Cyclistic Case Study

Thiago

15/12/2021

0 - Introduction

This is a Capstone Project for the Google Data Analytics Professional Certification.

0.1 - The Scenario

I'm a junior data analyst, working in the marketing analytics team at Cyclistic, a *fictional* bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. My task for this project is to understand **how casual riders and annual members use Cyclistic bikes differently** and **design a new marketing strategy to convert casual riders into annual members**. But before that, executives must approve the recommendations, with well structured and compelling data insights and visualizations.

0.2 - The Company

Business Model

Since 2016, Cyclistic is a bike-sharing geotracked company across Chicago.

Costumer types:

- Casual Rider, that purchase **single-ride** or **full-day** passes; and
- Annual Member, that purchase an **annual** subscription.

The bikes can be unlocked from one station and returned to any other station in the system anytime.

Bicycle variety:

- Classic;

- Electric; and
- Docked.

Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike.

Stakeholders

- Cyclistic marketing analytics team: responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy;
- Lily Moreno: The director of marketing, is responsible for the development of campaigns and initiatives to promote the bike-share program; and
- Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Facts and Numbers

- 5,824 bikes;
- 692 docking stations;
- More than 50% of riders select traditional bikes;
- 8% of riders opt for the assistive bike options;
- 30% of users bike to commute to work each day;
- Users are more likely to ride for leisure; and
- Casual riders have chosen Cyclistic for their mobility need.

Competitive advantages

- Bicycle variety;
- Pricing flexibility; and
- Adaptive tools to people with disabilities and riders who can't use a standard two-wheeled bike.

0.3 - The Problem

Cyclistic's finance analysts have concluded that **annual members are much more profitable than casual riders**. Although the **pricing flexibility helps Cyclistic attract more customers**, Moreno believes that maximizing the number of annual members will be key to future growth.

Moreno has set a clear goal: **Design marketing strategies aimed at converting casual riders into annual members**. In order to do that, however, the marketing analyst team needs to better understand:

- how annual members and casual riders differ;
- why casual riders would buy a membership; and
- how digital media could affect their marketing tactics.

0.4 - Scope of work

1 - Ask

A clear statement of the business task

2 - Prepare

A description of all data sources used

3 - Process

Documentation of any cleaning or manipulation of data

4 - Analyze

A summary of your analysis

5 - Share

Supporting visualizations and key findings

6 - Act

Top three recommendations based on the analysis

1 - Ask

- A clear statement of the business task

Me and my team will use the six steps of the analysis process (Ask, Prepare, Process, Analyze, Share and Act) to analyze the twelve previous months of Cyclistic historical bike trip data and guide my analysis of the databases to answer the three questions related in "The Problem" above. After that, we will be able to know if the goal of the director of marketing is tangible and (if it is) how is the best way to accomplish that, with recommendations based on the study findings and present to the executive team.

2 - Prepare

- A description of all data sources used

The data are stored at https://divvy-tripdata.s3.amazonaws.com/index.html and organized by month. Since there is no filter, or previous calculating in the raw data and all the consumer are included, we concluded that there is no issues with bias or credibility in the data. To ensure that, we used the ROCCC system:

R - Reliability:

- Missing values represents 3.58% from the total and it's smaller than 10% of statistical significance. So, it's ok.
- 0,03% from the total of registries that haves datetimes starting before start datetimes. In this specific cases we can't fix it, the percentile is to small so the data will be disregarded and will be ok
- There is to much primary keys and with this problem will be more difficult to clean and work with the data, so we can solve it just by filtering variables according to the actual step.

O - Originality:

- The data is original, just with a different name because Cyclistic is a fictional company.

C - Comprehensiveness:

- This database is comprehensive enough.

C - Current:

- The database are up to date and updated monthly

C - Cited:

The data is sourced by a first-party group (Motivate International Inc.), under this license https://www.divvybikes.com/data-license-agreement.

3 - Process

- Documentation of any cleaning or manipulation of data

All this step will be completed in RStudio, this tool is powerful for handling large amounts of data. To maintain the data integrity all the data will be merged and loaded.

3.1 - Loading packages

3.2 - Merging the database

```
raw_data_2020_11 <- read_csv("data/raw/202011-divvy-tripdata.csv")
```

```
## Rows: 259716 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2020_12 <- read_csv("data/raw/202012-divvy-tripdata.csv")
```

```
## Rows: 131573 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_01 <- read_csv("data/raw/202101-divvy-tripdata.csv")
```

```
## Rows: 96834 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_02 <- read_csv("data/raw/202102-divvy-tripdata.csv")
```

```
## Rows: 49622 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_03 <- read_csv("data/raw/202103-divvy-tripdata.csv")
```

```
## Rows: 228496 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_04 <- read_csv("data/raw/202104-divvy-tripdata.csv")
```

```
## Rows: 337230 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_05 <- read_csv("data/raw/202105-divvy-tripdata.csv")
```

```
## Rows: 531633 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_06 <- read_csv("data/raw/202106-divvy-tripdata.csv")
```

```
## Rows: 729595 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_07 <- read_csv("data/raw/202107-divvy-tripdata.csv")
```

```
## Rows: 822410 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_08 <- read_csv("data/raw/202108-divvy-tripdata.csv")
```

```
## Rows: 804352 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_09 <- read_csv("data/raw/202109-divvy-tripdata.csv")
```

```
## Rows: 756147 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
raw_data_2021_10 <- read_csv("data/raw/202110-divvy-tripdata.csv")
```

```
## Rows: 631226 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
data1_merged <- rbind(
  raw_data_2020_11,
  raw_data_2020_12,
  raw_data_2021_01,
  raw_data_2021_02,
  raw_data_2021_03,
  raw_data_2021_04,
  raw_data_2021_05,
  raw_data_2021_06,
  raw_data_2021_07,
  raw_data_2021_08,
  raw_data_2021_09,
  raw_data_2021_10
  )

rm(list = c("raw_data_2020_11",
            "raw_data_2020_12",
            "raw_data_2021_01",
            "raw_data_2021_02",
            "raw_data_2021_03",
            "raw_data_2021_04",
            "raw_data_2021_05",
            "raw_data_2021_06",
            "raw_data_2021_07",
            "raw_data_2021_08",
            "raw_data_2021_09",
            "raw_data_2021_10"
            ))

save(data1_merged, file = "data/processed/data1_merged.RData")
```

3.3 - Pre-exploring and discovering percentile of missing values

Number of rows, number of column and structure

```r
glimpse(data1_merged)
```

```
## Rows: 5,378,834
## Columns: 13
## $ ride_id            <chr> "BD0A6FF6FFF9B921", "96A7A7A4BDE4F82D", "C61526D065~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2020-11-01 13:36:00, 2020-11-01 10:03:26, 2020-11-~
## $ ended_at           <dttm> 2020-11-01 13:45:40, 2020-11-01 10:14:45, 2020-11-~
## $ start_station_name <chr> "Dearborn St & Erie St", "Franklin St & Illinois St~
## $ start_station_id   <chr> "110", "672", "76", "659", "2", "72", "76", NA, "58~
## $ end_station_name   <chr> "St. Clair St & Erie St", "Noble St & Milwaukee Ave~
## $ end_station_id     <chr> "211", "29", "41", "185", "2", "76", "72", NA, "288~
## $ start_lat          <dbl> 41.89418, 41.89096, 41.88098, 41.89550, 41.87650, 4~
## $ start_lng          <dbl> -87.62913, -87.63534, -87.61675, -87.68201, -87.620~
## $ end_lat            <dbl> 41.89443, 41.90067, 41.87205, 41.91774, 41.87645, 4~
## $ end_lng            <dbl> -87.62338, -87.66248, -87.62955, -87.69139, -87.620~
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "~
```

```
glimpse(data1_merged)
```

```
## Rows: 5,378,834
## Columns: 13
## $ ride_id            <chr> "BD0A6FF6FFF9B921", "96A7A7A4BDE4F82D", "C61526D065~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2020-11-01 13:36:00, 2020-11-01 10:03:26, 2020-11-~
## $ ended_at           <dttm> 2020-11-01 13:45:40, 2020-11-01 10:14:45, 2020-11-~
## $ start_station_name <chr> "Dearborn St & Erie St", "Franklin St & Illinois St~
## $ start_station_id   <chr> "110", "672", "76", "659", "2", "72", "76", NA, "58~
## $ end_station_name   <chr> "St. Clair St & Erie St", "Noble St & Milwaukee Ave~
## $ end_station_id     <chr> "211", "29", "41", "185", "2", "76", "72", NA, "288~
## $ start_lat          <dbl> 41.89418, 41.89096, 41.88098, 41.89550, 41.87650, 4~
## $ start_lng          <dbl> -87.62913, -87.63534, -87.61675, -87.68201, -87.620~
## $ end_lat            <dbl> 41.89443, 41.90067, 41.87205, 41.91774, 41.87645, 4~
## $ end_lng            <dbl> -87.62338, -87.66248, -87.62955, -87.69139, -87.620~
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "~
```

```
round(sum(is.na(data1_merged)) /
        (nrow(data1_merged) * ncol(data1_merged)),
      digits =  4) * 100
```

```
## [1] 3.58
```

NAs variables percentile from total **3.58%**. It's smaller than 10% of adopted for the statistical significance level.

