BikeSharing Case Study

By Daniil Mikhelkis

2023-08-30

# Adjust margins as needed

# Remove page number from the title page

BikeSharing Case Study By Daniil Mikhelkis

Date: 2023-08-30

1 Introduction

1.1 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.

1.2 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.

1.3 About the company

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 bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. **Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.** 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.** Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs. **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. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

2 Analyzing steps

Firstly, we are going to divide our analysis into 7 different phases: **Ask-Prepare-Process-Analyze-Share-Act**

2.1 Ask

Three questions will guide the future marketing program:

- *How do annual members and casual riders use Cyclistic bikes differently?*
- *Why would casual riders buy Cyclistic annual memberships?*
- *How can Cyclistic use digital media to influence casual riders to become members?*

Lets identify the **Buisness Task**:

Based on data, design marketing strategies aimed at converting casual riders into annual members since maximizing the number of annual members will be key to future growth of the company.

My analysis will be focused on answering the first question: *How do annual members and casual riders use Cyclistic bikes differently?*

2.2 Prepare

The data has been made available by Motivate International Inc. under this license.

Data for the analysis is located here.

We download the Data sets ***from January 2022 to December 2022*** and store into ***"Coursera__Capstone__Project"*** folder.

Then rename files according to the file-naming conventions to simplify the readability and management. (Ex.: "202201-divvy-tripdata.csv" -> "2022_01_tripdata.csv")

2.3 Process

Since the data set is large, we will be using **Rstudio** IDE for the analysis, which also possesses advanced data manipulation and visualization tools.

The ***Process*** phase will require a *Tidyverse* package collection:

```
install.packages("tidyverse")
library(tidyverse)
```

We import our files:

```
Jan_2022 <-read.csv("2022_01_tripdata.csv")
Feb_2022 <-read.csv("2022_02_tripdata.csv")
...
Dec_2022 <-read.csv("2022_12_tripdata.csv")
```

We merge all our data sets into one named ***"all__trips__2022"***, since they all share identical format(same number of columns and their names), we wil be using ***bind__rows()*** function from **dplyr** package.

```
all_trips_2022 <- bind_rows(Jan_2022, Feb_2022, Mar_2022, Apr_2022, May_2022, Jun_2022, Jul_2022, Aug_2
```

Lets **clean** the data set: We will be using ***clean_names*** function(which removes special characters and spaces, converts column names to lowercase) from **janitor** package:

```
install.packages("janitor")
library(janitor)
all_trips_2022 <- clean_names(all_trips_2022)
```

Removing lat, long as this data was dropped beginning in 2020:

```
all_trips_2022 <- all_trips_2022 %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

There are a few problems we will need to fix:

1. (1) The data can only be aggregated at the ride-level, which is too granular. We will want to add some additional columns of data – such as day, month, year – that provide additional opportunities to aggregate the data.
2. (2) We will want to add a calculated field for length of ride.(we can use basic subtraction of two columns or use ***difftime()*** function)
3. (3) There are some rides where tripduration(ride_length) shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. We will want to delete these rides.

As we can see the ***"started_at"*** and ***"ended_at"*** columns are ***character*** type, for further analysis and manipulation, we will have to convert it into ***Date*** type, using ***as.POSIXct()*** function from **lubridate** package to extract required data:

(1)Converting data type and extracting additional columns

```
all_trips_2022$ended_at <- as.POSIXct(all_trips_2022$ended_at, format = "%Y-%m-%d %H:%M:%S")
all_trips_2022$started_at <- as.POSIXct(all_trips_2022$started_at, format = "%Y-%m-%d %H:%M:%S")
all_trips_2022$year <- year(all_trips_2022$started_at)
all_trips_2022$month <- month(all_trips_2022$started_at)
all_trips_2022$day <- day(all_trips_2022$started_at)
all_trips_2022$hour <- hour(all_trips_2022$started_at)
all_trips_2022$week_day <- wday(all_trips_2022$started_at)
```

(2)Adding new column ***trip_duration*** and converting it into 'numeric' data type for further analysis

```
all_trips_2022$trip_duration <- all_trips_2022$ended_at - all_trips_2022$started_at
all_trips_2022$trip_duration <- as.numeric(all_trips_2022$trip_duration)
```

(3)Removing **bad** data

```
all_trips_2022 <- all_trips_2022[!(all_trips_2022$start_station_name == "HQ QR" | all_trips_2022$trip_du
all_trips_2022 <- na.omit(all_trips_2022)
```

Now we are set for the next phase.

2.4 Analyze

2.4.1 Descriptive analysis on “trip_duration”:

```
all_trips_2022 <- read.csv("all_trips_2022_cleaned.csv")
all_trips_2022 %>%
  summarise(
    mean_duration = mean(trip_duration),
    max_duration = max(trip_duration),
    min_duration = min(trip_duration)
  )
```

