Bike share Case study

Andres Esequiel Tomatis

2022-06-13

Ask

Business Task

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 company has two types of customers, casual riders and annual members, being the last ones much more profitables. So, the company's future success depends on maximizing the number of annual memberships. The Task: understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics, analyzing the Cyclistic historical bike trip data to identify trends. From all of these insights, the marketing team will design a new strategy to convert casual riders into annual members.

Stakeholders

- The director of marketing (my manager).
- Cyclistic marketing analytics team.
- Cyclistic executive team.

Prepare

Data sources

I will use Cyclistic's historical trip data from the previous tuelve months (Jun, 2021 - May, 2022) to analyze and identify trends. Click here to download the data.
The datasets have a different name because Cyclistic is a fictional company. For the purposes of this case study, the datasets are appropriate and will allow business questions to be answered. The data has been made available by Motivate International Inc. under this license license. *Personal information has been removed due to data-privacy issues*.

I have decided to use R for data cleaning because of the size of the data set. So, I am going to install all packages I need, and upload the data to work with.

```
library(tidyverse)        #helps wrangle data
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)        #helps wrangle date attributes
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)        #helps visualize data
library(scales)         #helps format data
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

Step 1 - Collect data

I will upload the data set, notice that file names have been changed to be more readable.

```
tripdata_2021_06 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_06.csv")
tripdata_2021_07 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_07.csv")
tripdata_2021_08 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_08.csv")
tripdata_2021_09 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_09.csv")
tripdata_2021_10 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_10.csv")
tripdata_2021_11 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_11.csv")
tripdata_2021_12 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2021_12.csv")
tripdata_2022_01 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2022_01.csv")
tripdata_2022_02 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2022_02.csv")
tripdata_2022_03 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2022_03.csv")
tripdata_2022_04 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2022_04.csv")
tripdata_2022_05 <- read.csv("C:/Users/Usuario/Desktop/dataset/tripdata_2022_05.csv")
```

Step 2 - Wrangle data and combine into a single file

Compare column names each of the files. While the names don't have to be in the same order, they DO need to match perfectly before we can use a command to join them into one file.

```
colnames(tripdata_2021_06)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_07)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_08)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_09)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_10)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_11)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2021_12)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2022_01)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2022_02)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2022_03)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2022_04)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(tripdata_2022_05)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

- Combine all the data sets

```
all_trips <- bind_rows(tripdata_2021_06,
                       tripdata_2021_07,
                       tripdata_2021_08,
                       tripdata_2021_09,
                       tripdata_2021_10,
                       tripdata_2021_11,
                       tripdata_2021_12,
                       tripdata_2022_01,
                       tripdata_2022_02,
                       tripdata_2022_03,
                       tripdata_2022_04,
                       tripdata_2022_05)
```

- Then I can see the structure of the new data set

```
str(all_trips)
```

```
## 'data.frame':    5860776 obs. of  13 variables:
##  $ ride_id           : chr  "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C412214" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-06-13 14:31:28" "2021-06-04 11:18:02" "2021-06-04 09:49:35" "2021-06-03 19:56:05" ...
##  $ ended_at          : chr  "2021-06-13 14:34:11" "2021-06-04 11:24:19" "2021-06-04 09:55:34" "2021-06-03 20:21:55" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Process

Step 3: Clean up and add data to prepare for analysis

In this phase, I am going to clean the data, which is very important to be able to analyze it correctly later.

First of all, I will check every column from left to right if appropriate, to find out if there is any issue.

`ride_id`

```
lenght_char_ride_id <- nchar(all_trips$ride_id)
summary(lenght_char_ride_id)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      16      16      16      16      16      16
```

We can see that every row in the ride_id column has 16 characters.

`rideable_type`

```
table(all_trips$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##       3217737        274447       2368592
```

Here, we can see the three kinds of entries in the rideable_type column, and also that, there are no typos.

`started_at` & `ended_at`

```
all_trips$started_at <- ymd_hms(all_trips$started_at)
all_trips$ended_at <- ymd_hms(all_trips$ended_at)
str(all_trips)
```

```
## 'data.frame':    5860776 obs. of  13 variables:
##  $ ride_id           : chr  "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C412214" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct, format: "2021-06-13 14:31:28" "2021-06-04 11:18:02" ...
##  $ ended_at          : POSIXct, format: "2021-06-13 14:34:11" "2021-06-04 11:24:19" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Date format is correct now.

- About station names I think it is not worth doing these kind of analysis because I can't detect typos, and the number of characters is not necessarily the same.
- It would be a good idea to find out which id corresponds to each station because it isn't clear. But, for this, I would have to talk with stakeholders. For practical purposes, I can move forward with the next steps of this analysis.