3.4 - Adapting database for study

3.4.1 - Filtering not relevant variables

Variable "ride_id" it's not relevant for the study.

```
data2_adapted <- select(data1_merged,
                        everything(),
                        - ride_id)
```

The variables: "start_station_id" and "end_station_id" can be the primary key for the variables "start_station_name" and "end_station_name".

```
data2_adapted <- select(data2_adapted,
                        everything(),
                        - start_station_name,
                        - end_station_name)
```

Latitude and longitude data must match with "start_station_id" and "end_station_id". So, at this time we will not use the following variables "start_lat", "start_lng", "end_lat", "end_lng". Later we will explore the map visualization, and then we will add this variables again.

```
data2_adapted <- select(data2_adapted,
                        everything(),
                        - start_lat,
                        - start_lng,
                        - end_lat,
                        - end_lng)
```

3.4.2 - Adding relevant variables

According to the project's tasks

```
ride_length <- data2_adapted$ended_at - data2_adapted$started_at

day_of_week <- as.Date(data2_adapted$started_at)
day_of_week <- format(day_of_week,"%u")
day_of_week <- as.numeric(day_of_week)
day_of_week <- day_of_week + 1
day_of_week <- replace(day_of_week, day_of_week == 8, 1)
```

Another variables, on my own

```
hour <- hour(data2_adapted$started_at)

month_number <- month(data2_adapted$started_at)
```

3.4.3 - Combining everything

```
data2_adapted <- mutate(data2_adapted,
                        ride_length,
                        hour,
                        day_of_week,
                        month_number
                        )
```

3.4.4 - Checking out and removing used data

```
rm(list = c("ride_length",
            "day_of_week",
            "hour",
            "month_number"
            ))

glimpse(data2_adapted)
```

```
## Rows: 5,378,834
## Columns: 10
## $ rideable_type    <chr> "electric_bike", "electric_bike", "electric_bike", "e~
```

```
## $ started_at       <dttm> 2020-11-01 13:36:00, 2020-11-01 10:03:26, 2020-11-01~
## $ ended_at         <dttm> 2020-11-01 13:45:40, 2020-11-01 10:14:45, 2020-11-01~
## $ start_station_id <chr> "110", "672", "76", "659", "2", "72", "76", NA, "58",~
## $ end_station_id   <chr> "211", "29", "41", "185", "2", "76", "72", NA, "288",~
## $ member_casual    <chr> "casual", "casual", "casual", "casual", "casual", "ca~
## $ ride_length      <drtn> 580 secs, 679 secs, 1741 secs, 555 secs, 2007 secs, ~
## $ hour             <int> 13, 10, 0, 0, 15, 15, 16, 16, 16, 1, 12, 9, 15, 13, 7~
## $ day_of_week      <dbl> 1, 1, 1, 1, 1, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,~
## $ month_number     <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 1~
```

3.4.5 - Cleaning

3.4.5.1 - Identifying problems

```
summary(data2_adapted)
```

```
##  rideable_type        started_at                     ended_at
##  Length:5378834     Min.   :2020-11-01 00:00:08   Min.   :2020-11-01 00:02:20
##  Class :character   1st Qu.:2021-05-17 12:45:18   1st Qu.:2021-05-17 13:07:36
##  Mode  :character   Median :2021-07-13 22:33:14   Median :2021-07-13 22:57:23
##                     Mean   :2021-06-27 18:37:41   Mean   :2021-06-27 18:58:10
##                     3rd Qu.:2021-09-02 18:18:14   3rd Qu.:2021-09-02 18:35:16
##                     Max.   :2021-10-31 23:59:49   Max.   :2021-11-03 21:45:48
##  start_station_id   end_station_id     member_casual      ride_length
##  Length:5378834     Length:5378834     Length:5378834     Length:5378834
##  Class :character   Class :character   Class :character   Class :difftime
##  Mode  :character   Mode  :character   Mode  :character   Mode  :numeric
##
##
##
##       hour        day_of_week     month_number
##  Min.   : 0.00   Min.   :1.000   Min.   : 1.000
##  1st Qu.:11.00   1st Qu.:2.000   1st Qu.: 6.000
##  Median :15.00   Median :4.000   Median : 7.000
##  Mean   :14.26   Mean   :4.114   Mean   : 7.253
##  3rd Qu.:18.00   3rd Qu.:6.000   3rd Qu.: 9.000
##  Max.   :23.00   Max.   :7.000   Max.   :12.000
```

```
min(data2_adapted$ride_length, na.rm = T)
```

```
## Time difference of -1742998 secs
```

The variable ride length must be bigger than zero.

3.4.5.2 - Investigating the percentile from total

```
round(nrow(filter(data2_adapted,
       ride_length <= 0)
     ) / nrow(data2_adapted), digits = 4) * 100
```

```
## [1] 0.03
```

Just 3% of the total observations from “ride_lenght” aren’t bigger than zero

3.4.5.3 - Fixing

```
data3_cleaned <- mutate(data2_adapted,
                        ride_length = replace(
                          ride_length, ride_length <= 0, NA))
```

3.4.5.4 - Checking out the results and the NAs percentile

```
min(data3_cleaned$ride_length, na.rm = T)
```

```
## Time difference of 1 secs
```

```
round(sum(is.na(data3_cleaned)) /
        (nrow(data3_cleaned) * ncol(data3_cleaned)),
      digits =  4) * 100
```

```
## [1] 2.32
```

2.32% of the total observations are now missing values

3.4.5.5 - Saving the process

```
save(data3_cleaned, file = "data/processed/data2_cleaned.RData")
```

4 - Analyze

- A summary of your analysis

This step will be splitted into two studies: * Count Study; and * Ride Length Study.

4.1 Count Study

4.1.1 Using only variables to this study

```
data4_filtered_1count <- select(data3_cleaned,
                                rideable_type,
                                member_casual,
                                hour,
                                day_of_week,
                                month_number)
```

Checking out

```
head(data4_filtered_1count)
```

```
## # A tibble: 6 x 5
##   rideable_type member_casual  hour day_of_week month_number
##   <chr>         <chr>         <int>       <dbl>        <dbl>
## 1 electric_bike casual           13           1           11
## 2 electric_bike casual           10           1           11
## 3 electric_bike casual            0           1           11
## 4 electric_bike casual            0           1           11
## 5 electric_bike casual           15           1           11
## 6 electric_bike casual           15           7           11
```

4.1.2 Exploring data by tables

- Member and Casual Table

```
table_count1_mc <- data4_filtered_1count %>%
  group_by(member_casual) %>%
  summarise(count = n())


table_count1_mc
```

```
## # A tibble: 2 x 2
##   member_casual   count
##   <chr>           <int>
## 1 casual        2470517
## 2 member        2908317
```

- Member and Casual by Hour Table

```
table_count2_mc_hour <- data4_filtered_1count %>%
  group_by(member_casual, hour) %>%
  summarise(count = n())
```

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

```
table_count2_mc_hour
```

```
## # A tibble: 48 x 3
## # Groups:   member_casual [2]
##    member_casual  hour count
##    <chr>         <int> <int>
##  1 casual            0 52016
##  2 casual            1 37785
##  3 casual            2 24389
##  4 casual            3 13425
##  5 casual            4  9491
```

```
##  6 casual            5 11832
##  7 casual            6 24533
##  8 casual            7 44120
##  9 casual            8 60753
## 10 casual            9 73823
## # ... with 38 more rows
```

- Member and Casual by Day Table

```
table_count3_mc_day <- data4_filtered_1count %>%
  group_by(member_casual, day_of_week) %>%
  summarise(count = n())
```

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

```
table_count3_mc_day
```

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual day_of_week  count
##    <chr>               <dbl>  <int>
##  1 casual                  1 476195
##  2 casual                  2 278268
##  3 casual                  3 264350
##  4 casual                  4 267485
##  5 casual                  5 277344
##  6 casual                  6 354993
##  7 casual                  7 551882
##  8 member                  1 368494
##  9 member                  2 391381
## 10 member                  3 431849
## 11 member                  4 444387
## 12 member                  5 425671
## 13 member                  6 425245
## 14 member                  7 421290
```

- Member and Casual by Month Table

```
table_count4_mc_month <- data4_filtered_1count %>%
  group_by(member_casual, month_number) %>%
  summarise(count = n())
```

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

```
table_count4_mc_month
```

```
## # A tibble: 24 x 3
## # Groups:   member_casual [2]
##    member_casual month_number  count
##    <chr>                <dbl>  <int>
##  1 casual                   1  18117
```

```
##  2 casual                   2  10131
##  3 casual                   3  84033
##  4 casual                   4 136601
##  5 casual                   5 256916
##  6 casual                   6 370681
##  7 casual                   7 442056
##  8 casual                   8 412671
##  9 casual                   9 363890
## 10 casual                  10 257242
## # ... with 14 more rows
```

- Member and Casual by Ride Type Table

```
table_count5_mc_ride <- data4_filtered_1count %>%
  group_by(member_casual, rideable_type) %>%
  summarise(count = n())
```

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

```
table_count5_mc_ride
```

```
## # A tibble: 6 x 3
## # Groups:   member_casual [2]
##   member_casual rideable_type   count
##   <chr>         <chr>           <int>
## 1 casual        classic_bike  1226304
## 2 casual        docked_bike    350781
## 3 casual        electric_bike  893432
## 4 member        classic_bike  1840666
## 5 member        docked_bike    113606
## 6 member        electric_bike  954045
```

4.2 Ride Length Study

4.2.1 Using only variables to this study

```
data4_filtered_2length <- select(data3_cleaned,
                                 rideable_type,
                                 member_casual,
                                 hour,
                                 day_of_week,
                                 month_number,
                                 ride_length)
```

Checking out

```
head(data4_filtered_2length)
```

```
## # A tibble: 6 x 6
##   rideable_type member_casual  hour day_of_week month_number ride_length
##   <chr>         <chr>         <int>       <dbl>        <dbl> <drtn>
## 1 electric_bike casual           13           1           11  580 secs
## 2 electric_bike casual           10           1           11  679 secs
## 3 electric_bike casual            0           1           11 1741 secs
## 4 electric_bike casual            0           1           11  555 secs
## 5 electric_bike casual           15           1           11 2007 secs
## 6 electric_bike casual           15           7           11 2961 secs
```

4.2.2 Exploring data by tables

- Member and Casual Table

```
table_length1_mc <- data4_filtered_2length %>%
  group_by(member_casual) %>%
  summarise(mean = round(seconds_to_period(
    mean(ride_length, na.rm = TRUE)), digits = 2),
    min = seconds_to_period(min(ride_length, na.rm = TRUE)),
    max = seconds_to_period(max(ride_length, na.rm = TRUE))
    )

table_length1_mc
```

```
## # A tibble: 2 x 4
##   member_casual mean       min      max
##   <chr>         <Period>   <Period> <Period>
## 1 casual        32M 33.64S 1S       38d 20H 24M 9S
## 2 member        13M 57.85S 1S       1d 1H 59M 56S
```

- Member and Casual by Hour Table

```
table_length2_mc_hour <- data4_filtered_2length %>%
  group_by(member_casual, hour) %>%
  summarise(mean = round(seconds_to_period(
    mean(ride_length, na.rm = TRUE)), digits = 2),
    min = seconds_to_period(min(ride_length, na.rm = TRUE)),
    max = seconds_to_period(max(ride_length, na.rm = TRUE))
    )
```

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

```
table_length2_mc_hour
```

```
## # A tibble: 48 x 5
## # Groups:   member_casual [2]
##    member_casual  hour mean       min      max
##    <chr>         <int> <Period>   <Period> <Period>
##  1 casual            0 34M 50.57S 1S       24d 19H 37M 56S
##  2 casual            1 39M 10.53S 1S       23d 9H 24M 26S
##  3 casual            2 43M 45.17S 1S       38d 20H 24M 9S
```

