Bike Share Data Analytics Project

Ouatatchin Kone

2022-10-28

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.

Stakeholders

Lily Moreno: The director of marketing and 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. Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Executive Summary

The aim of this analysis is to focus on Cyclistic historical bike trip data to identify trends. More specifically, the analysis should help answer the following question: How do annual members and casual riders use Cyclistic bikes differently? The insights discovered will then help design a new marketing strategy to convert casual riders into annual members as the company's finance analysts have concluded that annual members are much more profitable than casual riders. This report covers different phases in my analysis to help answer the business questions raised by the management. These phases include Ask questions, Prepare data, Process data, Analyze data, Share data, and Act.

Methodology

Before performing the analysis, the data was collected through a public domain (http://www.divvy-tripdata.s3.amazonaws.com/index.html), then wrangled to make sure it's cleaned, reliable and error-free by finding and filling missing values and normalizing data. After that, I explored and gained insights through variables, proceeded to data visualization to better capture trends and insights and finally made highly recommendations to the executive team.

Ask phase

The executive team asked to analyze Cyclistic historical bike trip data as they could help design a new marketing strategy to convert more annual members for the company. More

specifically, the following business question has been raised: How do annual members and casual riders use Cyclistic bikes differently?

Prepare phase

The data to explore and analyze was made available by Motivate International Inc. through: http://www.divvy-tripdata.s3.amazonaws.com/index.html. I proceeded to the collection and storage of data by making sure they meet the requirements in terms of integrity, reliability, credibility and security. I focused my analysis on the following twelve (12) data sets: 202004-divvy-tripdata.csv 202005-divvy-tripdata.csv 202006-divvy-tripdata.csv 202007-divvy-tripdata.csv 202008-divvy-tripdata.csv 202009-divvy-tripdata.csv 202010-divvy-tripdata.csv 202011-divvy-tripdata.csv 202012-divvy-tripdata.csv 202101-divvy-tripdata.csv 202102-divvy-tripdata.csv 202103-divvy-tripdata.csv

Let's import the data sets

```
April_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202004-divvy-tripdata.csv")
May_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202005-divvy-tripdata.csv")
June_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202006-divvy-tripdata.csv")
July_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202007-divvy-tripdata.csv")
August_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202008-divvy-tripdata.csv")
September_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202009-divvy-tripdata.csv")
October_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202010-divvy-tripdata.csv")
November_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202011-divvy-tripdata.csv")
December_2020_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202012-divvy-tripdata.csv")
January_2021_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202101-divvy-tripdata.csv")
February_2021_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202102-divvy-tripdata.csv")
March_2021_Trip <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\Bike-share Data\\202103-divvy-tripdata.csv")
```

Process phase

Before analyzing data, let's load a number of packages.

```
library(tidyverse)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'T':
## please set environment variable 'TZ'
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidyr)
library(dplyr)
library(ggplot2)
library(janitor)

##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

library(lubridate)

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

library(readr)
library(skimr)
library(tibble)
library(yaml)
library(gapminder)
library(ggpubr)
library(zoo)

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

library(data.table)

##
## Attaching package: 'data.table'
##
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

Now let's find and remove any duplicates in the data sets.

```
sum(duplicated(April_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(May_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(June_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(July_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(August_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(September_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(October_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(November_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(December_2020_Trip))
```

```
[1] 0
```

```
sum(duplicated(January_2021_Trip))

[1] 0

sum(duplicated(February_2021_Trip))

[1] 0

sum(duplicated(March_2021_Trip))

[1] 0
```

No duplicates found

Here, I will search and found any missing values in my data sets.