member_casual

I will change this column name to be more representative.

```
all_trips <- all_trips %>% rename(user_type = member_casual)
head(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 99FEC93BA843FB20 electric_bike 2021-06-13 14:31:28 2021-06-13 14:34:11
## 2 06048DCFC8520CAF electric_bike 2021-06-04 11:18:02 2021-06-04 11:24:19
## 3 9598066F68045DF2 electric_bike 2021-06-04 09:49:35 2021-06-04 09:55:34
## 4 B03C0FE48C412214 electric_bike 2021-06-03 19:56:05 2021-06-03 20:21:55
## 5 B9EEA89F8FEE73B7 electric_bike 2021-06-04 14:05:51 2021-06-04 14:09:59
## 6 62B943CEAAA420BA electric_bike 2021-06-03 19:32:01 2021-06-03 19:38:46
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                              41.80
## 2                                                              41.79
## 3                                                              41.80
## 4                                                              41.78
## 5                                                              41.80
## 6                                                              41.78
##   start_lng end_lat end_lng user_type
## 1    -87.59   41.80  -87.60    member
## 2    -87.59   41.80  -87.60    member
## 3    -87.60   41.79  -87.59    member
## 4    -87.58   41.80  -87.60    member
## 5    -87.59   41.79  -87.59    member
## 6    -87.58   41.78  -87.58    member
```

```
table(all_trips$user_type)
```

```
##
##  casual  member
## 2559857 3300919
```

There are no typos.

I have already checked all columns, now, I will drop all rows that have all null values.

```
all_trips <- drop_na(all_trips)
dim(all_trips)
```

```
## [1] 5855740      13
```

Add data to prepare for analysis

Now, I am going to create new columns that list date, hour, day, month and year of each ride. This will be really useful to make some calculations in next steps.

```
all_trips$date <- as.Date(all_trips$started_at)              #The default format is yyyy-mm-dd
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
all_trips$hour <- as.numeric(format(as.POSIXct(all_trips$started_at), "%H"))
colnames(all_trips)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "user_type"          "date"               "month"
## [16] "day"                "year"               "day_of_week"
## [19] "hour"
```

It can be seen that new columns were added. Also, I will add a column to show length (in seconds) of each ride.

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)
```

```
## [1] TRUE
```

```
str(all_trips)
```

```
## 'data.frame':    5855740 obs. of  20 variables:
##  $ ride_id          : chr  "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C412214" ...
##  $ rideable_type    : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at       : POSIXct, format: "2021-06-13 14:31:28" "2021-06-04 11:18:02" ...
##  $ ended_at         : POSIXct, format: "2021-06-13 14:34:11" "2021-06-04 11:24:19" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id : chr  "" "" "" "" ...
##  $ end_station_name : chr  "" "" "" "" ...
##  $ end_station_id   : chr  "" "" "" "" ...
##  $ start_lat        : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng        : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat          : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng          : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ user_type        : chr  "member" "member" "member" "member" ...
##  $ date             : Date, format: "2021-06-13" "2021-06-04" ...
##  $ month            : chr  "06" "06" "06" "06" ...
##  $ day              : chr  "13" "04" "04" "03" ...
##  $ year             : chr  "2021" "2021" "2021" "2021" ...
##  $ day_of_week      : chr  "Sunday" "Friday" "Friday" "Thursday" ...
##  $ hour             : num  14 11 9 19 14 19 16 17 12 17 ...
##  $ ride_length      : num  163 377 359 1550 248 405 371 378 526 551 ...
```

I have already converted ride_length to a numeric format so I can use it for calculation.

Remove “bad” data

The data frame includes a few hundred entries when bikes were taken out of docks and checked for quality (start_station_name = HQ QR) or ride_length was negative. I will create a new version of the data frame (v2) since data is being removed.

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]
```

Analyse

Step 4: Conduct descriptive analysis

Descriptive analysis on ride_length (all figures in seconds)

```
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     381     680    1162    1234 3356649
```

I will compare members with casual users.

```
all_trips_v2 %>% group_by(user_type) %>% summarize(mean_lenght = mean(ride_length))
```

```
## # A tibble: 2 × 2
##   user_type mean_lenght
##   <chr>           <dbl>
## 1 casual          1672.
## 2 member           767.
```

```
all_trips_v2 %>% group_by(user_type) %>% summarize(median_lenght = median(ride_length))
```

```
## # A tibble: 2 × 2
##   user_type median_lenght
##   <chr>             <dbl>
## 1 casual              914
## 2 member              546
```

```
all_trips_v2 %>% group_by(user_type) %>% summarize(max_lenght = max(ride_length))
```

```
## # A tibble: 2 × 2
##   user_type max_lenght
##   <chr>          <dbl>
## 1 casual       3356649
## 2 member         89996
```

```
all_trips_v2 %>% group_by(user_type) %>% summarize(min_lenght = min(ride_length))
```

```
## # A tibble: 2 × 2
##   user_type min_lenght
##   <chr>          <dbl>
## 1 casual             0
## 2 member             0
```

Days of week will be arranged alphabetically. So, I have to fix it.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"
))
```

```
user_type_mean_d <- all_trips_v2 %>%
  group_by(day_of_week) %>%
  filter(grepl('member', user_type)) %>%
  summarize(mean_lenght_member = mean(ride_length))

casual_mean_d <- all_trips_v2 %>%
  group_by(day_of_week) %>%
  filter(grepl('casual', user_type)) %>%
  summarize(mean_lenght_casual = mean(ride_length))

user_type_mean_d$mean_lenght_casual <- casual_mean_d$mean_lenght_casual

user_type_mean_d_percentage <- user_type_mean_d %>% summarize(day_of_week, mean_lenght_casual, mean_lenght_member, casual_vs_member
= (mean_lenght_casual/mean_lenght_member)-1)

user_type_mean_d_percentage$casual_vs_member <- percent(user_type_mean_d_percentage$casual_vs_member, accuracy = 3)

user_type_mean_d_percentage
```

```
## # A tibble: 7 × 4
##   day_of_week mean_lenght_casual mean_lenght_member casual_vs_member
##   <ord>                   <dbl>              <dbl> <chr>
## 1 Monday                  1666.               744. 123%
## 2 Tuesday                 1452.               723. 102%
## 3 Wednesday               1473.               729. 102%
## 4 Thursday                1505.               732. 105%
## 5 Friday                  1573.               752. 108%
## 6 Saturday                1839.               859. 114%
## 7 Sunday                  1921.               865. 123%
```

I have to reorganize months too.

