'ended_at', 'rideable_type'. You can
## override using the `.groups` argument.
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

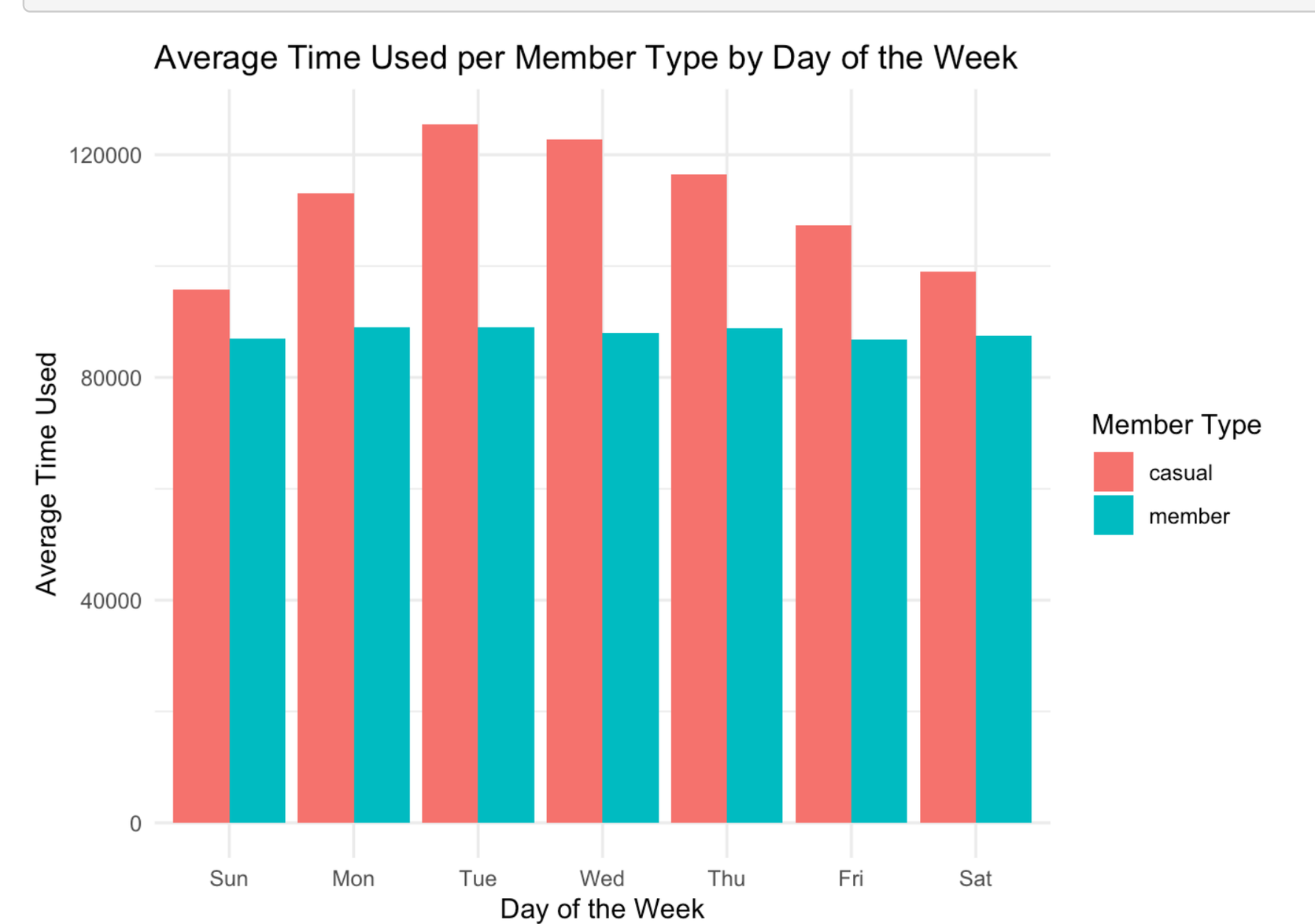


By doing a more granular dive in order to understand the behavior of the users per day of the week, it is important to depict the amount of time the user used the ride and their average amounts of usage per day. These two behaviors will further expand the differentiation between the two types of members.

```
Combined_df %>%
    mutate(member_casual = as.factor(member_casual),
           Day_of_week = as.factor(Day_of_week)) %>%
    count(member_casual, Day_of_week) %>%
    group_by(member_casual, Day_of_week) %>%
    mutate(proportion = sum(n)) %>%
    ggplot(aes(x = Day_of_week, y = proportion, fill = member_casual)) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = "Ammount of time used by weekday",
         x = "Day of the Week",
         y = "Ammount of time used") +
    scale_fill_discrete(name = "Member type") + theme_light()
```




```
Combined_df %>%
  group_by(member_casual, Day_of_week) %>%
  summarise(Ride_length_mean = mean(as.numeric(Ride_length)), .groups = 'drop') %>%
  ggplot(aes(x = Day_of_week, y = Ride_length_mean, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Time Used per Member Type by Day of the Week",
       x = "Day of the Week",
       y = "Average Time Used") +
  scale_fill_discrete(name = "Member Type") +
  theme_minimal()
```



The analysis of the data allowed us to better understand the nature of the people who used the ride. After developing a process from beginning to end, a clear solution of the initial questions is provided. The analysis gave answers to the key differences in members and why according to their behavior the casual user should buy a membership.