```
##  4 casual            3 44M 7.62S  2S       18d 11H 25M 41S
##  5 casual            4 48M 52.14S 1S       22d 6H 38M 51S
##  6 casual            5 23M 37.09S 1S       5d 6H 1M 19S
##  7 casual            6 21M 22.28S 1S       8d 2H 54M 3S
##  8 casual            7 21M 58.64S 1S       22d 14H 36M 30S
##  9 casual            8 24M 6.65S  1S       10d 9H 11M 39S
## 10 casual            9 29M 0.51S  1S       8d 5H 4M 14S
## # ... with 38 more rows
```

- Member and Casual by Day Table

```
table_length3_mc_day <- data4_filtered_2length %>%
  group_by(member_casual, day_of_week) %>%
  summarise(mean = round(seconds_to_period(
    mean(ride_length, na.rm = TRUE)), digits = 2),
    min = seconds_to_period(min(ride_length, na.rm = TRUE)),
    max = seconds_to_period(max(ride_length, na.rm = TRUE))
    )
```

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

```
table_length3_mc_day
```

```
## # A tibble: 14 x 5
## # Groups:   member_casual [2]
##    member_casual day_of_week mean       min      max
##    <chr>               <dbl> <Period>   <Period> <Period>
##  1 casual                  1 38M 6.63S  1S       37d 10H 41M 36S
##  2 casual                  2 32M 19.39S 1S       22d 0H 1M 39S
##  3 casual                  3 28M 37.38S 1S       27d 0H 42M 55S
##  4 casual                  4 28M 9.12S  1S       27d 1H 23M 5S
##  5 casual                  5 28M 12.48S 1S       34d 2H 27M 9S
##  6 casual                  6 30M 54.75S 1S       38d 16H 11M 41S
##  7 casual                  7 35M 9.74S  1S       38d 20H 24M 9S
##  8 member                  1 15M 55.26S 1S       1d 0H 59M 56S
##  9 member                  2 13M 31.2S  1S       1d 0H 59M 57S
## 10 member                  3 13M 6.36S  1S       1d 0H 59M 57S
## 11 member                  4 13M 10.03S 1S       1d 0H 59M 58S
## 12 member                  5 13M 6.19S  1S       1d 0H 59M 57S
## 13 member                  6 13M 39.85S 1S       1d 0H 59M 57S
## 14 member                  7 15M 33.5S  1S       1d 1H 59M 56S
```

- Member and Casual by Month Table

```
table_length4_mc_month <- data4_filtered_2length %>%
  group_by(member_casual, month_number) %>%
  summarise(mean = round(seconds_to_period(
    mean(ride_length, na.rm = TRUE)), digits = 2),
    min = seconds_to_period(min(ride_length, na.rm = TRUE)),
    max = seconds_to_period(max(ride_length, na.rm = TRUE))
    )
```

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

```
table_length4_mc_month
```

```
## # A tibble: 24 x 5
## # Groups:   member_casual [2]
##    member_casual month_number mean       min      max
##    <chr>                <dbl> <Period>   <Period> <Period>
##  1 casual                   1 25M 41.08S 1S       13d 18H 25M 55S
##  2 casual                   2 49M 22.69S 2S       20d 22H 9M 14S
##  3 casual                   3 38M 9.66S  1S       22d 0H 1M 39S
##  4 casual                   4 38M 1.56S  1S       33d 4H 16M 42S
##  5 casual                   5 38M 14.11S 1S       37d 10H 41M 36S
##  6 casual                   6 37M 7.56S  1S       38d 20H 24M 9S
##  7 casual                   7 32M 47.61S 1S       34d 2H 27M 9S
##  8 casual                   8 28M 47.45S 1S       28d 21H 49M 10S
##  9 casual                   9 27M 49.13S 1S       22d 19H 38M 32S
## 10 casual                  10 28M 40.7S  1S       28d 6H 25M 1S
## # ... with 14 more rows
```

- Member and Casual by Ride Type Table

```
table_length5_mc_ride <- data4_filtered_2length %>%
  group_by(member_casual, rideable_type) %>%
  summarise(mean = round(seconds_to_period(
    mean(ride_length, na.rm = TRUE)), digits = 2),
    min = seconds_to_period(min(ride_length, na.rm = TRUE)),
    max = seconds_to_period(max(ride_length, na.rm = TRUE))
    )
```

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

```
table_length5_mc_ride
```

```
## # A tibble: 6 x 5
## # Groups:   member_casual [2]
##   member_casual rideable_type mean          min      max
##   <chr>         <chr>         <Period>      <Period> <Period>
## 1 casual        classic_bike  28M 59.87S    1S       1d 1H 59M 56S
## 2 casual        docked_bike   1H 15M 48.64S 1S       38d 20H 24M 9S
## 3 casual        electric_bike 20M 28.63S    1S       8H 7M 16S
## 4 member        classic_bike  14M 19.94S    1S       1d 1H 59M 56S
## 5 member        docked_bike   14M 13.99S    1S       1d 0H 59M 56S
## 6 member        electric_bike 13M 13.32S    1S       8H 0M 31S
```

With all this tables, we can have some idea about the database. But, in the next step we will be able to clearly understand what is going on.

5 - Share

- Supporting visualizations and key findings

Now it's time to create some data viz to support the analysis. To carry out this step, the tables that have been created will be used and one more study will be added, the map study. So it will look like this:

- Count Study Tables and plot by **RStudio**;
- Ride Length Study Tables and plot by **RStudio**; and
- Map Study Tables and plot by **Power BI** and **Tableau**.

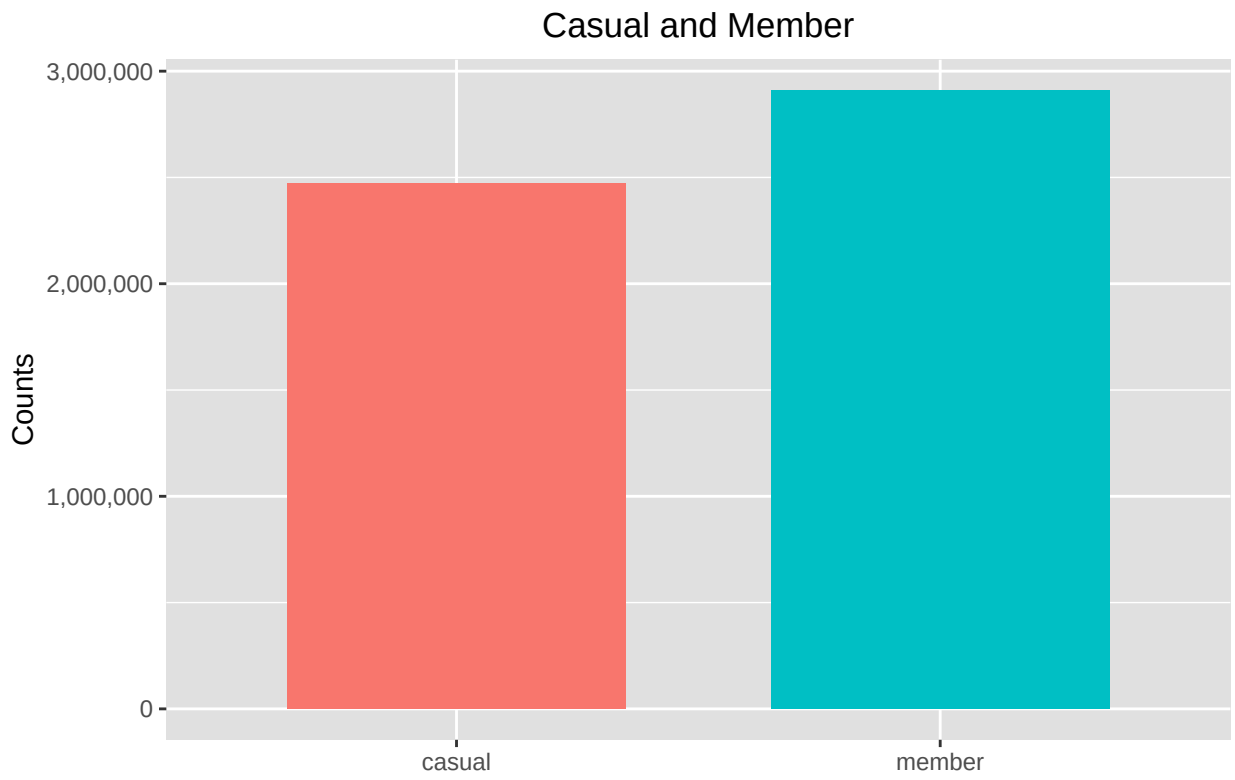
5.1 - Count Study

Now we will see how members and casuals are different about the frequency of rides.

5.1.1 - Member and Casual Viz

- Bar

```
ggplot(data=table_count1_mc,
       aes(x = member_casual, y = count, fill = member_casual)) +
  geom_bar(stat="identity", width = 0.7) +
  labs(title="Casual and Member",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Counts") +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0), legend.position="none",
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```

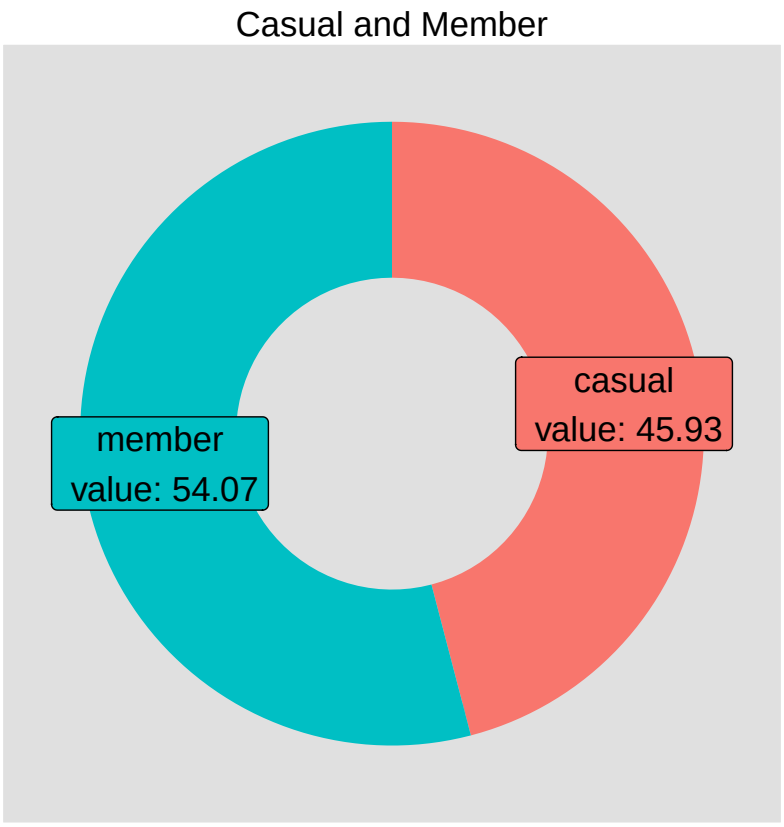


As we can see, most of the rides data are from annual members, but it's not so different.