```
colSums(is.na(April_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
99
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                 99
99
##      member_casual
##                  0

colSums(is.na(May_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
321
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                321
321
```

```
##      member_casual
##                 0

colSums(is.na(June_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
468
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                468
468
##      member_casual
##                  0

colSums(is.na(July_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                152                  0
969
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                770
770
##      member_casual
##                  0

colSums(is.na(August_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0               7691                  0
10110
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                938
938
```

```
##      member_casual
##                 0

colSums(is.na(September_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0              19901                  0
23524
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                789
789
##      member_casual
##                  0

colSums(is.na(October_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0              31405                  0
35787
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                474
474
##      member_casual
##                  0

colSums(is.na(November_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0              24434                  0
26826
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                284
284
```

```
##      member_casual
##                 0

colSums(is.na(December_2020_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
0
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                111
111
##      member_casual
##                  0

colSums(is.na(January_2021_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
0
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                103
103
##      member_casual
##                  0

colSums(is.na(February_2021_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
0
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                214
214
```

```
##      member_casual
##                 0

colSums(is.na(March_2021_Trip))

##            ride_id      rideable_type         started_at
ended_at
##                  0                  0                  0
0
## start_station_name   start_station_id   end_station_name
end_station_id
##                  0                  0                  0
0
##          start_lat          start_lng            end_lat
end_lng
##                  0                  0                167
167
##      member_casual
##                  0
```

All of the data sets contain missing values in columns end_lat and end_lng while columns start_station_id and end_station_id have some missing values but in certain data sets. To handle those missing values, I will use the method of replacing with column mean.

```
April_2020_Trip <- April_2020_Trip

for(i in 1:ncol(April_2020_Trip)) {April_2020_Trip[
,i][is.na(April_2020_Trip[ ,i])] <- mean(April_2020_Trip[ ,i], na.rm = TRUE)}



May_2020_Trip <- May_2020_Trip

for(i in 1:ncol(May_2020_Trip)) {May_2020_Trip[ ,i][is.na(May_2020_Trip[
,i])] <- mean(May_2020_Trip[ ,i], na.rm = TRUE)}



June_2020_Trip <- June_2020_Trip

for(i in 1:ncol(June_2020_Trip)) {June_2020_Trip[ ,i][is.na(June_2020_Trip[
,i])] <- mean(June_2020_Trip[ ,i], na.rm = TRUE)}



July_2020_Trip <- July_2020_Trip

for(i in 1:ncol(July_2020_Trip)) {July_2020_Trip[ ,i][is.na(July_2020_Trip[
,i])] <- mean(July_2020_Trip[ ,i], na.rm = TRUE)}
```

```
August_2020_Trip <- August_2020_Trip

for(i in 1:ncol(August_2020_Trip)) {August_2020_Trip[
,i][is.na(August_2020_Trip[ ,i])] <- mean(August_2020_Trip[ ,i], na.rm =
TRUE)}



September_2020_Trip <- September_2020_Trip

for(i in 1:ncol(September_2020_Trip)) {September_2020_Trip[
,i][is.na(September_2020_Trip[ ,i])] <- mean(September_2020_Trip[ ,i], na.rm
= TRUE)}



October_2020_Trip <- October_2020_Trip

for(i in 1:ncol(October_2020_Trip)) {October_2020_Trip[
,i][is.na(October_2020_Trip[ ,i])] <- mean(October_2020_Trip[ ,i], na.rm =
TRUE)}



November_2020_Trip <- November_2020_Trip

for(i in 1:ncol(November_2020_Trip)) {November_2020_Trip[
,i][is.na(November_2020_Trip[ ,i])] <- mean(November_2020_Trip[ ,i], na.rm =
TRUE)}



December_2020_Trip <- December_2020_Trip

for(i in 1:ncol(December_2020_Trip)) {December_2020_Trip[
,i][is.na(December_2020_Trip[ ,i])] <- mean(December_2020_Trip[ ,i], na.rm =
TRUE)}



January_2021_Trip <- January_2021_Trip

for(i in 1:ncol(January_2021_Trip)) {January_2021_Trip[
,i][is.na(January_2021_Trip[ ,i])] <- mean(January_2021_Trip[ ,i], na.rm =
TRUE)}