```
##   mean_duration max_duration min_duration
## 1      1166.942      2486835            1
```

2.4.2 Let’s find the average ride time and amount of rides grouped by each day for members and casual users:

```
all_trips_2022$day_of_week <- ordered(all_trips_2022$day_of_week, levels=c("Monday","Tuesday", "Wednesda
all_trips_2022 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(trip_duration)) %>%
  arrange(member_casual, day_of_week)
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 casual        Monday               277649            1751.
##  2 casual        Tuesday              263706            1550.
##  3 casual        Wednesday            274339            1485.
##  4 casual        Thursday             309297            1533.
##  5 casual        Friday               334667            1683.
##  6 casual        Saturday             473130            1957.
##  7 casual        Sunday               388950            2045.
##  8 member        Monday               473305             736.
##  9 member        Tuesday              518584             728.
## 10 member        Wednesday            523836             726.
## 11 member        Thursday             532215             738.
## 12 member        Friday               467051             752.
## 13 member        Saturday             443245             848.
## 14 member        Sunday               387123             842.
```

2.4.3 Let’s identify the most popular stations:

```r
all_trips_2022 %>%
  group_by(start_station_name) %>%
  summarise(station_count=n()) %>%
  arrange(desc(station_count))
```

```
## # A tibble: 1,675 x 2
##    start_station_name                   station_count
##    <chr>                                        <int>
##  1 ""                                          833018
##  2 "Streeter Dr & Grand Ave"                    75222
##  3 "DuSable Lake Shore Dr & Monroe St"          41276
##  4 "DuSable Lake Shore Dr & North Blvd"         40087
##  5 "Michigan Ave & Oak St"                      39657
##  6 "Wells St & Concord Ln"                      37513
##  7 "Clark St & Elm St"                          35032
##  8 "Millennium Park"                            34999
##  9 "Kingsbury St & Kinzie St"                   33723
## 10 "Theater on the Lake"                        32974
## # i 1,665 more rows
```