- Donuts

```
donuts <- data.frame(table_count1_mc$member_casual,
                     table_count1_mc$count)

donuts$fraction <- donuts$table_count1_mc.count / sum(donuts$table_count1_mc.count)
donuts$ymax <- cumsum(donuts$fraction)
donuts$ymin <- c(0, head(donuts$ymax, n = -1))
donuts$labelposition <- (donuts$ymax + donuts$ymin) / 2
donuts$label <- paste0(donuts$table_count1_mc.member_casual,
                       "\n value: ",
                       round(donuts$fraction*100, digits = 2))

ggplot(donuts, aes(ymax=ymax, ymin=ymin,
                   xmax=4, xmin=3,
                   fill = table_count1_mc.member_casual )) +
  geom_rect() +
  geom_label(x = 3.5, aes(y = labelposition, label = label), size = 4.5) +
  coord_polar(theta="y") +
  xlim(c(2, 4)) +
  theme_void() +
  labs(title="Casual and Member",
       caption = "Data from jan/20 to oct/21") +
```

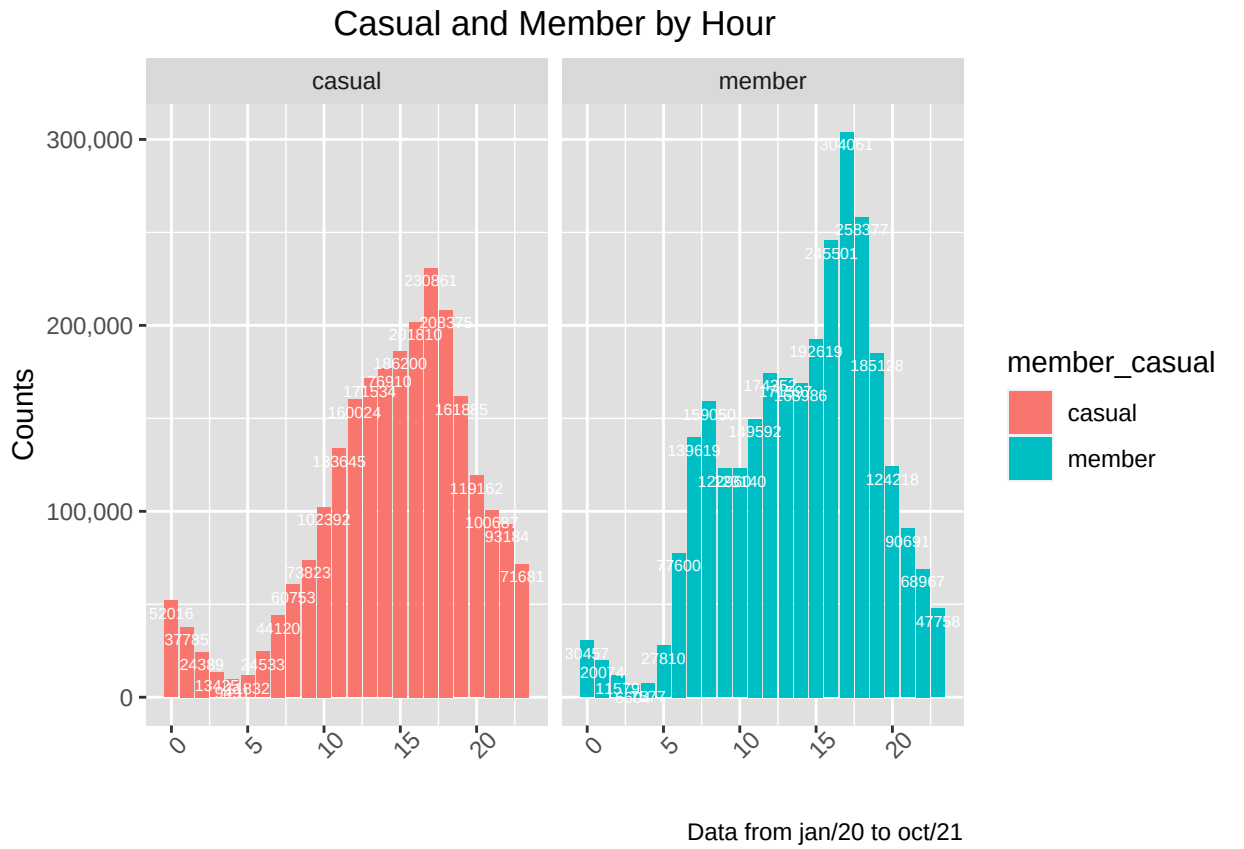
```
  theme(plot.title = element_text(hjust=0.5), legend.position="none",
        panel.background = element_rect(fill = "gray88", color = "white"))
```



Now we have the proportion.

5.1.2 - Hour Viz

```
ggplot(data=table_count2_mc_hour,
       aes(x = hour, y = count, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Hour",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Counts") +
  geom_text(aes(label=count), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 45),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```

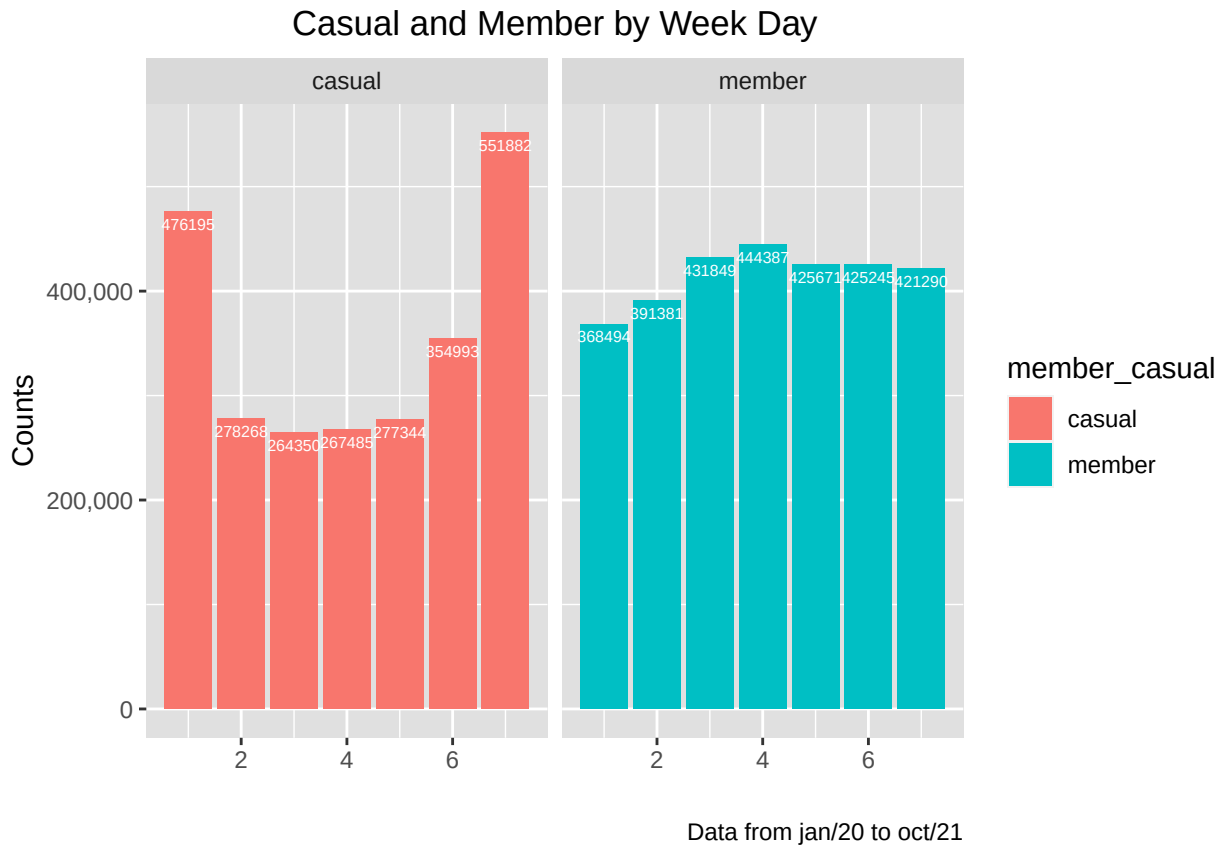


Members and Casuals are different when the variable is the hour to ride. While Members ride mostly at 7am to 7pm, a few Casuals ride at the morning. At night there is more Casuals than Member riding (as we see before, there is more Annual Members than Casuals Riders).

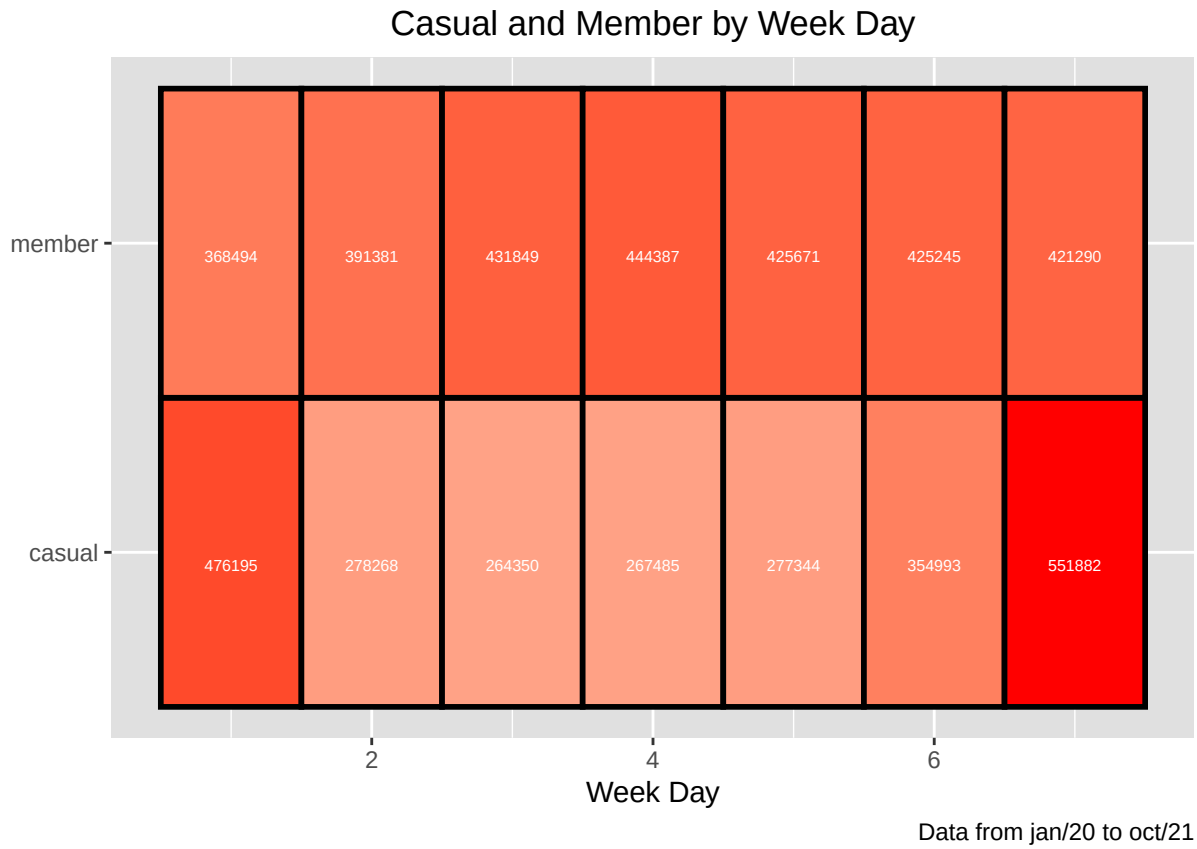
5.1.3 - Day Viz

- Bar