February_2021_Trip <- February_2021_Trip

for(i in 1:ncol(February_2021_Trip)) {February_2021_Trip[
,i][is.na(February_2021_Trip[ ,i])] <- mean(February_2021_Trip[ ,i], na.rm =
TRUE)}
```

```
March_2021_Trip <- March_2021_Trip

for(i in 1:ncol(March_2021_Trip)) {March_2021_Trip[
,i][is.na(March_2021_Trip[ ,i])] <- mean(March_2021_Trip[ ,i], na.rm = TRUE)}
```

In this section, I will create a column called “ride_length” and calculate the length of each ride in seconds for the two user types: casual riders and annual members.

```
April_2020_Trip$ride_length <-
difftime(April_2020_Trip$ended_at,April_2020_Trip$started_at)

May_2020_Trip$ride_length <-
difftime(May_2020_Trip$ended_at,May_2020_Trip$started_at)

June_2020_Trip$ride_length <-
difftime(June_2020_Trip$ended_at,June_2020_Trip$started_at)

July_2020_Trip$ride_length <-
difftime(July_2020_Trip$ended_at,July_2020_Trip$started_at)

August_2020_Trip$ride_length <-
difftime(August_2020_Trip$ended_at,August_2020_Trip$started_at)

September_2020_Trip$ride_length <-
difftime(September_2020_Trip$ended_at,September_2020_Trip$started_at)

October_2020_Trip$ride_length <-
difftime(October_2020_Trip$ended_at,October_2020_Trip$started_at)

November_2020_Trip$ride_length <-
difftime(November_2020_Trip$ended_at,November_2020_Trip$started_at)

December_2020_Trip$ride_length <-
difftime(December_2020_Trip$ended_at,December_2020_Trip$started_at)

January_2021_Trip$ride_length <-
difftime(January_2021_Trip$ended_at,January_2021_Trip$started_at)

February_2021_Trip$ride_length <-
difftime(February_2021_Trip$ended_at,February_2021_Trip$started_at)

March_2021_Trip$ride_length <-
difftime(March_2021_Trip$ended_at,March_2021_Trip$started_at)
```

Here I will create a column called “day_of_week” that will help us understand how casual riders and annual members use the company’s bikes each day of the week.

```
April_2020_Trip <- April_2020_Trip
April_2020_Trip$day_of_week<- weekdays(as.Date(April_2020_Trip$started_at))

May_2020_Trip <- May_2020_Trip
May_2020_Trip$day_of_week<- weekdays(as.Date(May_2020_Trip$started_at))

June_2020_Trip <- June_2020_Trip
June_2020_Trip$day_of_week<- weekdays(as.Date(June_2020_Trip$started_at))

July_2020_Trip <- July_2020_Trip
July_2020_Trip$day_of_week<- weekdays(as.Date(July_2020_Trip$started_at))

August_2020_Trip <- August_2020_Trip
August_2020_Trip$day_of_week<- weekdays(as.Date(August_2020_Trip$started_at))

September_2020_Trip <- September_2020_Trip
September_2020_Trip$day_of_week<-
weekdays(as.Date(September_2020_Trip$started_at))

October_2020_Trip <- October_2020_Trip
October_2020_Trip$day_of_week<-
weekdays(as.Date(October_2020_Trip$started_at))

November_2020_Trip <- November_2020_Trip
November_2020_Trip$day_of_week<-
weekdays(as.Date(November_2020_Trip$started_at))

December_2020_Trip <- December_2020_Trip
December_2020_Trip$day_of_week<-
weekdays(as.Date(December_2020_Trip$started_at))

January_2021_Trip <- January_2021_Trip
January_2021_Trip$day_of_week<-
weekdays(as.Date(January_2021_Trip$started_at))

February_2021_Trip <- February_2021_Trip
February_2021_Trip$day_of_week<-
weekdays(as.Date(February_2021_Trip$started_at))

March_2021_Trip <- March_2021_Trip
March_2021_Trip$day_of_week<- weekdays(as.Date(March_2021_Trip$started_at))
```

Here, I found negative ride lengths which are inconsistent for my analysis. Then, I will remove them to avoid any problems with my calculations.