```
all_trips_v2$month <-
  ordered(all_trips_v2$month, levels = c('06', '07', '08', '09', '10', '11', '12', '01', '02', '03', '04', '05'))
```

```
user_type_mean <- all_trips_v2 %>%
  group_by(month) %>%
  filter(grepl('member', user_type)) %>%
  summarize(mean_lenght_member = mean(ride_length))

casual_mean <- all_trips_v2 %>%
  group_by(month) %>%
  filter(grepl('casual', user_type)) %>%
  summarize(mean_lenght_casual = mean(ride_length))

user_type_mean$mean_lenght_casual <- casual_mean$mean_lenght_casual

user_type_mean_percentage <- user_type_mean %>% summarize(month, mean_lenght_casual, mean_lenght_member, casual_vs_member = (mean_len
ght_casual/mean_lenght_member)-1)

user_type_mean_percentage$casual_vs_member <- percent(user_type_mean_percentage$casual_vs_member, accuracy = 3)

user_type_mean_percentage
```

```
## # A tibble: 12 × 4
##    month mean_lenght_casual mean_lenght_member casual_vs_member
##    <ord>              <dbl>              <dbl> <chr>
##  1 06                2139.               860. 150%
##  2 07                1886.               840. 123%
##  3 08                1645.               830. 99%
##  4 09                1600.               809. 99%
##  5 10                1450.               735. 96%
##  6 11                1205.               665. 81%
##  7 12                1277.               650. 96%
##  8 01                1424.               697. 105%
##  9 02                1325.               663. 99%
## 10 03                1544.               705. 120%
## 11 04                1401.               682. 105%
## 12 05                1533.               784. 96%
```

Observations:

- Regardless of the month or day of week, Casual riders' trips last twice as long as Members'.
- All users take longer trips (long time) on weekends. Ride time increases by 25% for casual riders, and 15% for members.
- The highest average ride times are in the hottest months (June, July y August).

Descriptive analysis on number_of_trips

```
user_type_num_d <- all_trips_v2 %>%
  group_by(day_of_week) %>%
  filter(grepl('member', user_type)) %>%
  summarize(num_trips_member = n())

casual_num_d <- all_trips_v2 %>%
  group_by(day_of_week) %>%
  filter(grepl('casual', user_type)) %>%
  summarize(num_trips_casual = n())

user_type_num_d$num_trips_casual <- casual_num_d$num_trips_casual

user_type_num_d_percentage <- user_type_num_d %>% summarize(day_of_week, num_trips_casual, num_trips_member, casual_vs_member = (num_tr
ips_casual/num_trips_member)-1)

user_type_num_d_percentage$casual_vs_member <- percent(user_type_num_d_percentage$casual_vs_member, accuracy = 3)

user_type_num_d_percentage
```

```
## # A tibble: 7 × 4
##   day_of_week num_trips_casual num_trips_member casual_vs_member
##   <ord>                 <int>           <int> <chr>
## 1 Monday               301586          465993 -36%
## 2 Tuesday              286618          524661 -45%
## 3 Wednesday            285396          512491 -45%
## 4 Thursday             308131          501664 -39%
## 5 Friday               359455          459649 -21%
## 6 Saturday             545234          440882 24%
## 7 Sunday               469306          394535 18%
```

```
user_type_num <- all_trips_v2 %>%
  group_by(month) %>%
  filter(grepl('member', user_type)) %>%
  summarize(num_trips_member = n())

casual_num <- all_trips_v2 %>%
  group_by(month) %>%
  filter(grepl('casual', user_type)) %>%
  summarize(num_trips_casual = n())

user_type_num$num_trips_casual <- casual_num$num_trips_casual

user_type_num_percentage <- user_type_num %>% summarize(month, num_trips_casual, num_trips_member, casual_vs_member = (num_trips_casual/
num_trips_member)-1)

user_type_num_percentage$casual_vs_member <- percent(user_type_num_percentage$casual_vs_member, accuracy = 3)

user_type_num_percentage
```

```
## # A tibble: 12 × 4
##    month num_trips_casual num_trips_member casual_vs_member
##    <ord>          <int>           <int> <chr>
##  1 06            370153          358720 3%
##  2 07            441465          380201 15%
##  3 08            412101          391516 6%
##  4 09            363460          392056 -6%
##  5 10            256826          373916 -30%
##  6 11            106755          252979 -57%
##  7 12             69615          177781 -60%
##  8 01             18463           85221 -78%
##  9 02             21361           94171 -78%
## 10 03             89642          194132 -54%
## 11 04            126121          244811 -48%
## 12 05            279764          354371 -21%
```

Observations:

- On weekends, there are 20% more trips from casual riders than members. But on weekdays, there are almost 40% more rides from members.
- In the hottest months, the number of rides by casual riders barely top members', but in the rest of the year, trips from members top casual riders' by up to almost 80%.
- In both cases, number of trips decreases when temperature does.

Descriptive analysis on rideable_type

