Google Data Analytics case study: Cyclistic bike-share analysis

Giang Nguyen

2023-01-26

This is my analysis of the case study of the **Google Data Analytics** certificate program. I'll be performing my process of data cleaning, analyzing, and visualizing data, then summarizing the data and delivering insights to solve business questions.

Scenario

You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

Characters and teams

- **Cyclistic:** A bike-share program that features more than 5,800 bicycles and 600 docking stations. 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. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.
- **Lily Moreno:** The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.
- **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals — as well as how you, as a junior data analyst, can help Cyclistic achieve them.
- **Cyclistic executive team:** The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

I'll be following the six phases of data analysis:

- Ask
- Prepare
- Process
- Analyze
- Share
- Act

Ask

In this phase, we define the problem to be solved and make sure to understand the stakeholder expectations.

The problem of this case study is to find out **How do annual members and casual riders use Cyclistic bikes differently?**

The stakeholder expectation is to **Design a new marketing strategy to convert casual riders into annual members.**

Prepare

In this phase we will collect and store data then use for upcomming analysis process.

Identify which kinds of data are most useful for solving a particular problem.

The data can be downloaded at divvy_trip

I will use the most recent year of data based on my current time. The data time frame is from **1/2022 to 12/2022**. The data is separated by each month in each file.

This data is suitable for solving this business problem because it contains insights into riders' patterns of behavior.

Load data

First, we will load the package needed for the data analysis process.

```r
library(tidyverse) # data manipulation, exploration and visualization package
library(skimr) # for checking the structure of the data
library(hydroTSM) # for converting date to seasons
```

Then we load each file to it's respective month.

```r
M1 <- read.csv('F:\\case\\202201-divvy-tripdata.csv')
M2 <- read.csv('F:\\case\\202202-divvy-tripdata.csv')
M3 <- read.csv('F:\\case\\202203-divvy-tripdata.csv')
M4 <- read.csv('F:\\case\\202204-divvy-tripdata.csv')
M5 <- read.csv('F:\\case\\202205-divvy-tripdata.csv')
M6 <- read.csv('F:\\case\\202206-divvy-tripdata.csv')
M7 <- read.csv('F:\\case\\202207-divvy-tripdata.csv')
M8 <- read.csv('F:\\case\\202208-divvy-tripdata.csv')
M9 <- read.csv('F:\\case\\202209-divvy-tripdata.csv')
M10 <- read.csv('F:\\case\\202210-divvy-tripdata.csv')
M11 <- read.csv('F:\\case\\202211-divvy-tripdata.csv')
M12 <- read.csv('F:\\case\\202212-divvy-tripdata.csv')
```

We then check for the column names if they are matched to each other.

```r
# Create a list of month
month_list <- list()
for (i in 2:12){
  month_list <- append(month_list, paste('M', i, sep = ""))
}
# Iterate through the list, then check if the variable exists in the column names of the first month.
for (i in month_list){
  a <- c(colnames(eval(parse(text = i))))
  for (j in a){
    if(!(j %in% colnames(M1))){
      print(j)
    }
  }
}
```

All the columns names of everymonth matched each other

Next We combine every month to a single dataframe

```
data_combined <- rbind(M1, M2, M3, M4, M5, M6, M7,M8, M9, M10, M11, M12)
colnames(data_combined)
```

```
##  [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"
```

We only take the useful variables and exclude the rest. Change irrelevant data type(Char data type to the Date data type).

```
data_combined <- select(data_combined, -c(ride_id, start_station_id, end_station_id, start_lat, start_ln
data_combined <- mutate(data_combined, started_at = as.POSIXct(started_at, format = "%Y-%m-%d %H:%M:%S")
                         ended_at = as.POSIXct(ended_at, format = "%Y-%m-%d %H:%M:%S"))
colnames(data_combined)
```

```
## [1] "rideable_type"      "started_at"         "ended_at"
## [4] "start_station_name" "end_station_name"   "member_casual"
```

Dimensions of the dataframe

```
dim(data_combined)
```

```
## [1] 5667717       6
```

Process

We will find and eliminate any error and inaccuracy in the data.