```
April_2020_Trip_V2 <- April_2020_Trip[!(April_2020_Trip$ride_length<0),]

May_2020_Trip_V2 <- May_2020_Trip[!(May_2020_Trip$ride_length<0),]

June_2020_Trip_V2 <- June_2020_Trip[!(June_2020_Trip$ride_length<0),]

July_2020_Trip_V2 <- July_2020_Trip[!(July_2020_Trip$ride_length<0),]

August_2020_Trip_V2 <- August_2020_Trip[!(August_2020_Trip$ride_length<0),]

September_2020_Trip_V2 <-
September_2020_Trip[!(September_2020_Trip$ride_length<0),]

October_2020_Trip_V2 <-
October_2020_Trip[!(October_2020_Trip$ride_length<0),]

November_2020_Trip_V2 <-
November_2020_Trip[!(November_2020_Trip$ride_length<0),]

December_2020_Trip_V2 <-
December_2020_Trip[!(December_2020_Trip$ride_length<0),]

January_2021_Trip_V2 <-
January_2021_Trip[!(January_2021_Trip$ride_length<0),]

February_2021_Trip_V2 <-
February_2021_Trip[!(February_2021_Trip$ride_length<0),]

March_2021_Trip_V2 <- March_2021_Trip[!(March_2021_Trip$ride_length<0),]
```

Now to make sure that merging the twelve data sets into one-year file will run smoothly, I will convert column “ride_length” from factor to numeric.

```
is.difftime(April_2020_Trip_V2$ride_length)
April_2020_Trip_V2$ride_length <-
as.numeric(as.character(April_2020_Trip_V2$ride_length))
is.numeric(April_2020_Trip_V2$ride_length)


is.difftime(May_2020_Trip_V2$ride_length)
May_2020_Trip_V2$ride_length <-
as.numeric(as.character(May_2020_Trip_V2$ride_length))
is.numeric(May_2020_Trip_V2$ride_length)
```

```
is.difftime(June_2020_Trip_V2$ride_length)
June_2020_Trip_V2$ride_length<-
as.numeric(as.character(June_2020_Trip_V2$ride_length))
is.numeric(June_2020_Trip_V2$ride_length)


is.difftime(July_2020_Trip_V2$ride_length)
July_2020_Trip_V2$ride_length <-
as.numeric(as.character(July_2020_Trip_V2$ride_length))
is.numeric(July_2020_Trip_V2$ride_length)


is.difftime(August_2020_Trip_V2$ride_length)
August_2020_Trip_V2$ride_length <-
as.numeric(as.character(August_2020_Trip_V2$ride_length))
is.numeric(August_2020_Trip_V2$ride_length)


is.difftime(September_2020_Trip_V2$ride_length)
September_2020_Trip_V2$ride_length <-
as.numeric(as.character(September_2020_Trip_V2$ride_length))
is.numeric(September_2020_Trip_V2$ride_length)


is.difftime(October_2020_Trip_V2$ride_length)
October_2020_Trip_V2$ride_length <-
as.numeric(as.character(October_2020_Trip_V2$ride_length))
is.numeric(October_2020_Trip_V2$ride_length)


is.difftime(November_2020_Trip_V2$ride_length)
November_2020_Trip_V2$ride_length <-
as.numeric(as.character(November_2020_Trip_V2$ride_length))
is.numeric(November_2020_Trip_V2$ride_length)


is.difftime(December_2020_Trip_V2$ride_length)
December_2020_Trip_V2$ride_length <-
as.numeric(as.character(December_2020_Trip_V2$ride_length))
is.numeric(December_2020_Trip_V2$ride_length)


is.difftime(January_2021_Trip_V2$ride_length)
January_2021_Trip_V2$ride_length <-
as.numeric(as.character(January_2021_Trip_V2$ride_length))
is.numeric(January_2021_Trip_V2$ride_length)
```