```
all_trips_v2 %>%
  group_by(user_type, rideable_type) %>%
  filter(rideable_type != 'docked_bike') %>%
  summarize(number_of_trips = n(), .groups = 'drop') %>%
  arrange(-number_of_trips)
```

```
## # A tibble: 4 × 3
##   user_type rideable_type number_of_trips
##   <chr>     <chr>                 <int>
## 1 member    classic_bike        1980184
## 2 member    electric_bike       1319691
## 3 casual    classic_bike        1233327
## 4 casual    electric_bike       1048847
```

```
all_trips_v2 %>%
  group_by(user_type, rideable_type) %>%
  filter(rideable_type != 'docked_bike') %>%
  summarize(mean_lenght = mean(ride_length), .groups = 'drop') %>%
  arrange(-mean_lenght)
```

```
## # A tibble: 4 × 3
##   user_type rideable_type mean_lenght
##   <chr>     <chr>              <dbl>
## 1 casual    classic_bike       1542.
## 2 casual    electric_bike      1137.
## 3 member    classic_bike        799.
## 4 member    electric_bike       720.
```

Observations:

- Both types of users, use classic bikes more times and for longer than electric bikes.

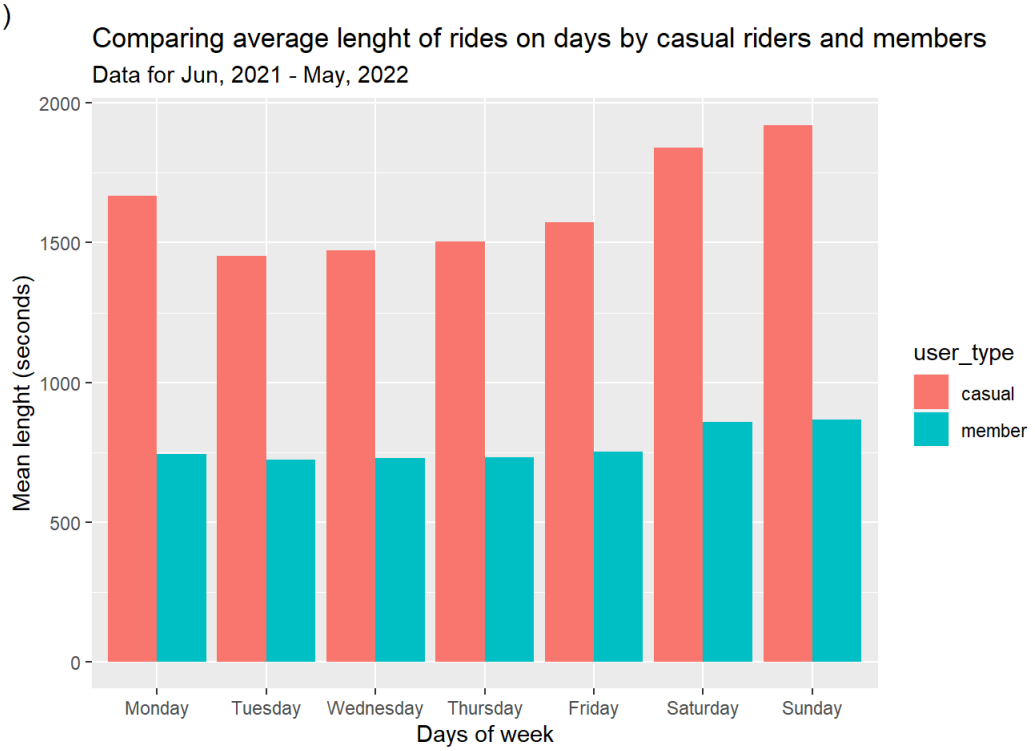
Share

Step 4: Create compelling visualizations

The following visualizations will help stakeholders to gain a better understanding of the business, this will allow them to make data driven decisions.

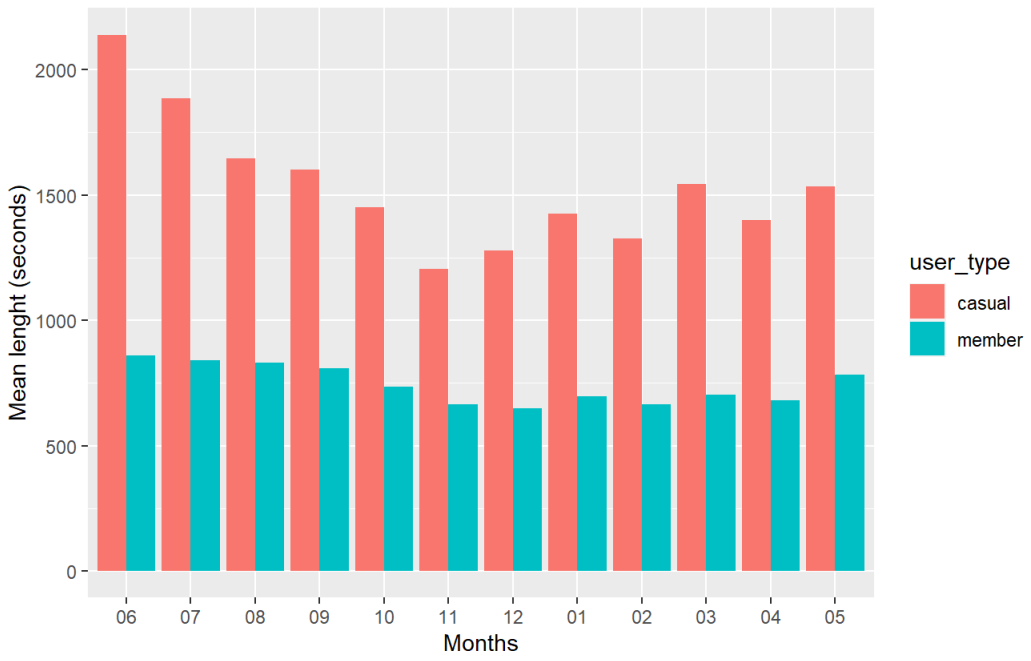
```
all_trips_v2 %>%
  group_by(user_type, day_of_week) %>%
  summarise(mean_lenght = mean(ride_length), .groups = 'drop') %>%
  ggplot(aes(x = day_of_week, y = mean_lenght, fill = user_type)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(tag="1)", title="Comparing average lenght of rides on days by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Days of week", y="Mean lenght (seconds)")
```

1)



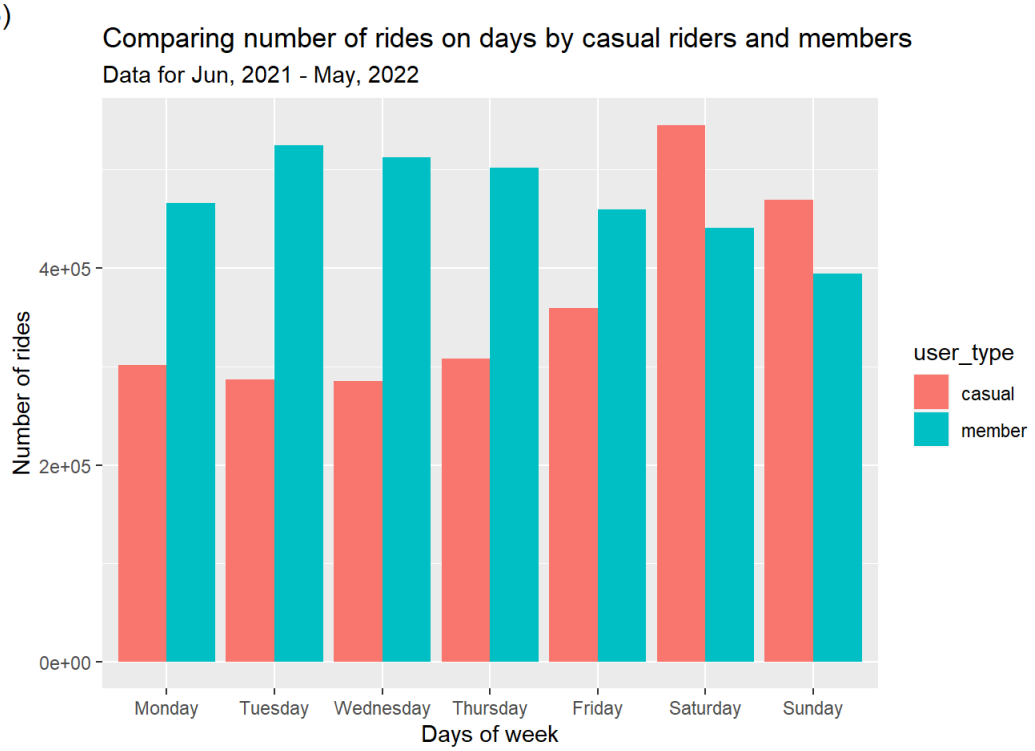