Add new columns(date, day, month, year, day of the week, season, ride_length)

```
data_processed <- data_combined
data_processed$date <- as.Date(data_processed$started_at)
data_processed$day <- format(as.Date(data_processed$date), '%d')
data_processed$month <- format(as.Date(data_processed$date), '%m')
data_processed$year <- format(as.Date(data_processed$date), '%Y')
data_processed$day_of_the_week <- format(as.Date(data_processed$date), '%A')
data_processed$season <- time2season(as.Date(data_processed$date), out.fmt = "seasons")
```

```
data_processed$ride_length <- difftime(data_processed$ended_at,data_processed$started_at, units = 'mins
data_processed$ride_length <- round(data_processed$ride_length, digits = 1)
```

Change ride_length data type to numeric.

```
data_processed$ride_length <- as.numeric(as.character(data_processed$ride_length))
```

We will be cleaning data next. We remove rows with ride_length below or equal to 0.

```
data_processed <- data_processed[!(data_processed$ride_length <= 0),]
```

```
skim_without_charts(data_combined)
```

Table 1: Data summary

Name	data_combined
Number of rows	5667717
Number of columns	6
Column type frequency:	
character	4
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
rideable_type	0	1	11	13	0	3	0
start_station_name	0	1	0	64	833064	1675	0
end_station_name	0	1	0	64	892742	1693	0
member_casual	0	1	6	6	0	2	0

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2022-01-01 00:00:05	2022-12-31 23:59:26	2022-07-22 15:03:59	4745862
ended_at	0	1	2022-01-01 00:01:48	2023-01-02 04:56:45	2022-07-22 15:24:44	4758633

```
summary(data_processed)
```

```
##  rideable_type        started_at                    
##  Length:5657380     Min.   :2022-01-01 00:00:05.00  
##  Class :character   1st Qu.:2022-05-28 19:46:56.75  
##  Mode  :character   Median :2022-07-22 15:19:45.00  
##                     Mean   :2022-07-20 07:47:12.82  
##                     3rd Qu.:2022-09-16 07:34:44.75  
##                     Max.   :2022-12-31 23:59:26.00  
##     ended_at                      start_station_name end_station_name  
##  Min.   :2022-01-01 00:01:48.00   Length:5657380     Length:5657380    
##  1st Qu.:2022-05-28 20:09:22.25   Class :character   Class :character  
##  Median :2022-07-22 15:41:27.50   Mode  :character   Mode  :character  
##  Mean   :2022-07-20 08:06:41.70                                        
##  3rd Qu.:2022-09-16 07:50:47.50                                        
##  Max.   :2023-01-02 04:56:45.00                                        
##  member_casual           date                day               month          
##  Length:5657380     Min.   :2021-12-31   Length:5657380     Length:5657380    
##  Class :character   1st Qu.:2022-05-28   Class :character   Class :character  
##  Mode  :character   Median :2022-07-22   Mode  :character   Mode  :character  
##                     Mean   :2022-07-19                                        
##                     3rd Qu.:2022-09-16                                        
##                      Max.   :2022-12-31
##      year            day_of_the_week       season           ride_length
##  Length:5657380     Length:5657380     Length:5657380     Min.   :    0.10
##  Class :character   Class :character   Class :character   1st Qu.:    5.80
##  Mode  :character   Mode  :character   Mode  :character   Median :   10.30
##                                                           Mean   :   19.48
##                                                           3rd Qu.:   18.50
##                                                           Max.   :41387.20
```

```
nrow(filter(data_processed, data_processed$start_station_name == ''))
```

```
## [1] 831146
```

```
summary(data_processed$ride_length)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.10     5.80    10.30    19.48    18.50 41387.20
```

Calculate mean by member casual

```
data_processed %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_length))
```

```
## # A tibble: 2 x 2
##   member_casual  mean
##   <chr>         <dbl>
## 1 casual         29.2
## 2 member         12.7
```

```
nrow(filter(data_processed, data_processed$member_casual == 'casual'))/nrow(data_processed)
```

```
## [1] 0.4098233
```

```
nrow(filter(data_processed, data_processed$member_casual == 'member'))/nrow(data_processed)
```