```r
all_trips_2022 %>%
  group_by(end_station_name) %>%
  summarise(count=n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 1,693 x 2
##    end_station_name                      count
##    <chr>                                 <int>
##  1 ""                                   892512
##  2 "Streeter Dr & Grand Ave"             75371
##  3 "DuSable Lake Shore Dr & North Blvd"  42138
##  4 "DuSable Lake Shore Dr & Monroe St"   40123
##  5 "Michigan Ave & Oak St"               40123
##  6 "Wells St & Concord Ln"               37419
##  7 "Millennium Park"                     35231
##  8 "Clark St & Elm St"                   34481
##  9 "Theater on the Lake"                 32987
## 10 "Kingsbury St & Kinzie St"            32379
## # i 1,683 more rows
```

Seems like ***“Streeter Dr & Grand Ave”*** is by far the most popular station, but the data is still hard to read.

.

2.5 Share and Act

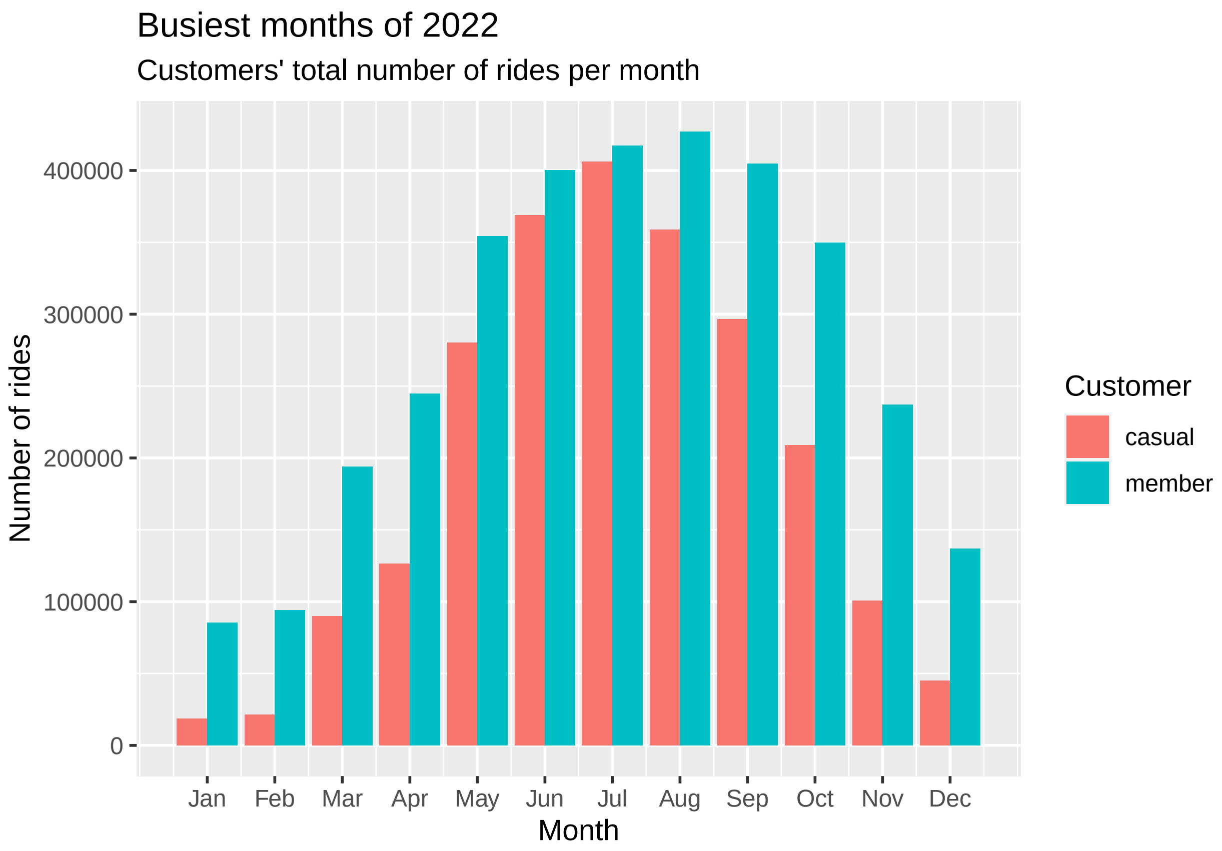
In order to share our findings in the most effective way, we will have to do it through the art of visualization, we are going to create a series of plots to tell our data story.

Each chunk of code will be followed with a plot.

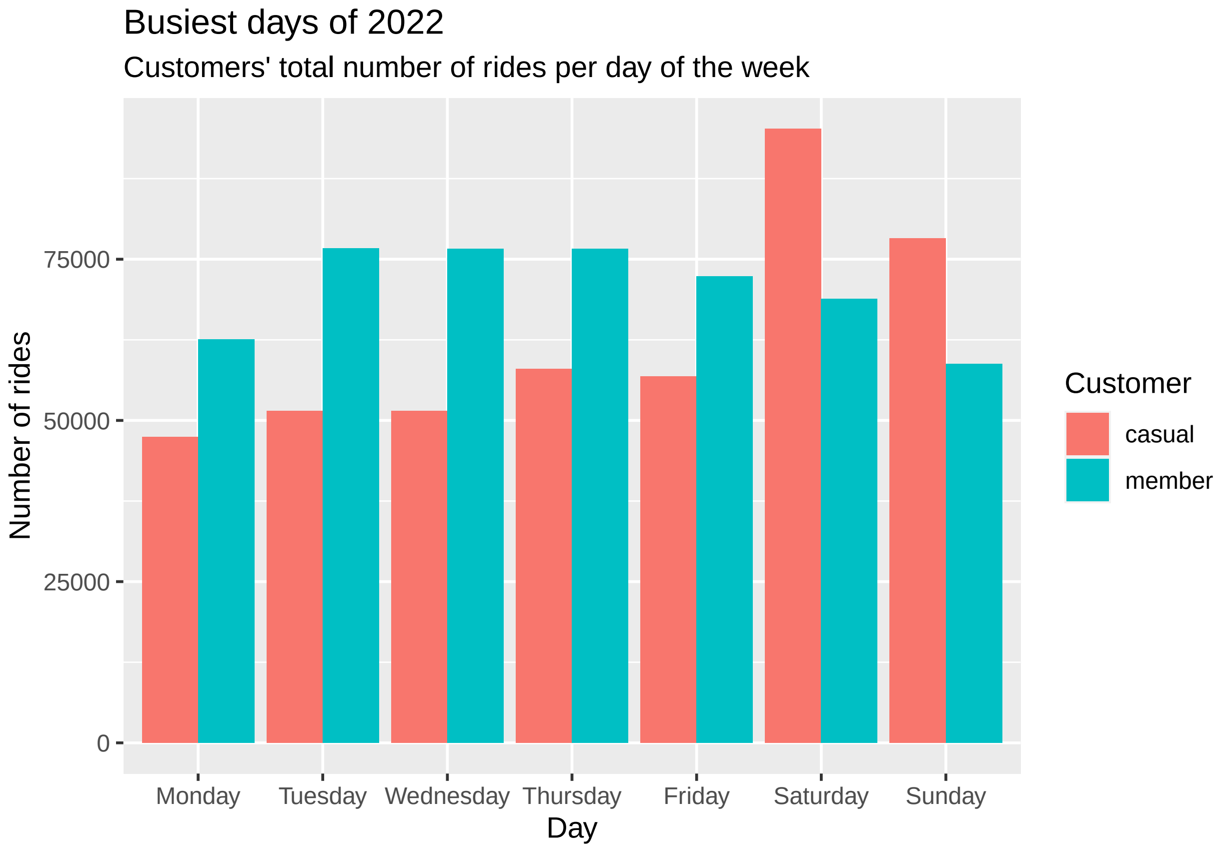
First, we will start with the total number of rides and average bike trip duration for each type of customer or busiest periods of time in short, then we will move to Customers' bike type preferences and conclude our visualizaion phase with the most popular bike stations. Let's begin:

```
options(scipen = 999)##removing scientific notations, while displaying number of rides(ex:2e+05->200000
all_trips_2022 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(),  average_duration = mean(trip_duration)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(x="Month", y="Number of rides", title = "Busiest months of 2022",subtitle = "Customers' total nu
  scale_x_continuous(breaks = 1:12,labels = month.abb[1:12])
```

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

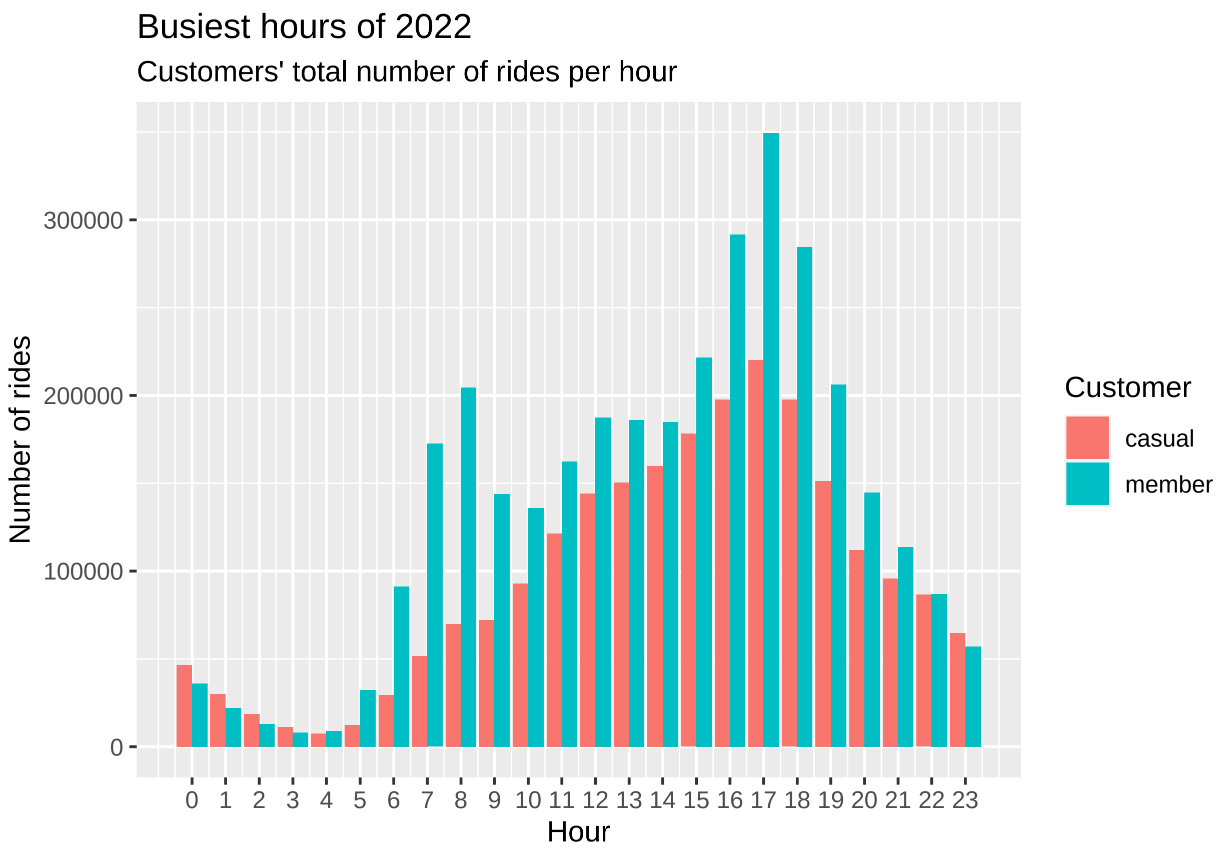