```
all_trips_v2 %>%
  group_by(user_type, month) %>%
  summarise(mean_lenght = mean(ride_length), .groups = 'drop') %>%
  ggplot(aes(x = month, y = mean_lenght, fill = user_type)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(tag="2)", title="Comparing average lenght of rides on months by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Months", y="Mean lenght (seconds)")
```

2)



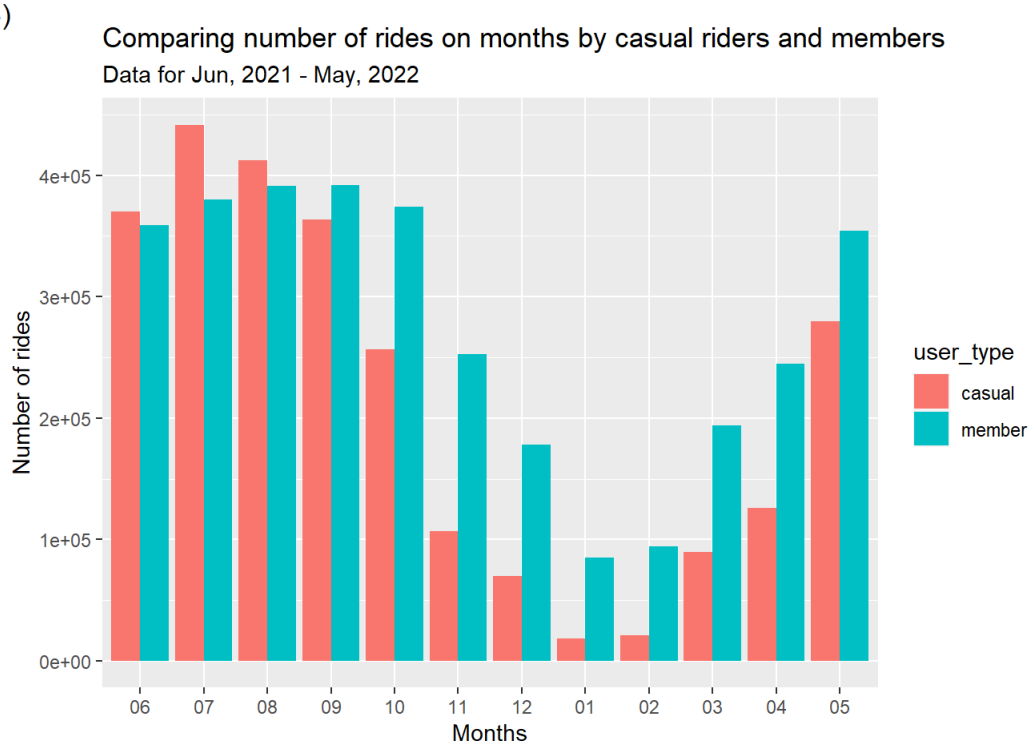

```
all_trips_v2 %>%
  group_by(user_type, day_of_week) %>%
  summarise(number_of_rides = n(), .groups = 'drop') %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = user_type)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(tag="3)", title="Comparing number of rides on days by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Days of week", y="Number of rides")
```

3)



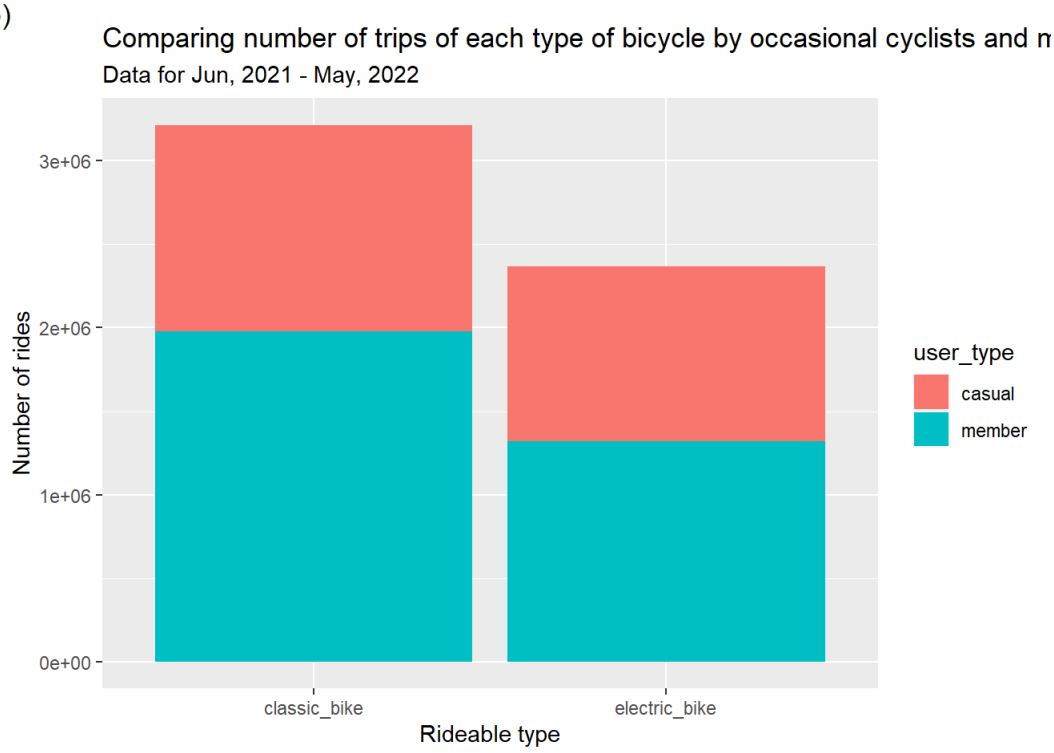