```
is.difftime(February_2021_Trip_V2$ride_length)
February_2021_Trip_V2$ride_length <-
as.numeric(as.character(February_2021_Trip_V2$ride_length))
is.numeric(February_2021_Trip_V2$ride_length)


is.difftime(March_2021_Trip_V2$ride_length)
March_2021_Trip_V2$ride_length <-
as.numeric(as.character(March_2021_Trip_V2$ride_length))
is.numeric(March_2021_Trip_V2$ride_length)
```

Now let's convert the columns "start_station_name", "end_station_name", "start_station_id", "end_station_id" to character so that they can merge correctly

```
April_2020_Trip_V2 <- mutate(April_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

May_2020_Trip_V2 <- mutate(May_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

June_2020_Trip_V2 <- mutate(June_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

July_2020_Trip_V2 <- mutate(July_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

August_2020_Trip_V2 <- mutate(August_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
```

```
as.character(start_station_id), end_station_id =
as.character(end_station_id))

September_2020_Trip_V2 <- mutate(September_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

October_2020_Trip_V2 <- mutate(October_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

November_2020_Trip_V2 <- mutate(November_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

December_2020_Trip_V2 <- mutate(December_2020_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

January_2021_Trip_V2 <- mutate(January_2021_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

February_2021_Trip_V2 <- mutate(February_2021_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
as.character(start_station_id), end_station_id =
as.character(end_station_id))

March_2021_Trip_V2 <- mutate(March_2021_Trip_V2, start_station_name =
as.character(start_station_name)
                                ,end_station_name =
as.character(end_station_name), start_station_id =
```

```
as.character(start_station_id), end_station_id =
as.character(end_station_id))
```

Data has been wrangled and manipulated to ensure they are cleaned and ready for analysis. Now it is time to merge them into one-year data frame.

```
one_year_trips <-
bind_rows(April_2020_Trip_V2,May_2020_Trip_V2,June_2020_Trip_V2,July_2020_Tri
p_V2, August_2020_Trip_V2, September_2020_Trip_V2, October_2020_Trip_V2,
November_2020_Trip_V2,December_2020_Trip_V2, January_2021_Trip_V2,
February_2021_Trip_V2, March_2021_Trip_V2)

str(one_year_trips)

## 'data.frame':    3479196 obs. of  15 variables:
##  $ ride_id           : chr  "A847FADBBC638E45" "5405B80E996FF60D"
"5DD24A79A4E006F4" "2A59BBDF5CDBA725" ...
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike"
"docked_bike" ...
##  $ started_at        : chr  "2020-04-26 17:45:14" "2020-04-17 17:08:54"
"2020-04-01 17:54:13" "2020-04-07 12:50:19" ...
##  $ ended_at          : chr  "2020-04-26 18:12:03" "2020-04-17 17:17:03"
"2020-04-01 18:08:36" "2020-04-07 13:02:31" ...
##  $ start_station_name: chr  "Eckhart Park" "Drake Ave & Fullerton Ave"
"McClurg Ct & Erie St" "California Ave & Division St" ...
##  $ start_station_id  : chr  "86" "503" "142" "216" ...
##  $ end_station_name  : chr  "Lincoln Ave & Diversey Pkwy" "Kosciuszko
Park" "Indiana Ave & Roosevelt Rd" "Wood St & Augusta Blvd" ...
##  $ end_station_id    : chr  "152" "499" "255" "657" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 42 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ ride_length       : num  1609 489 863 732 3175 ...
##  $ day_of_week       : chr  "Sunday" "Friday" "Wednesday" "Tuesday" ...