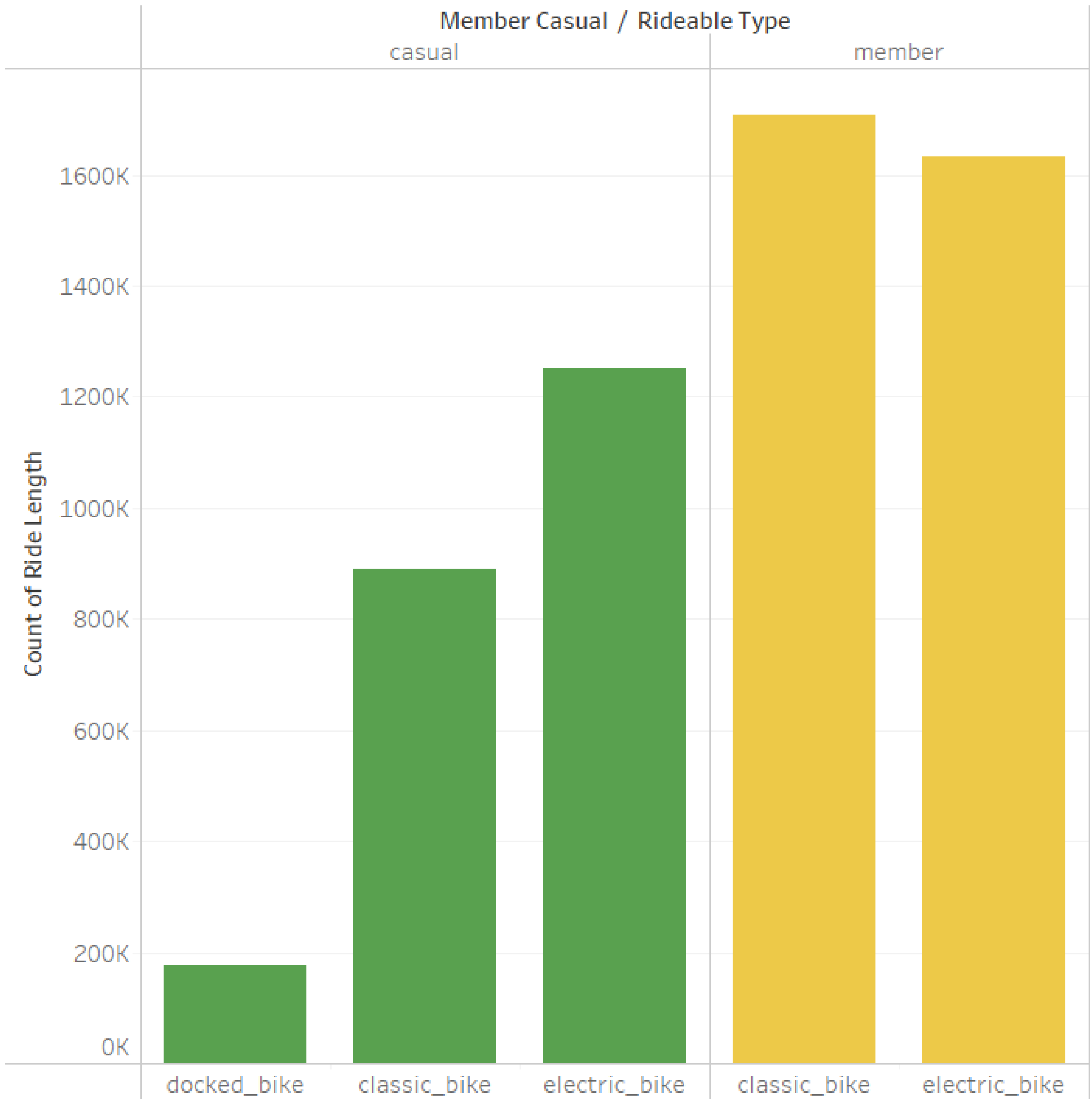
```
## [1] 0.5901767
```

Analyze

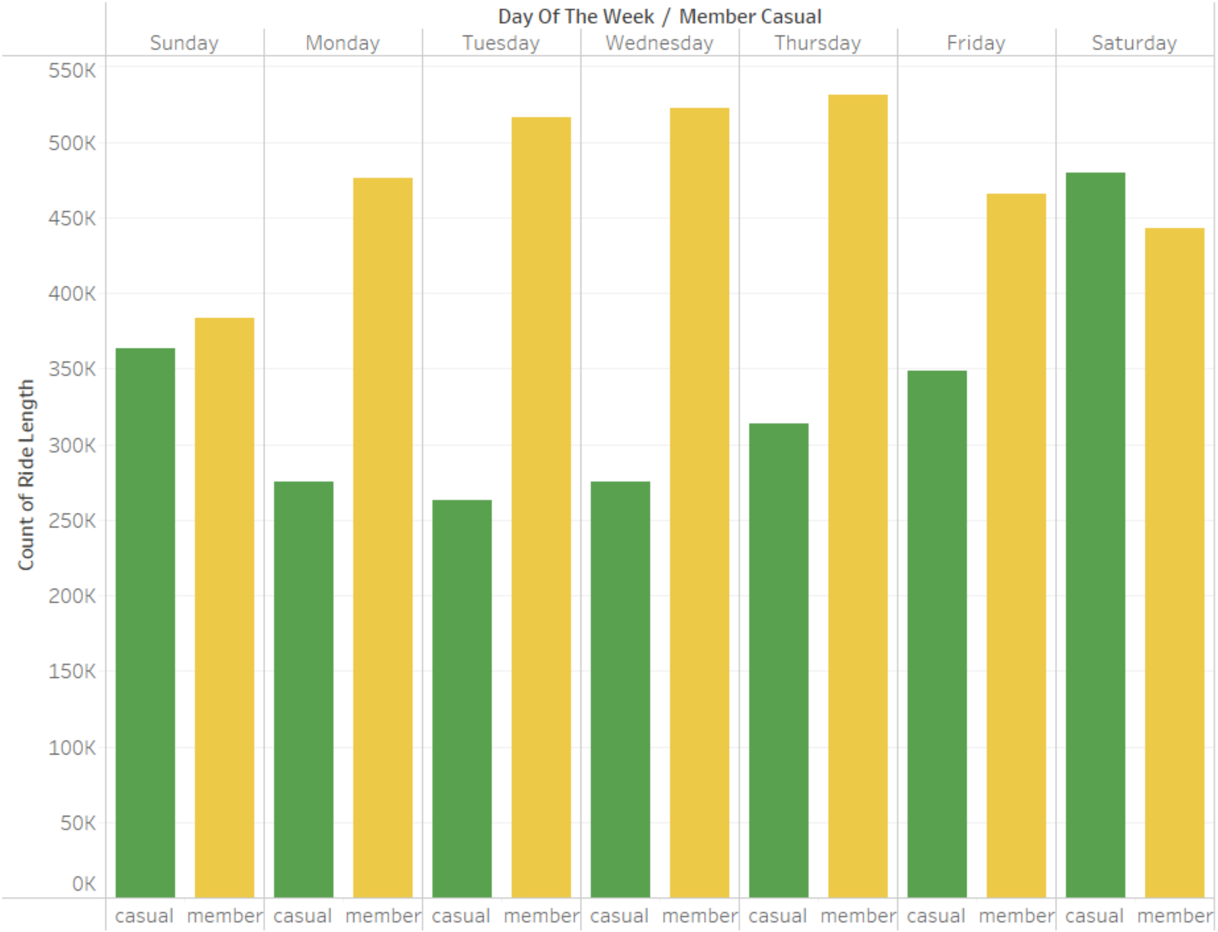
For this phase we will be using tools to transform and organize information so we can draw useful information/conclusion.

We can use R to visualize data and transform it in any way we like. But I would like to use Tableau as practice because it is also part of the course and I want to make use of every tool I was taught.