```
ggplot(data=table_count3_mc_day,
       aes(x = day_of_week, y = count, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Week Day",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Counts") +
  geom_text(aes(label=count), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```




```
ggplot(table_count3_mc_day,
       aes(day_of_week, member_casual, fill = count)) +
  geom_tile(color = "black",
            lwd = 1,
            linetype = 1) +
  scale_fill_gradient2(low = "white",
                       mid = "white",
                       high = "red") +
  labs(title="Casual and Member by Week Day",
       caption = "Data from jan/20 to oct/21",
       x = "Week Day",
       y = "") +
  geom_text(aes(label=count), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray88"),
        legend.position = "none")
```

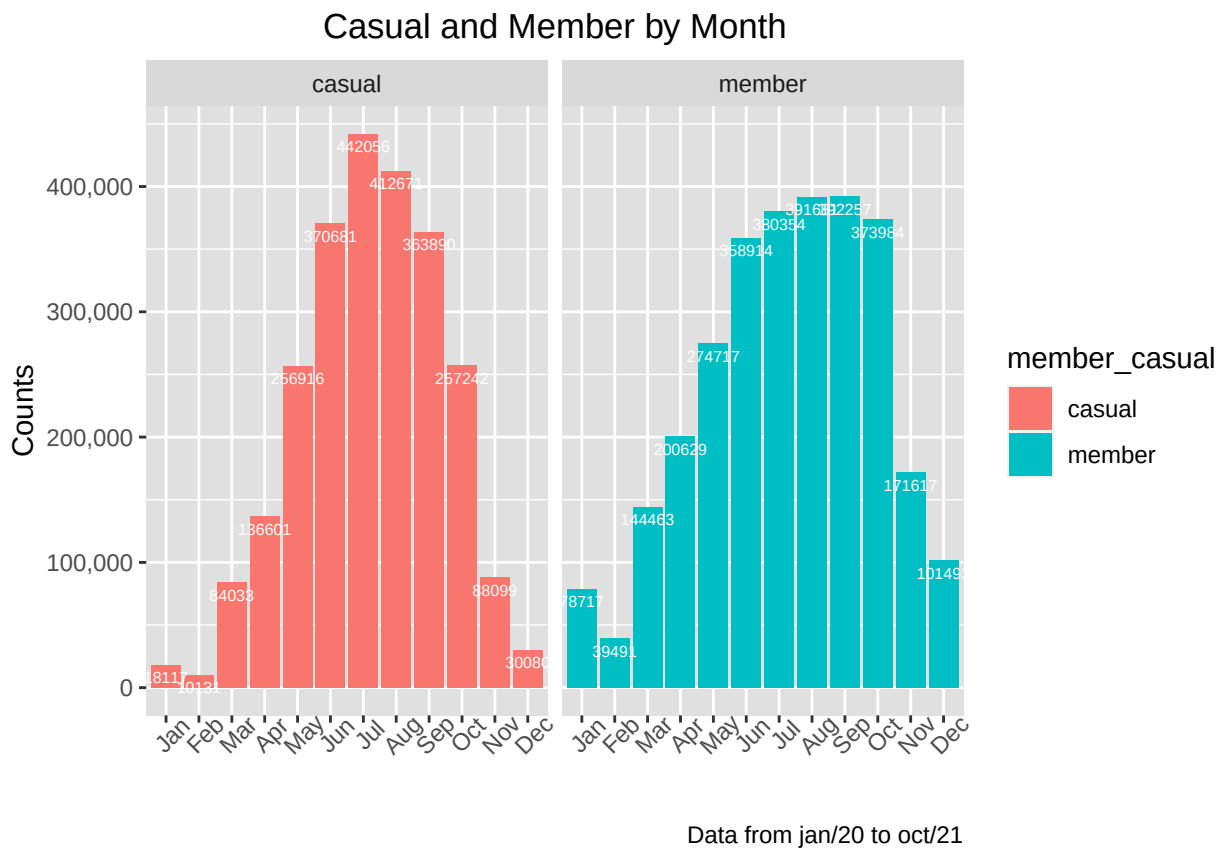


Here is a big difference: Casuals ride much more on Sundays (the first bar) and Saturdays (the last bar), while Members ride constantly in week days.

5.1.4 - Month Viz

```
table_count4_mc_month$month_number <- month.abb[table_count4_mc_month$month_number]

ggplot(data=table_count4_mc_month,
       aes(x = factor(month_number,
                      levels = c("Jan", "Feb", "Mar",
                                 "Apr", "May", "Jun",
                                 "Jul", "Aug", "Sep",
                                 "Oct", "Nov", "Dec")),
           y = count, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Month",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Counts") +
  geom_text(aes(label=count), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
```

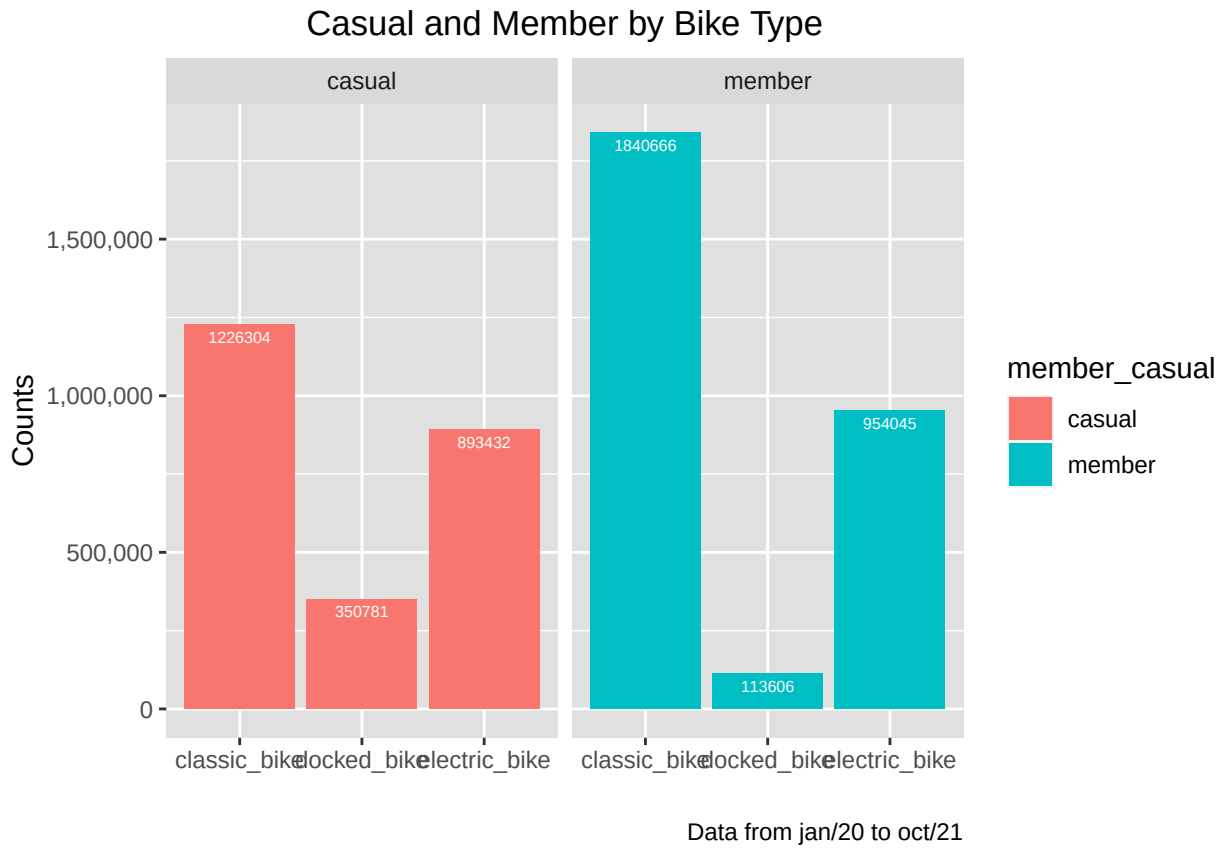
```
        axis.text.x = element_text(angle = 45),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



Both ride more at the summer and less at the winter.

5.1.5 - Rideable Bike Type Viz

```
ggplot(data=table_count5_mc_ride,
       aes(x = rideable_type, y = count, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Bike Type",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Counts") +
  geom_text(aes(label=count), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



Members use much more the classic bike than the docked bike if we compare with Casuals