```
all_trips_v2 %>%
  group_by(user_type, month) %>%
  summarise(number_of_rides = n(), .groups = 'drop') %>%
  ggplot(aes(x = month, y = number_of_rides, fill = user_type)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(tag="4)", title="Comparing number of rides on months by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Months", y="Number of rides")
```

4)



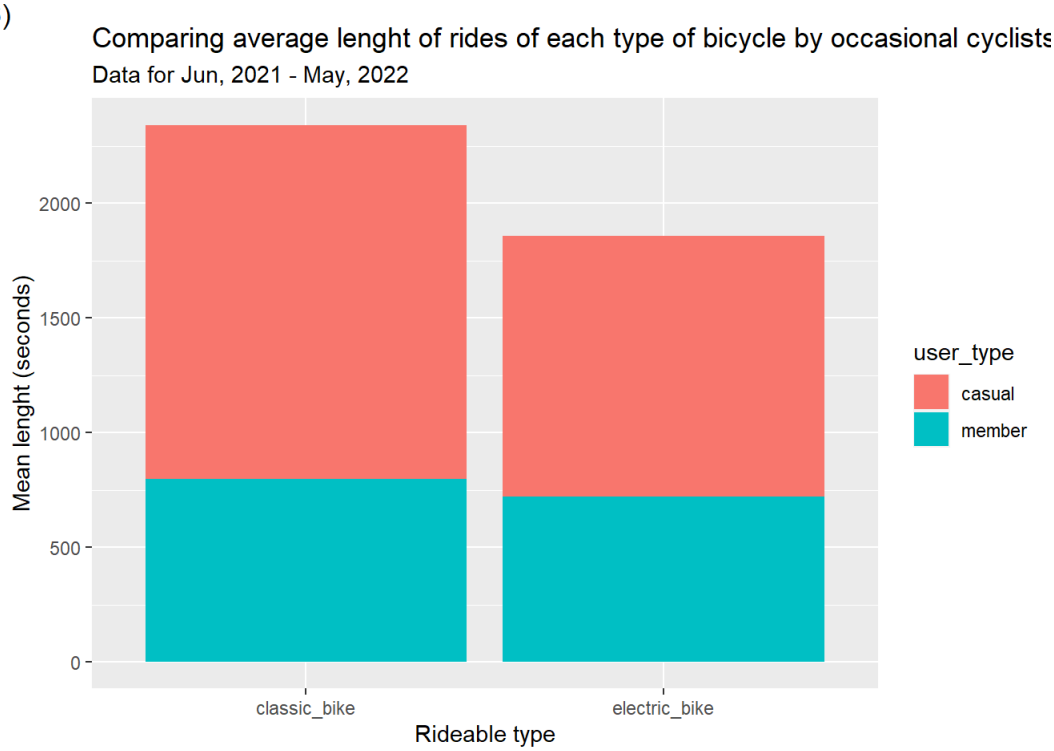

```
all_trips_v2 %>%
  group_by(user_type, rideable_type) %>%
  filter(rideable_type != 'docked_bike') %>%
  summarize(number_of_rides = n(), .groups = 'drop') %>%
  ggplot(aes(x = rideable_type, y = number_of_rides, fill = user_type)) +
  geom_bar(stat = "identity") +
  labs(tag="5)", title="Comparing number of trips of each type of bicycle by occasional cyclists and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Rideable type", y="Number of rides")
```

5)




```
all_trips_v2 %>%
  group_by(user_type, rideable_type) %>%
  filter(rideable_type != 'docked_bike') %>%
  summarize(mean_lenght = mean(ride_length), .groups = 'drop') %>%
  ggplot(aes(x = rideable_type, y = mean_lenght, fill = user_type)) +
  geom_bar(stat = "identity") +
  labs(tag="6)", title="Comparing average lenght of rides of each type of bicycle by occasional cyclists and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Rideable type", y="Mean lenght (seconds)")
```

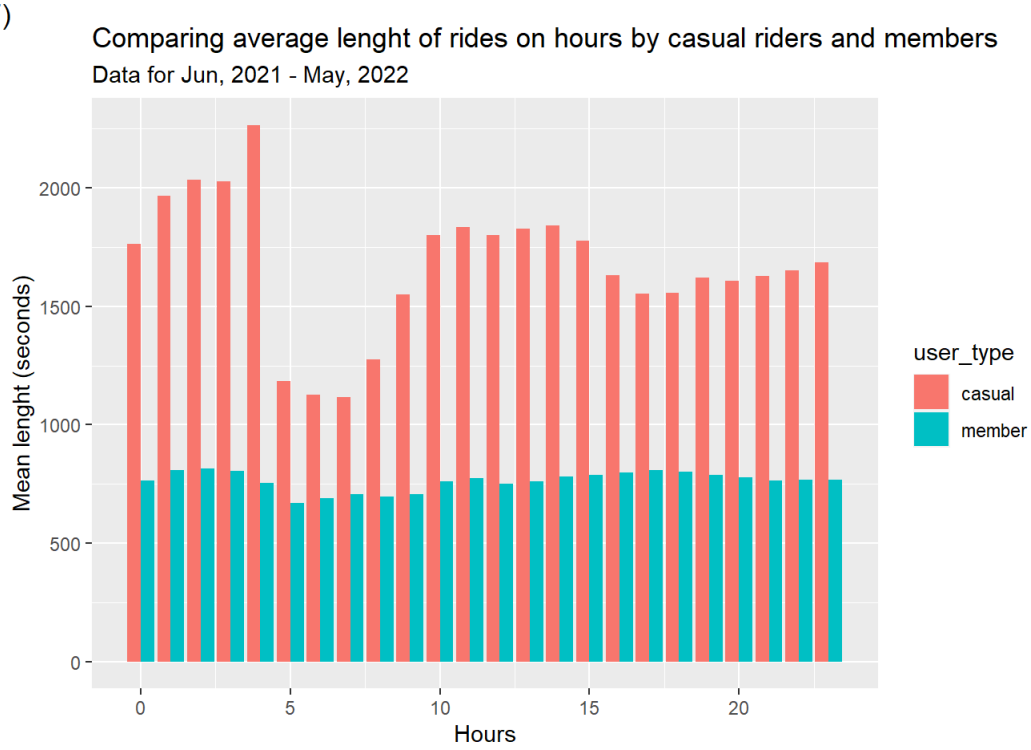
6)



Finally, I want to visualize duration and number of rides depending on the hour