```
all_trips_2022 %>%
  group_by(member_casual, day_of_week, month) %>%
  summarise(number_of_rides = n(),  average_duration = mean(trip_duration)) %>%
  arrange(member_casual, month, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(x="Day", y="Number of rides", title = "Busiest days of 2022", subtitle = "Customers' total numbe
## `summarise()` has grouped output by 'member_casual', 'day_of_week'. You can
## override using the `.groups` argument.
```



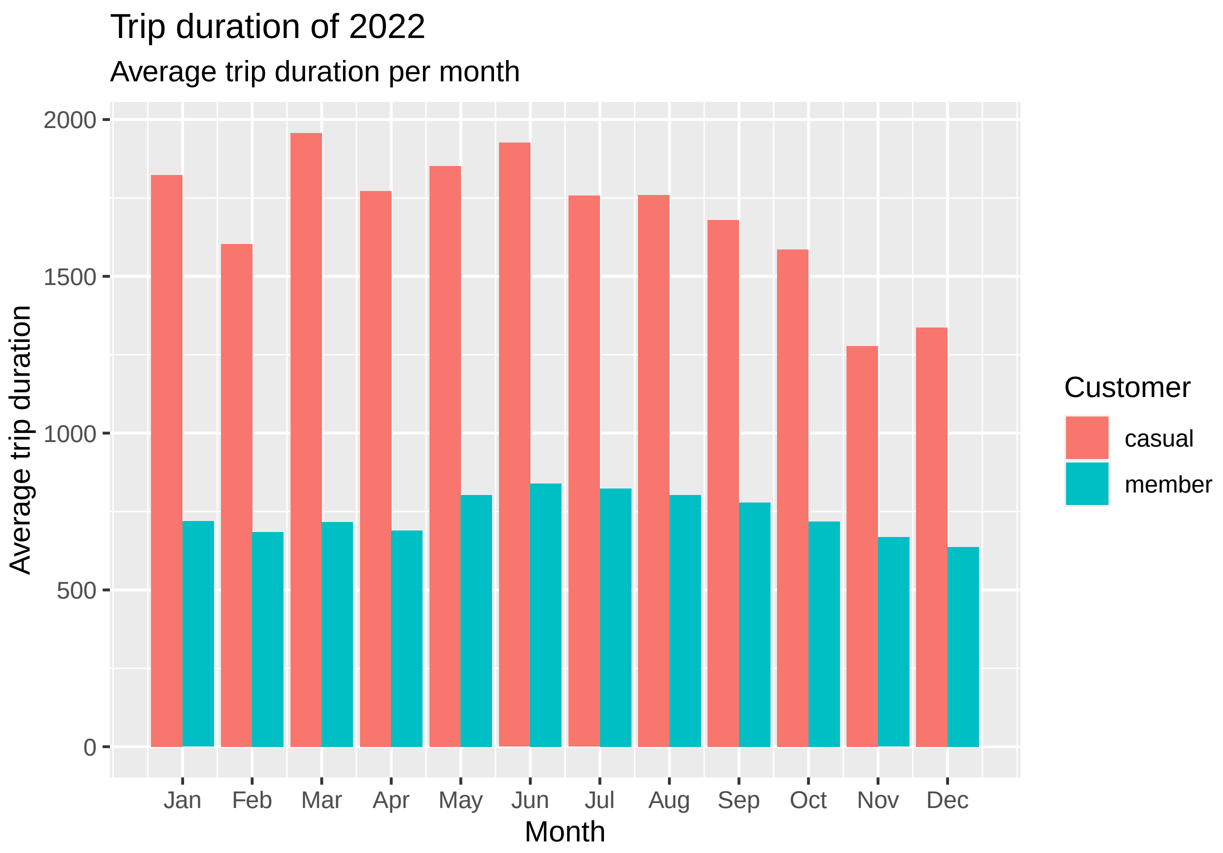
```
all_trips_2022 %>%
  group_by(member_casual, hour) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, hour) %>%
  ggplot(aes(x = hour, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(x="Hour", y="Number of rides", title = "Busiest hours of 2022",subtitle = "Customers' total numbe
  scale_x_continuous(breaks = 0:23)
```