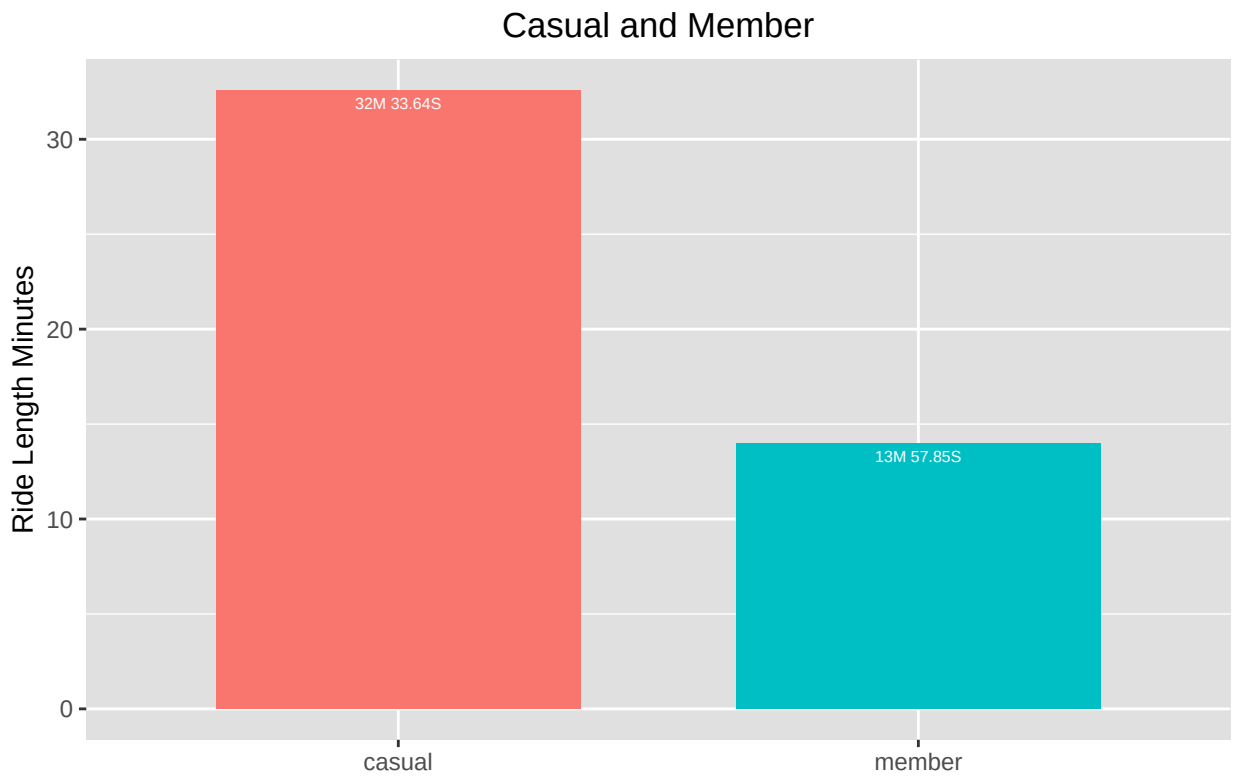
5.2 - Ride Length Study

At this point, we will analyse how members and casuals are different about the ride length in minutes.

5.2.1 - Member and Casual Viz

Bar

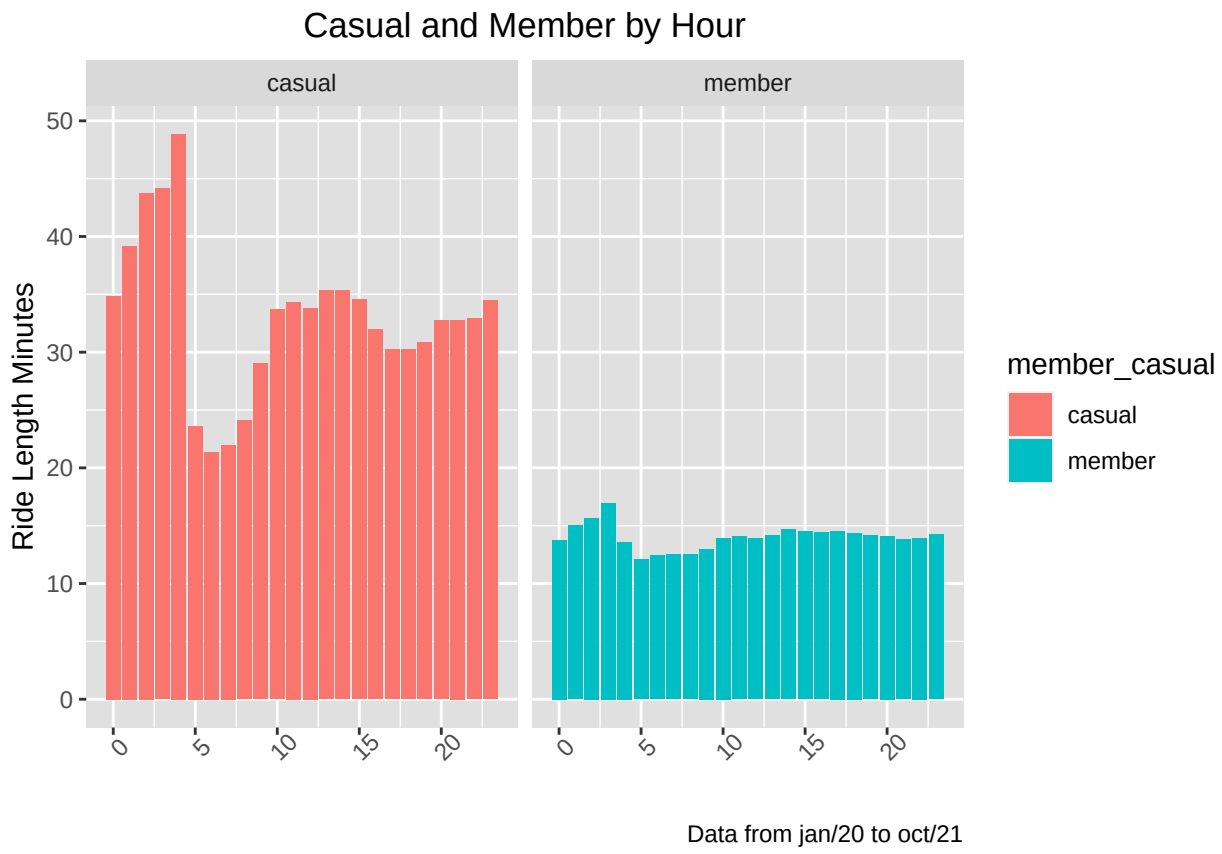
```
ggplot(data=table_length1_mc,
       aes(x = member_casual,
           y = as.numeric(mean)/60,
           fill = member_casual)) +
  geom_bar(stat="identity", width = 0.7) +
  labs(title="Casual and Member",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Ride Length Minutes") +
  geom_text(aes(label=mean), vjust=1.6, color="white",
            position = position_dodge(0.9), size=2,) +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0), legend.position="none",
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



Casuals users spend twice time more than Members pedaling.

5.2.2 - Hour Viz

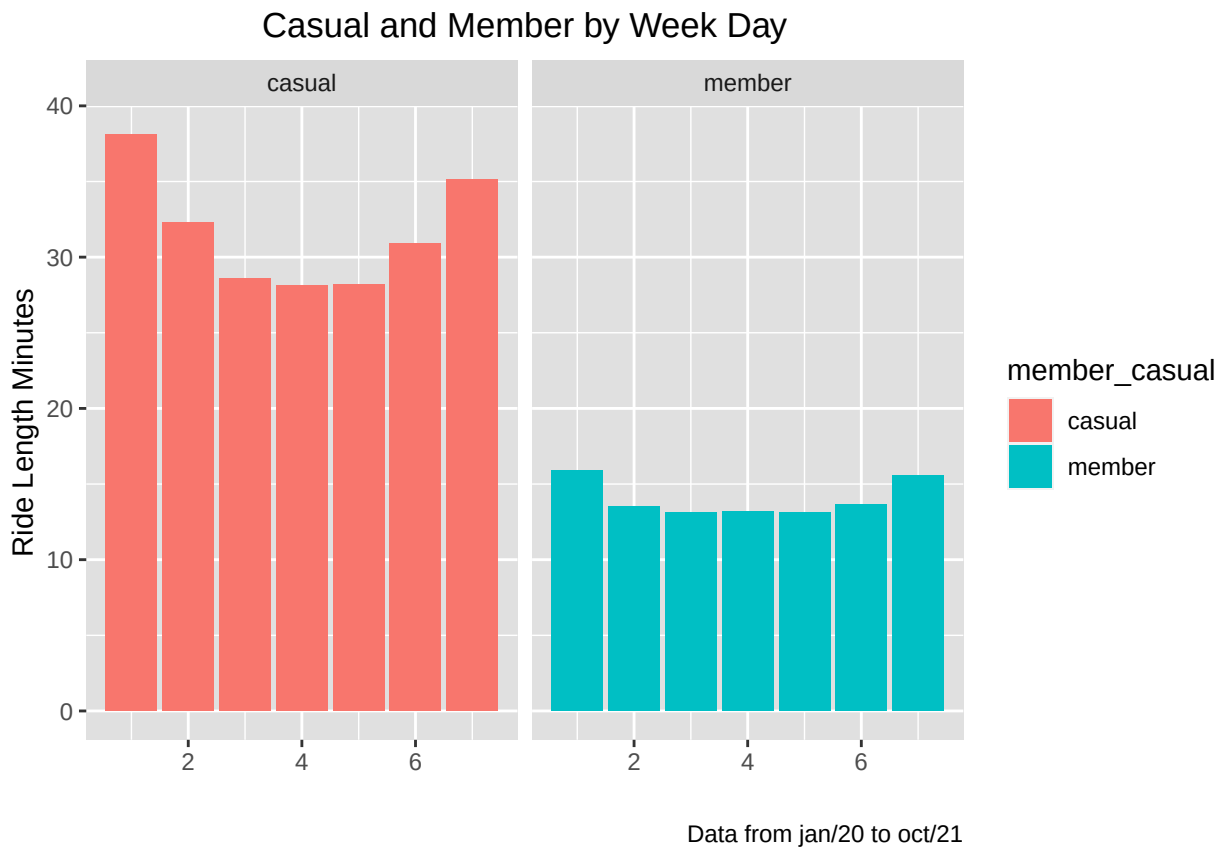
```
ggplot(data=table_length2_mc_hour,
       aes(x = hour, y = as.numeric(mean)/60, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Hour",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Ride Length Minutes") +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 45),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



The length of the rides of members are smaller and more constantly than casuals.