summary(one_year_trips)

##    ride_id          rideable_type       started_at          ended_at
##  Length:3479196     Length:3479196     Length:3479196     Length:3479196
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:3479196     Length:3479196     Length:3479196     Length:3479196
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
```

```
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.87   Min.   :41.54   Min.   :-88.07
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.64   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.08   Max.   :-87.52   Max.   :42.16   Max.   :-87.44
##  member_casual       ride_length      day_of_week
##  Length:3479196     Min.   :      0   Length:3479196
##  Class :character   1st Qu.:    476   Class :character
##  Mode  :character   Median :    874   Mode  :character
##                     Mean   :   1677
##                     3rd Qu.:   1601
##                     Max.   :3523202
```

Analysis Phase

Recalling that the aim of our analysis is to understand how annual members and casual riders use Cyclistic bikes differently. That is why my analysis will specifically focus on the following variables: ride_length, day_of_week, member_casual, and rideable_type.

```
summary(one_year_trips$ride_length)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     476     874    1677    1601 3523202

table(one_year_trips$day_of_week)

##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##    514949    418505    658179    527570    466855    429639    463499

table(one_year_trips$rideable_type)

##
##  classic_bike   docked_bike electric_bike
##        319871       2548128        611197

table(one_year_trips$member_casual)

##
##  casual  member
## 1427121 2052075
```

From the previous output, we notice that: * Riders (casuals and members) spend on average 1677 seconds riding between two (2) stations. * Docked bike is the most popular bike type for a total of 2,548,128 trips, followed by electric bike (611,197 trips) and classic bike (319,871 trips) * Weekends have the most important traffic with Saturday (658,179

trips) and Sunday (527,570 trips) * Annual members have the most important trips (2,052,075) compared to casual riders (1,427,121 trips)

In this section, I conduct thorough analysis to compare annual members and casual riders.

```
number_of_rides_and_average_duration_per_usertype <- one_year_trips %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n() ,average_duration =  mean(ride_length)) %>%
  arrange(member_casual, day_of_week)
```

This allows us to gain insights on how Cyclistic's clients (members and casuals) use the company's bikes differently in terms of weekdays, number of rides and average ride length. For example, casual riders are mostly active on Saturday, and Sunday for respectively 335,059 and 262,271 rides and spend on average 2817.35 and 3044.58 seconds between two (2) stations.

When it comes to annual members, they are mostly active on Friday (306,398 trips), and Saturday (323,120 trips), and spend respectively 920.22 and 1067.61 seconds between two (2) stations.