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



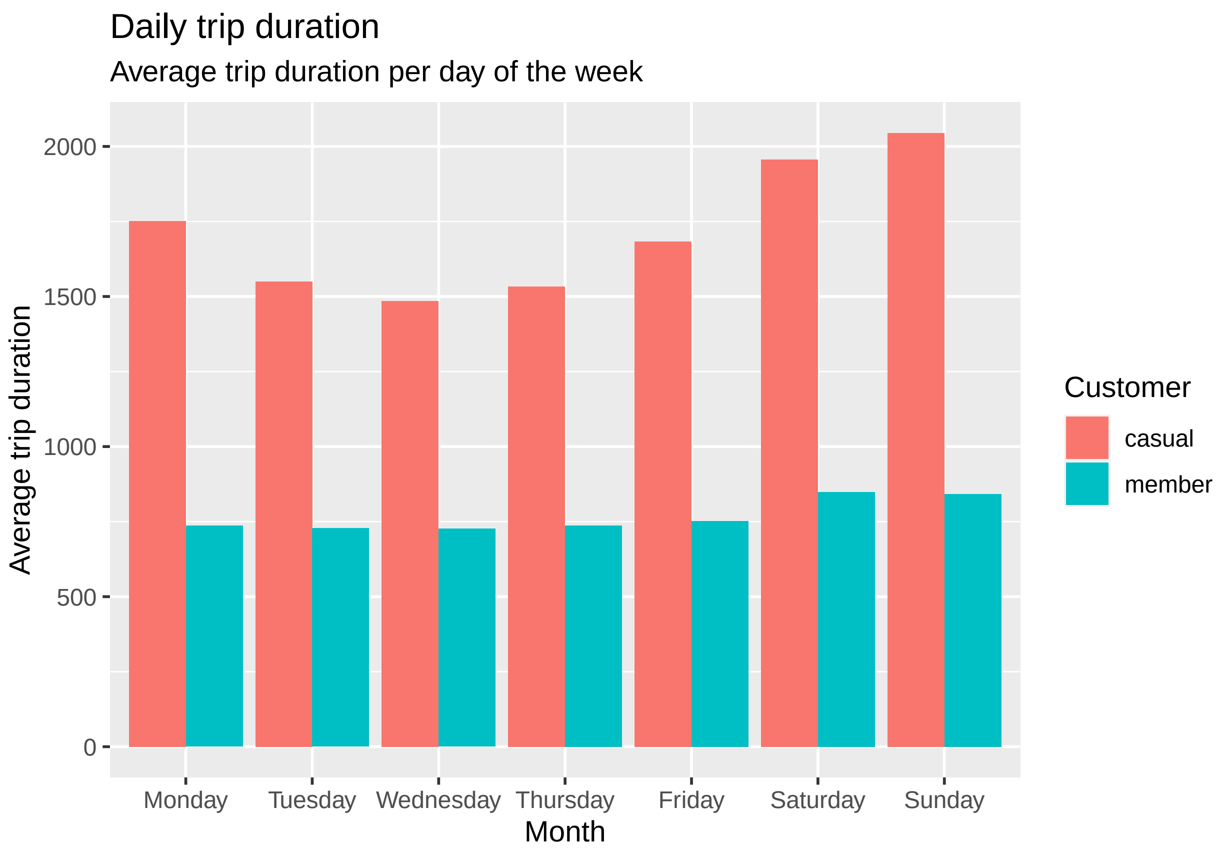
```
all_trips_2022 %>%
  group_by(member_casual, month) %>%
  summarise(average_duration = mean(trip_duration)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(x="Month", y="Average trip duration", title = "Trip duration of 2022",subtitle = "Average trip du
  scale_x_continuous(breaks = 1:12,labels = month.abb[1:12])
```

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



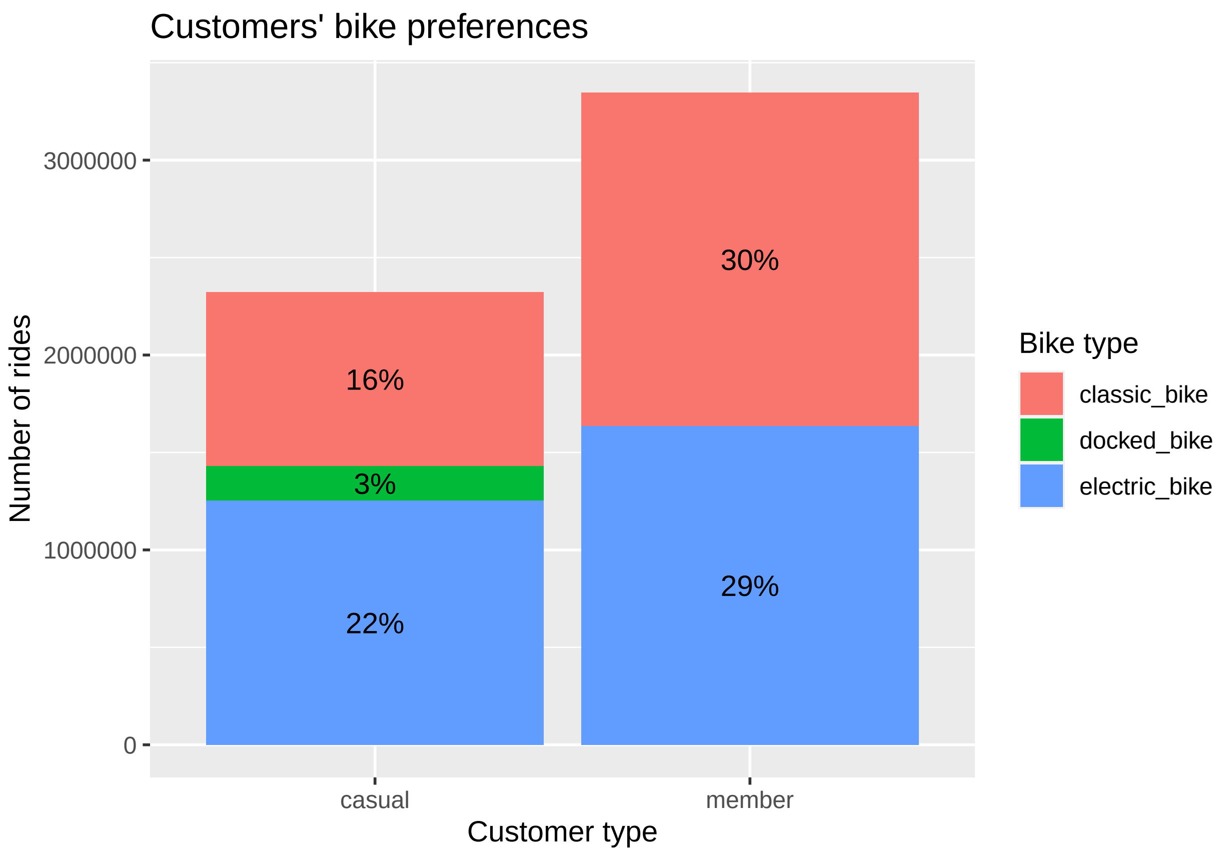
```
all_trips_2022 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_duration = mean(trip_duration)) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(x="Month", y="Average trip duration", title = "Daily trip duration ",subtitle = "Average trip du
```

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

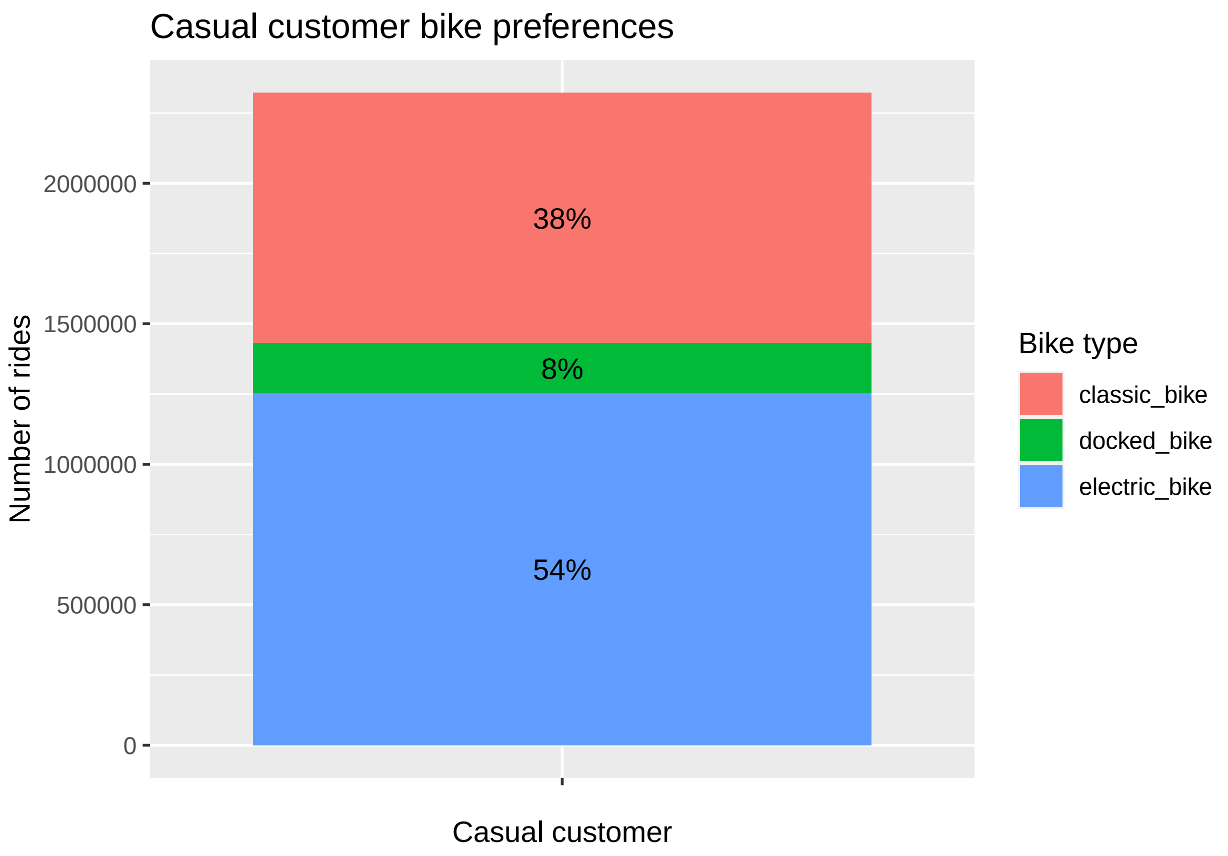