```
all_trips_v2 %>%
  group_by(user_type, hour) %>%
  summarise(mean_lenght = mean(ride_length), .groups = 'drop') %>%
  ggplot(aes(x = hour, y = mean_lenght, fill = user_type)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(tag="7)", title="Comparing average lenght of rides on hours by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Hours", y="Mean lenght (seconds)")
```

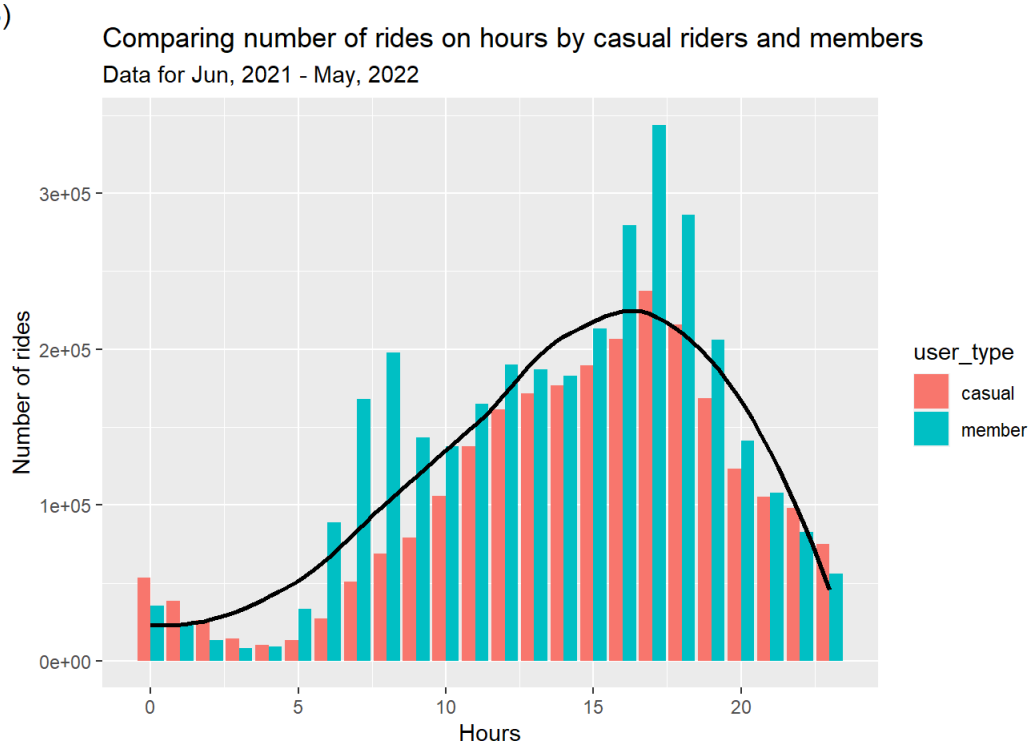
7)




```
all_trips_v2 %>%
  group_by(user_type, hour) %>%
  summarise(number_of_ride = n(), .groups = 'drop') %>%
  ggplot(aes(x = hour, y = number_of_ride)) +
  geom_bar(position = "dodge", stat = "identity", aes(fill = user_type)) + geom_smooth(se = FALSE, color = 'black') +
  labs(tag="8)", title="Comparing number of rides on hours by casual riders and members", subtitle = "Data for Jun, 2021 - May, 2022")+
  labs(x="Hours", y="Number of rides")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

8)



Observations:

- Mean Length of rides in hours is consistent with days and months' duration of trips.
- About number of rides, it starts to increase with the sunrise, maybe because of the light and temperature, and it starts to decrease with the sunset, creating a pattern that both, member and casual riders follow. But, members break that pattern at about 7 - 8 a.m. and again between 5 - 6 p.m. unusually raising the number of trips. This could mean that many of member rides have to do with commuting.

Act

Analyst recommendations

As it can be seen, if we consider user type behaviors (about ride length and number of rides in the different months, and even day of week and the

hour), one hypothesis could be that most of member rides have to do with commuting while individual rides are perhaps more recreational. This hypothesis possibly require a further analysis. Having said that, I will give my recommendations.

- It would be a good idea to offer monthly or seasonal subscriptions to those who want to travel around the city maybe for weeks. These plans could be offered with a set of route map recommendations that riders can take each day of their plan. This way, people who are not interested on annual plans, could get a shorter membership and the company would earn more than just a few individual rides. Also, we can offer summer memberships, that is when there are more rides.
- We can offer discounts to those who have to take the same ride everyday (commuting).
- Also, it would be interesting to give members points for each ride, and that way, they could get prizes. This could attract more members. Also, customers should receive more points the longer the trips, to attract those casual riders who have the highest mean ride length.
- Finally, we can encourage old members to bring new ones, offering both a referral discount for the first month of the new membership.