```
number_of_rides_and_biketype_per_usertype <- one_year_trips %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n() ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, rideable_type)
```

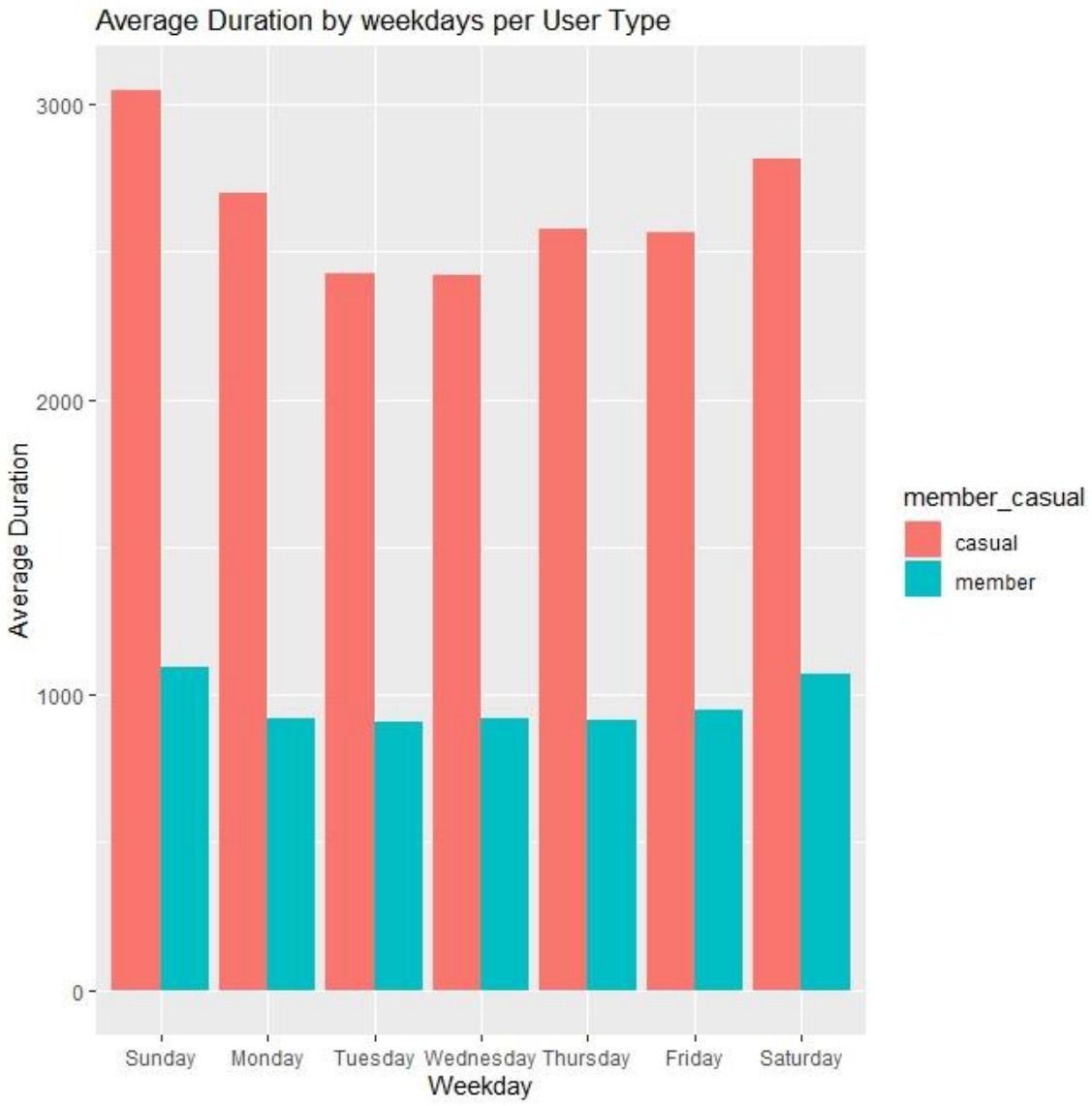
Here I found that casual riders and annual members have the same preference (counting the number of rides for each bike type) in terms of bike type used for their trips. Indeed, docked bike is most popular, followed by electric bike, and classic bike.

Visualization Phase

Here I will go through a series of visualizations which come from previous data sets created in the analysis. For each visualization, I will go deeper in my analysis and explain the trends found.

For this first graph, I plot average ride length against weekdays for each rider type.