```
all_trips_2022 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x= member_casual, y=number_of_rides, fill = rideable_type)) +
  geom_col() +
  geom_text(aes(label = paste0(round((number_of_rides / sum(number_of_rides)) * 100), "%")), position =
  labs(title = "Customers' bike preferences", x="Customer type", y = "Number of rides", fill="Bike type
```

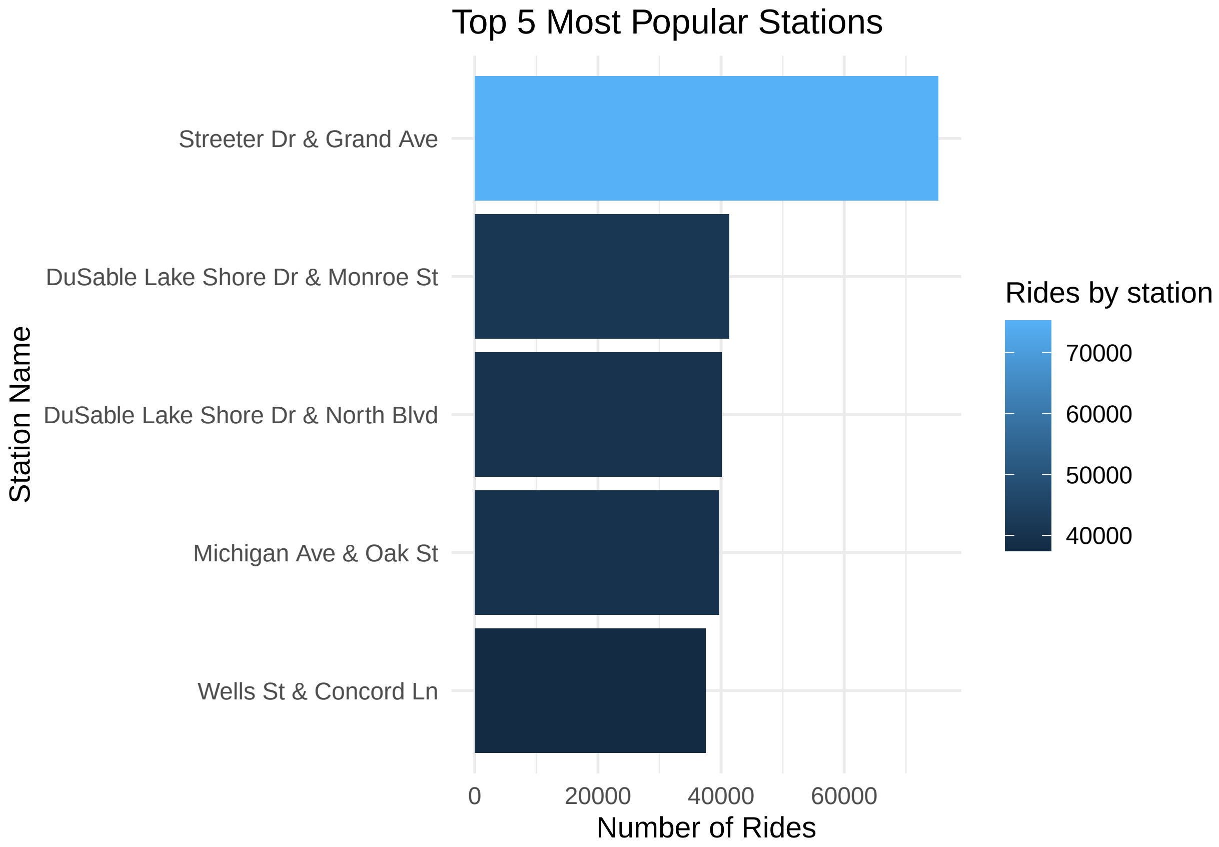
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