5.2.3 - Day Viz

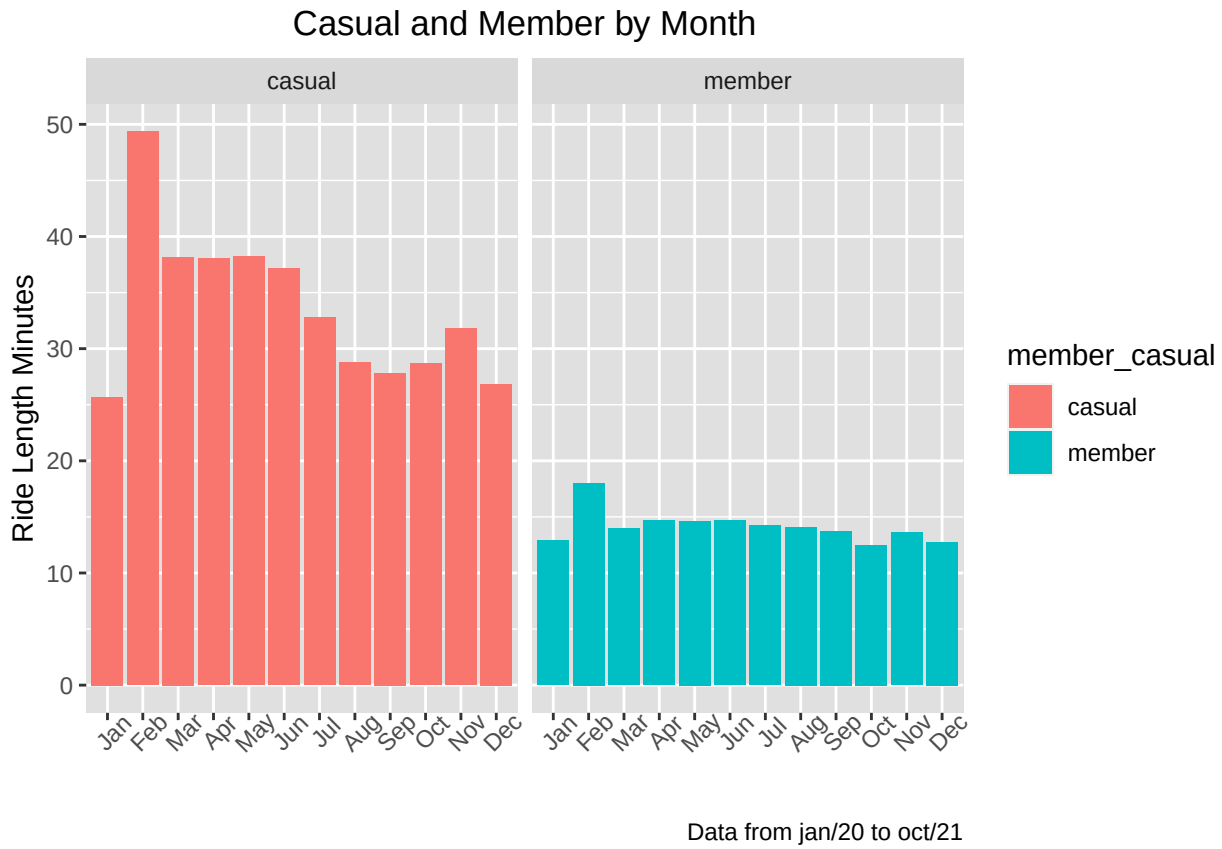
```
ggplot(data=table_length3_mc_day,
       aes(x = day_of_week, y = as.numeric(mean)/60, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Week Day",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Ride Length Minutes") +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



At Sundays and Saturdays Casuals spend more time riding

5.2.4 - Month Viz

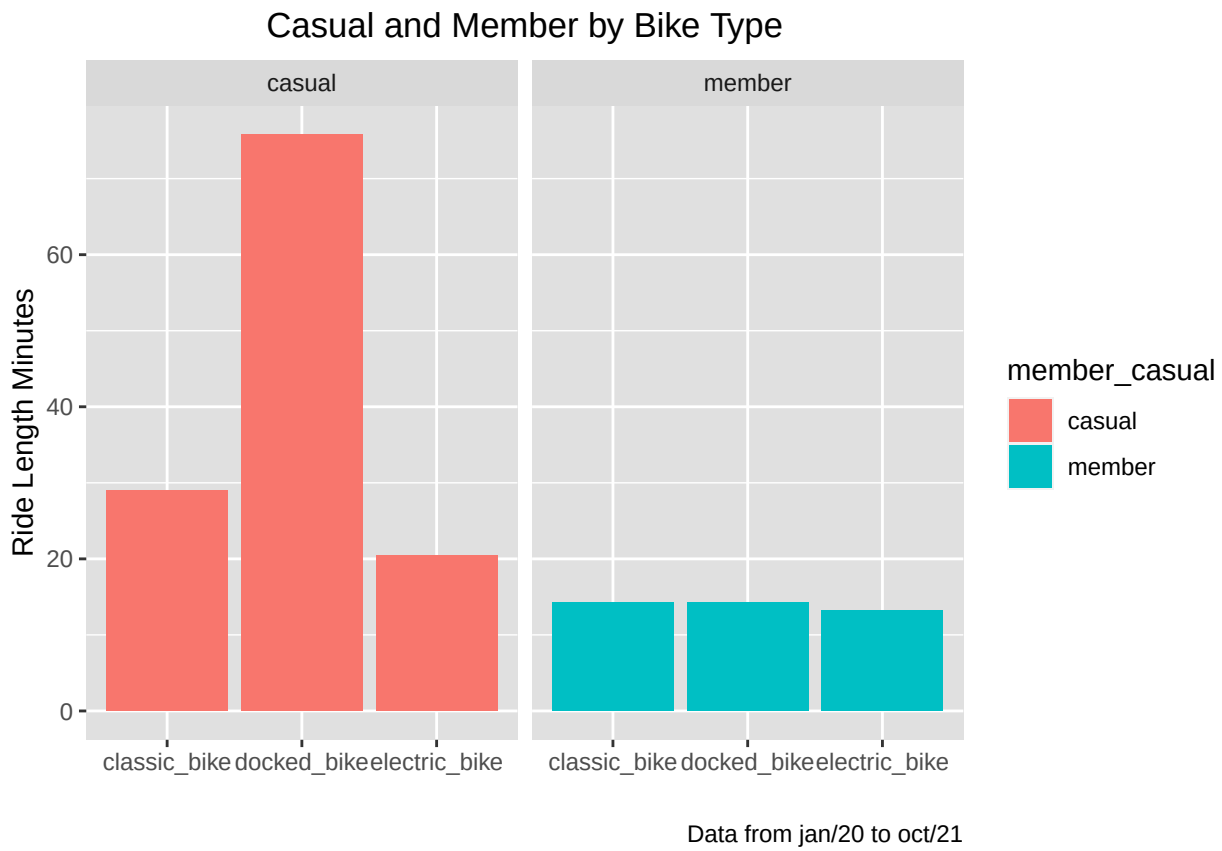
```
table_length4_mc_month$month_number <- month.abb[table_length4_mc_month$month_number]

ggplot(data=table_length4_mc_month,
       aes(x = factor(month_number,
                      levels = c("Jan", "Feb", "Mar",
                                 "Apr", "May", "Jun",
                                 "Jul", "Aug", "Sep",
                                 "Oct", "Nov", "Dec")),
           y = as.numeric(mean)/60, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Month",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Ride Length Minutes") +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 45),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



Independent of the month, Members's ride length are similar

5.2.5 - Rideable Bike Type Viz

```
ggplot(data=table_length5_mc_ride,
       aes(x = rideable_type,
           y = as.numeric(mean)/60, fill = member_casual)) +
  facet_grid(~member_casual) +
  geom_bar(stat="identity",
           position="dodge") +
  labs(title="Casual and Member by Bike Type",
       caption = "Data from jan/20 to oct/21",
       x = "",
       y = "Ride Length Minutes") +
  theme(plot.title = element_text(hjust=0.5),
        axis.text.x = element_text(angle = 0),
        panel.background = element_rect(fill = "gray88")) +
  scale_y_continuous(labels = scales::comma)
```



Casuals spends more time for the docked bike type

5.3 - Map Study (Power BI)

For this visualizations to be possible, since the database has almost 700 stations, it was necessary to applied a filter that shows only stations that registered at least 1000 observations of "start" at the study period (1 year). To ensure the credibility, a test was made and the vast majority of observations are still present.

92.84% from the total of the observations are included when applied the filter

```
table_stations <- data3_cleaned %>%
  group_by(member_casual, start_station_id) %>%
  summarise(count = n())
```

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

```
table_stations_filtered_1000 <- filter(table_stations,
                                       count > 1000)
round(sum(table_stations_filtered_1000$count) / sum(table_stations$count), digits = 4) * 100
```

```
## [1] 92.84
```

5.3.1 - Stations where Members and Casuals most unlock bikes

Casuals are mostly on the waterfront, while the Members are mostly in the city center.