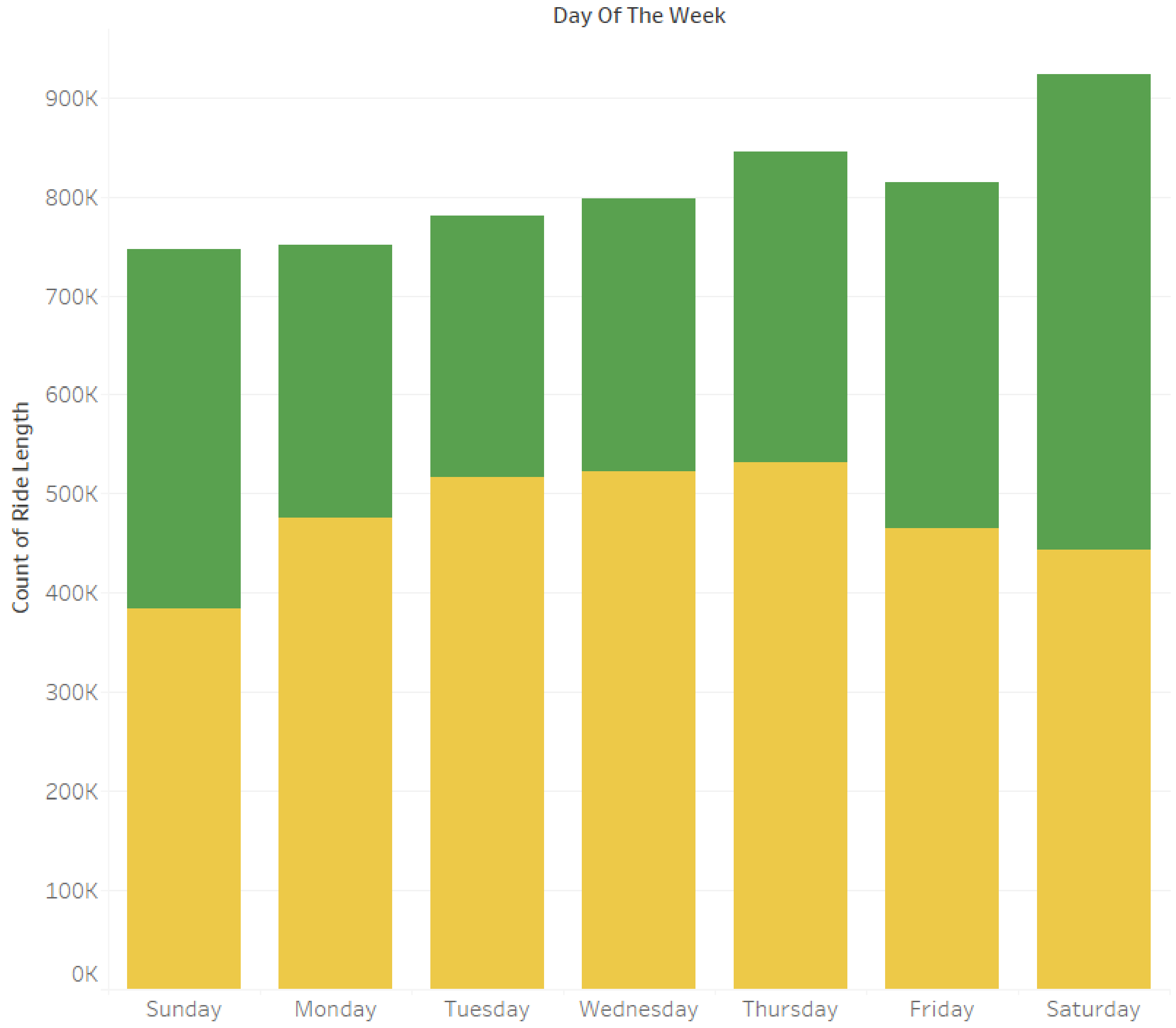
Bike Type prefered by Rider type

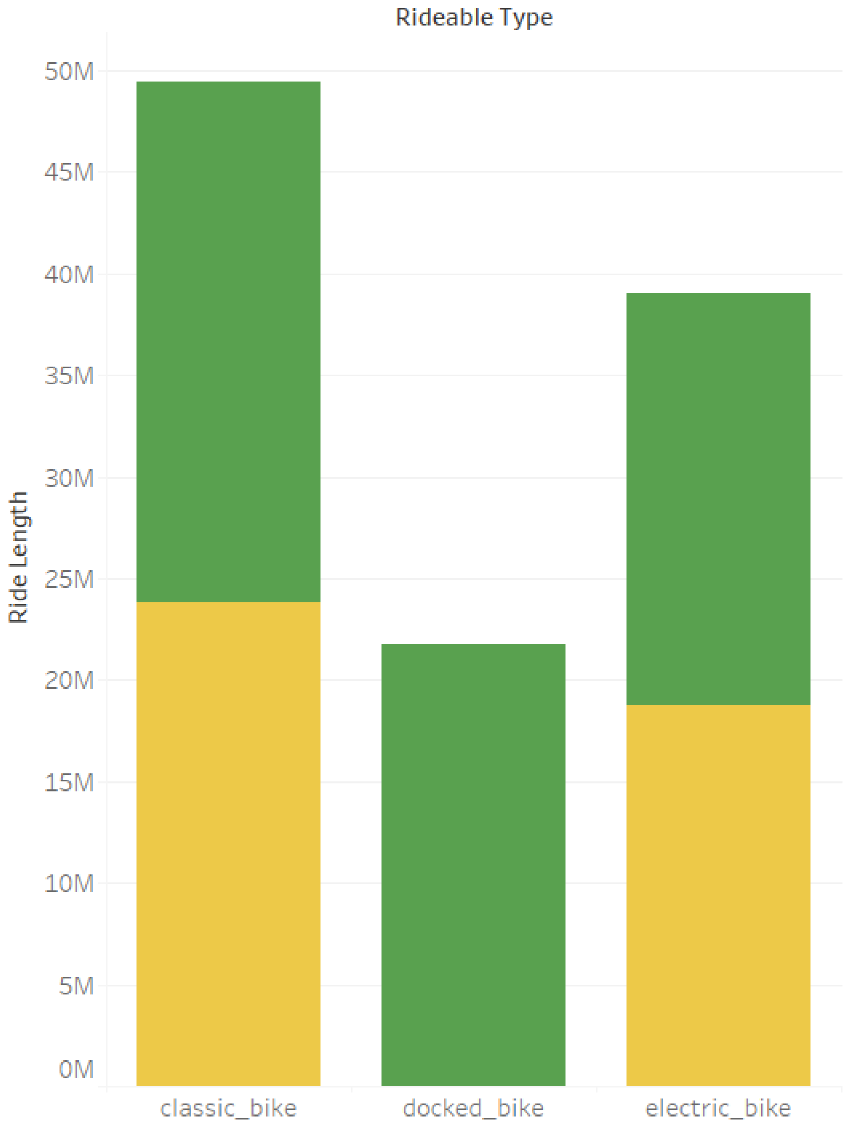


Total rides by weekdays and rider type

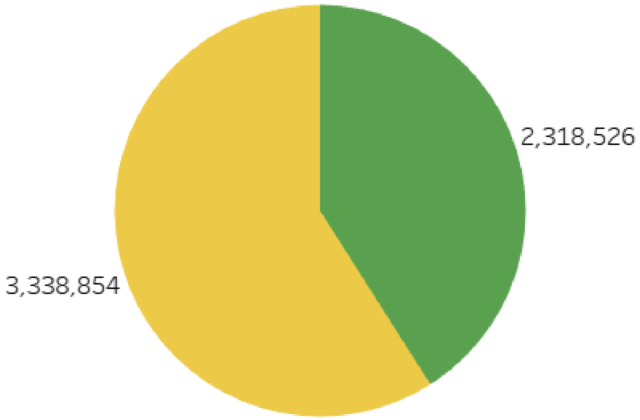


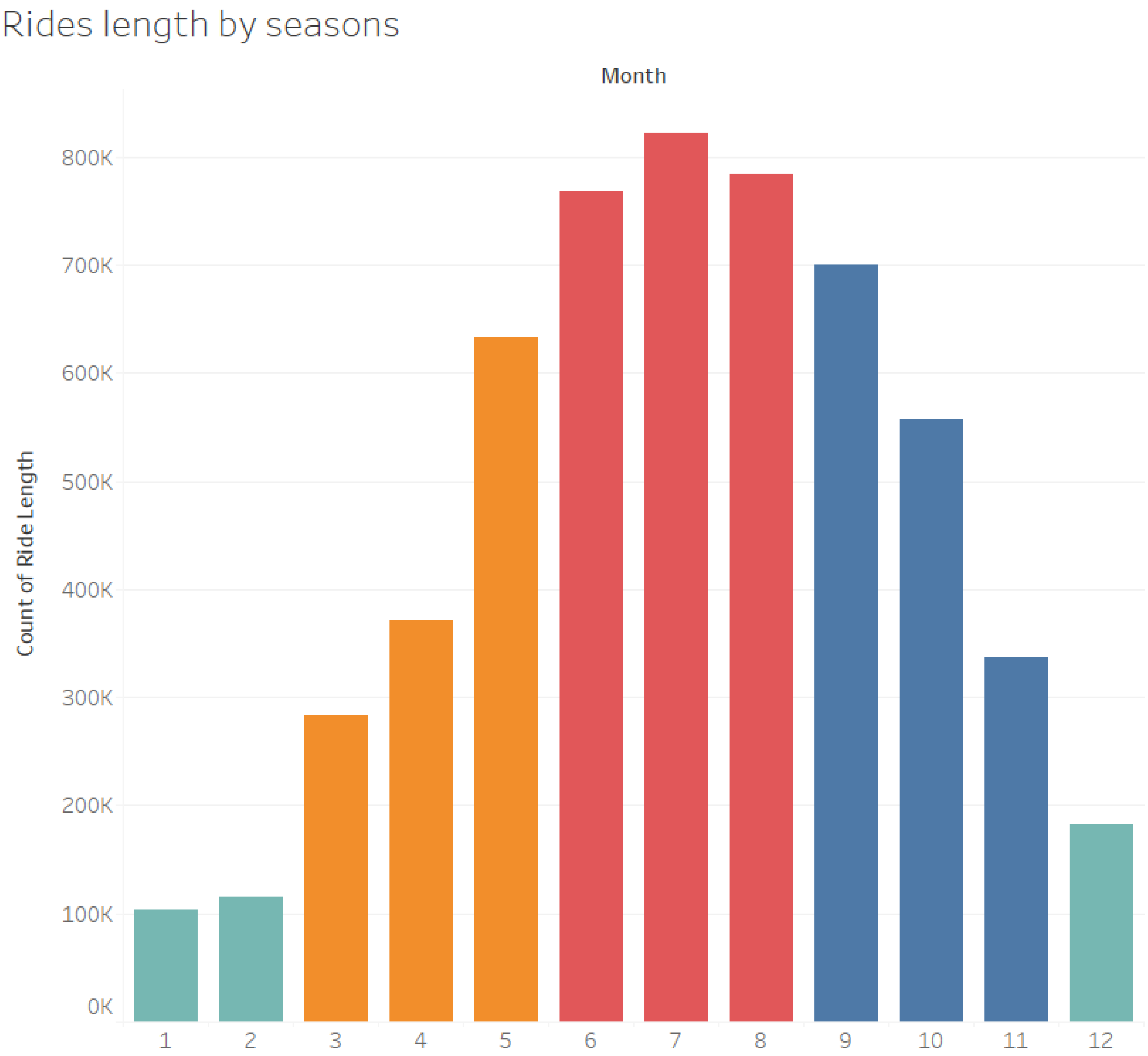
Total rides by weeksdays




Rideable Type
Ride Length
50M
45M
40M
35M
30M
25M
20M
15M
10M
5M
0M
classic_bike
docked_bike
electric_bike

Percent of member type on total rides





For the Tableau Dashboard you can visit Here

After visualization, we can infer some information:

- Members take **59%** of total rides.
- The most **popular bike types** are classic and electric. For casual riders, they prefer docked bikes more than member riders.
- Riders **ride the most** on **Saturday**. Member riders ride equally throughout the weekdays with slight increase in mid weekdays. Casual riders ride equally on weekdays and more on weekends.
- **Summer** is the **most busy** season for both riders type and **Winter** is the **least busy** season for both riders type.

Share

We will interpret result and share with others to help stakeholders making data-driven-decision

We will demonstrate and make a presentation to the stakeholders to deliver our findings in order to help them make a decision on the problem.

Act

- To turn casual riders to member, we can make discount on the time when riders ride the most like Summer or on the weekend.
- Show them the perks of becoming a member, customize the discount and membership program for their specific riding habits.