```
all_trips_2022 %>%
  filter(member_casual=="casual")%>%
  group_by(rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x= "", y=number_of_rides, fill = rideable_type)) +
  geom_col() +
  geom_text(aes(label = paste0(round((number_of_rides / sum(number_of_rides)) * 100), "%")), position =
  labs(title = "Casual customer bike preferences",x="Casual customer", y = "Number of rides", fill="Bik
```



```
all_trips_2022 %>%
  group_by(start_station_name) %>%
  summarise(station_count = n()) %>%
  arrange(desc(station_count)) %>%
  filter(station_count<800000)%>%
  slice_head(n = 5) %>%  # Keep only the top 5 stations
  mutate(start_station_name = reorder(start_station_name, station_count)) %>%  # Reorder based on count
  ggplot(aes(x = station_count, y = start_station_name, fill = station_count)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 5 Most Popular Stations",
       x = "Number of Rides",
       y = "Station Name", fill = "Rides by station") +
  theme_minimal()
```



3 Conclusion

3.1 Key findings from the analysis

- Period from June till August is the most active throughout the year for *casual* riders, **June being the busiest month**.
- Weekend is the most active time for *casual* riders throughout the week, **Saturday being the busiest**.
- The busiest hours for casual riders are **16-18 o'clock in the late afternoon** and **7-8 in the morning**
- The **average trip duration** for casual customers is nearly **double or even triple** that of members.
- More than half of casual customer rides were done on electric bike(~54%).
- The "Streeter Dr & Grand Ave" is by far the most popular bike station and the closest to the shore.

3.2 My top 3 suggestions based on analysis:

- Summer weekend evenings from 16-18 o'clock are the best time for marketing campaign, with the most popular stations being closer to the coast.

- Electric bikes being the most popular, I'd suggest to increase the amount of units and their availability.

- Since casual riders have the longest bike trips, my suggestion would be to initiate a campaign of offering different sorts of benefits and discounts for more distance covered by bike(free rides, minutes etc.) .