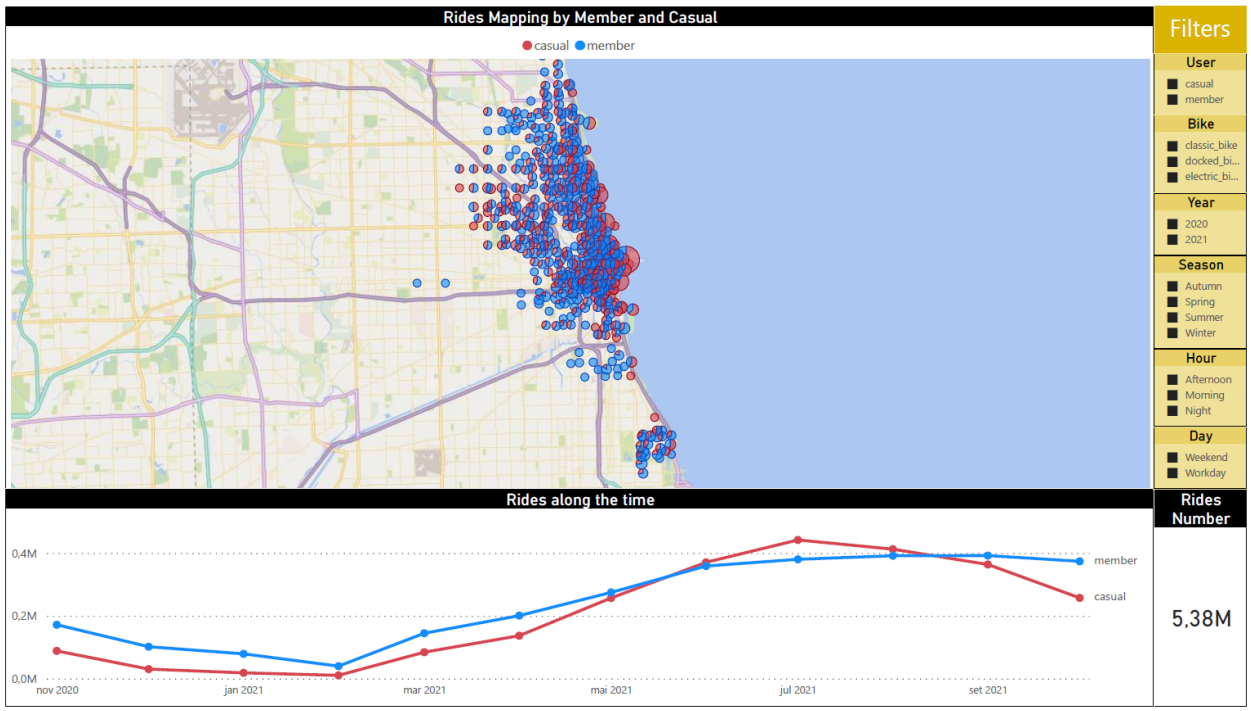
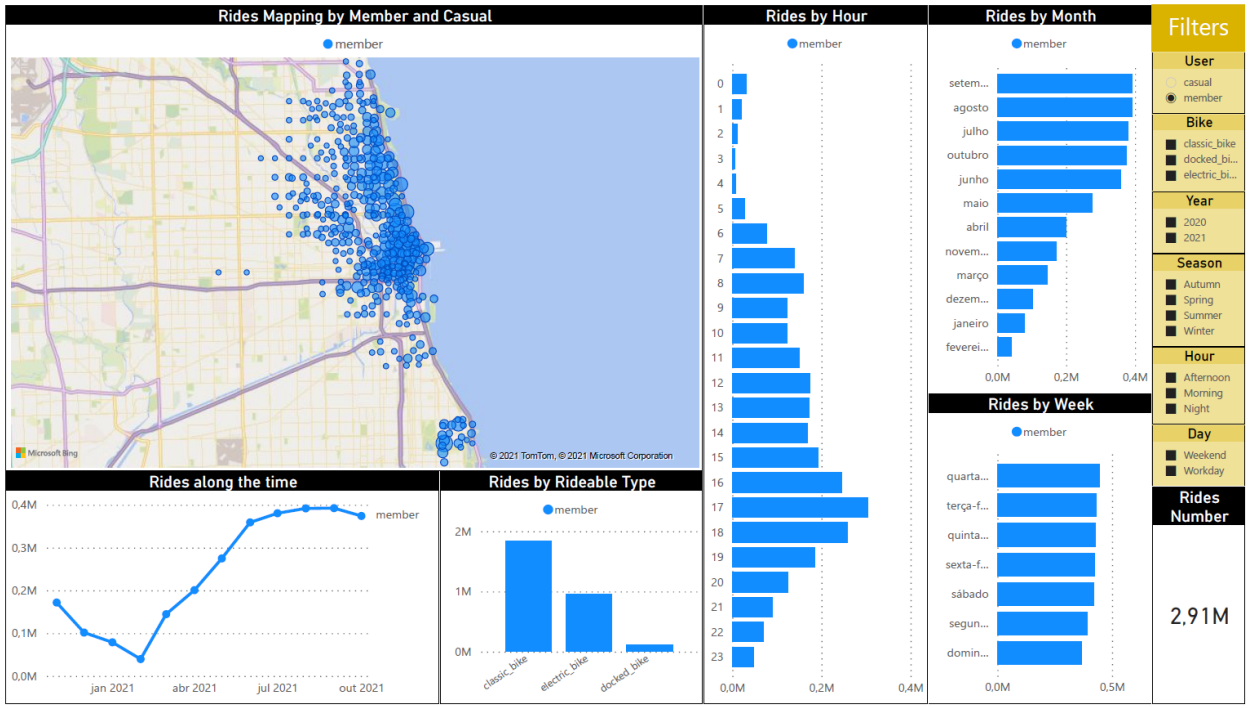
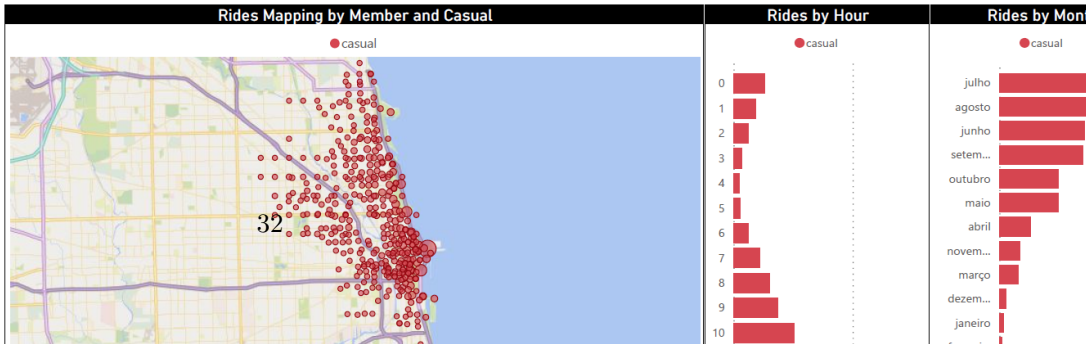


Figure 1: Image through Power BI

5.3.2 - Member Overview





Additional dinamic visualization (Tableau Public)

As it is not possible to publish through Power BI without a subscription, I created a dashboard through Tableau. With that it will be possible to filter the data and specific insights. To access, click on the following link:

https://public.tableau.com/app/profile/thiago.pasolini/viz/Cyclistic_16392298883560/member_casual

5.4 - Key Findings

5.4.1 Differences and Similarities

Members * Most ride at work time, at 7am to 7pm * They use constantly on weekdays * Use much more the classic bike than the docked bike (16x) when compared with Casuals (3.5x)

Casuals * Just few Casuals ride at morning * Although Members majority, at night Casuals are majority * Ride much more on Sundays and Saturdays * Independent of the day of week, time, month or bike type, Casuals spend twice time riding if compared with Members * Rides depart mostly on the waterfront

Both * Ride more at the summer and less at the winter * Rides depart mostly in the city center

5.4.2 Speculations

Based on the information above, with just one relevant finding to both and many differences to each one, we presume that, **they intent to use is for distinct reasons**. And these reasons we make the following assumptions:

- **Casuals** primarily use Cyclistic bikes for **leisure**; and
- **Members** primarily use Cyclistic bikes for commuting to **work** or other **routines**.

6 - Act

6.1 - Recommendation I - Convert Casuals to Member assertively

Design an annual plan for weekend users

It's not possible to design a new marketing strategy that converts all the Casuals to Members and get satisfactory results, because as we saw before, they have different reasons to ride. To make it clearer, here is an example about a Casual rider:

Maria has a car, works from Monday to Friday, lives with her husband and their two children. She always rides on Saturdays morning, she just wants to have a moment to combine fun with healthy at weekends. She has no interested in paying for an annual bike plan that enable her to use every day, there's no sense for her and for the mostly of the users like her (Casuals).

So, the solution is to create a plan to these specific riders. According to the records of the database, they represent almost 20% from the total.

Something we need to be careful about is the current Members. But, the risk of losing them to this campaign is very low due to the fact that we demonstrate earlier, that most Members ride on weekdays, for work and other routines.

Be assertive in the marketing strategy

How can we convert Casuals to Member if they apparently don't want this commitment? Highlighting:

- the mental health benefits of having leisure time, at least weekly;
- the health benefits of keep moving, riding constantly; and
- the real discounts, in a transparent and easy to understand way.

Who uses sporadically on Sundays or Saturdays can now, by being Member, the opportunity to use every Sundays and Saturdays, for the same price.

In few words, the campaign theme can be: more leisure, more healthier and more savings for Casuals riders.

Establish a good trade between cost and benefit

With this action, a good part of Casuals riders will be more loyal and this means lower investment in marketing and more profit. But, this have to be not just more economic than the other annual plan, but a win-win relationship and must be clear to the client's understanding.

Hit the target

Just a good strategy is not enough, you have to hit the target. I was looking for some inspiration on google maps, and well, we know that we want the Casuals rider. So, based on the information that we have, these kind of marketing action may work:

Traditional Publicity:

- Outdoors and Displays along the Chicago waterfront;
- Distribution of leaflets on weekends near Chicago bike stations;
- Create partnerships with food and beverage commerce along the maritime coast, like earn discount by being a Member ; and
- Install campaign display on bicycles

Digital Publicity

- With an mobile app (suggested later, at the Recommendation III), notify Casuals about the campaign; and
- On social medias like (Instagram and Twitter), filtering by Chicago, at Weekends, through Digital Influencers focused in healthy habits and sports.

Keep up a great job of bicycle logistics

After implanted this plan, maybe some dock station will be busier and another less busy. It's important to track closely, in order to avoid bicycle leftovers and shortages.

6.2 - Recommendation II - Create data-driven decision making culture

Know more to be more engaged and with a better reputation

If the company wants more loyalty from the clients, more data about they will be necessary. Knowing consumer's age, genre, where they lives and the main reasons to ride enables the company to create marketing actions more accurate and create products more adapted to customers needs and consequently more connection, more empathy and a longer relationship.

Make the relationship stronger closer through the app

A good and practice way to get consumer data is through the company's app. Create incentives to keep them moving, such as a monthly goal would be an excellent strategy for them to constantly use the app.

6.3 - Recommendation III - Increase Consumer Loyalty

Create a loyalty plan for Members

In addition to making even more loyal those who are already a Member, this will encourage Casuals to be a Member. Examples:

Disciplined use points

- At 3 months using every week, earns 20 Ciclystic Points;
- At 6 months using every week, earns 40 Ciclystic Points;

Distance covered points

- At 500 covered miles earns, 40 Ciclystic Points;
- At 1000 covered miles earns, 80 Ciclystic Points;

Friends indication points

- At the third indication for any annual plan, earns 160 Ciclystic Points (after payment confirmation);
- At the sixth indication for any annual plan, earns 200 Ciclystic Points (after payment confirmation);

Awards personalized by Ciclystic:

- A Towel for 20 Ciclystic Points;

- A T shirt for 40 Ciclystic Points;
- A Cap for 80 Ciclystic Points;
- A Thermos bottle for 150 Ciclystic Points; and
- A Bicycle helmet for 200 Ciclystic Points.

Create a discount system for annual plan renewal

For those who are already a member and wants to renewal the annual plan, regardless if the plan is to every day or just on weekends, concede (just for an example) 5% off.