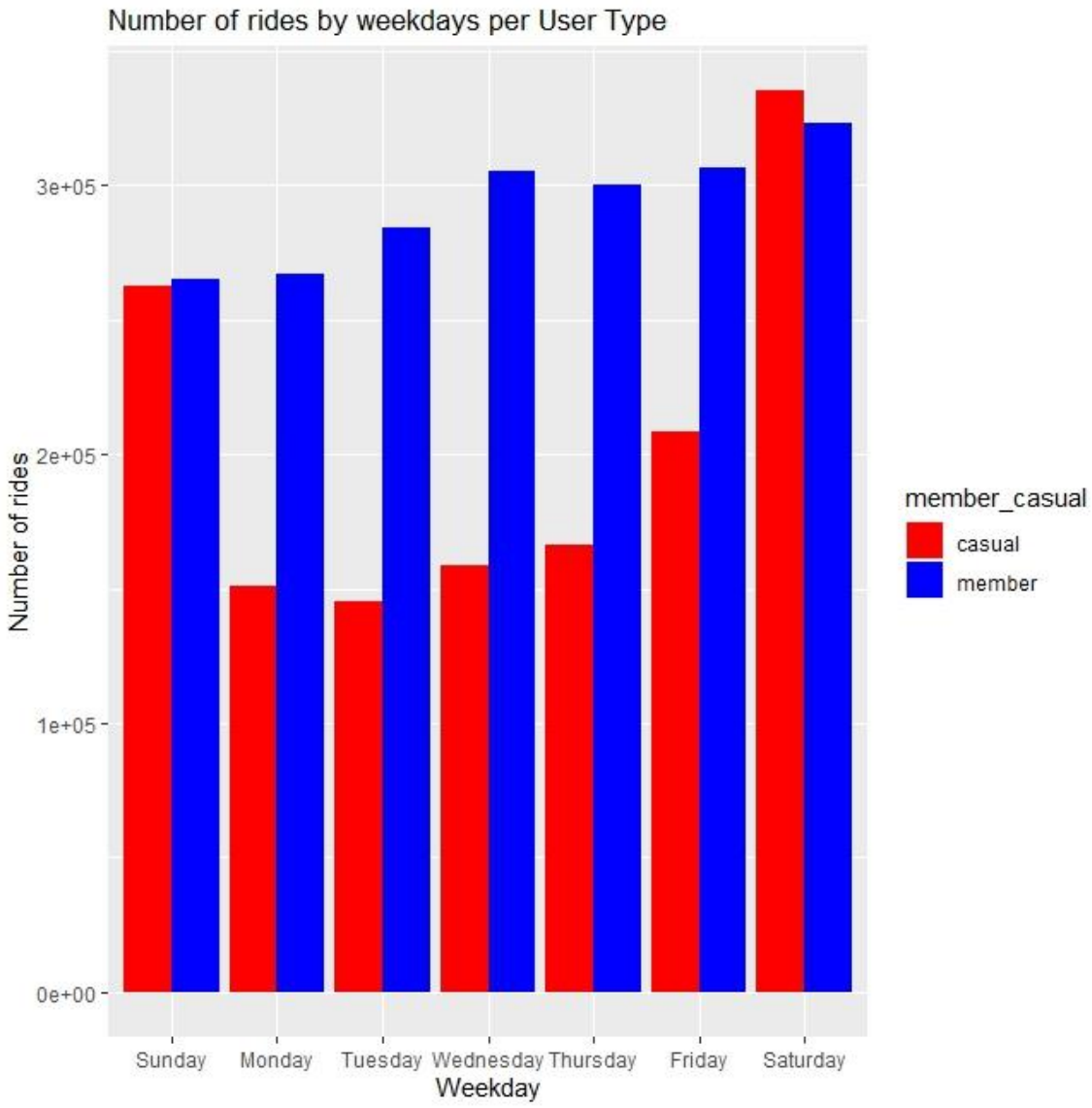
```
ggplot(data = number_of_rides_and_average_duration_per_usertype,
       aes(x = day_of_week, y = average_duration, fill = member_casual)) +
geom_col(position = "dodge") +
scale_color_manual(values = colors)+
labs(title = "Average Duration by weekdays per User Type", x = "Weekday", y =
"Average Duration")
```



We notice that in terms of ride duration, casual riders and annual members are mostly active at weekends. However, casual riders have on average the highest ride lengths compared to annual members. Specifically, casual riders spend on average 3044 seconds on Sunday (against 1093 seconds for annual members) and 2817 seconds on Saturday (against 1067 seconds for annual members).

Here I plot number of rides against weekdays for each rider type

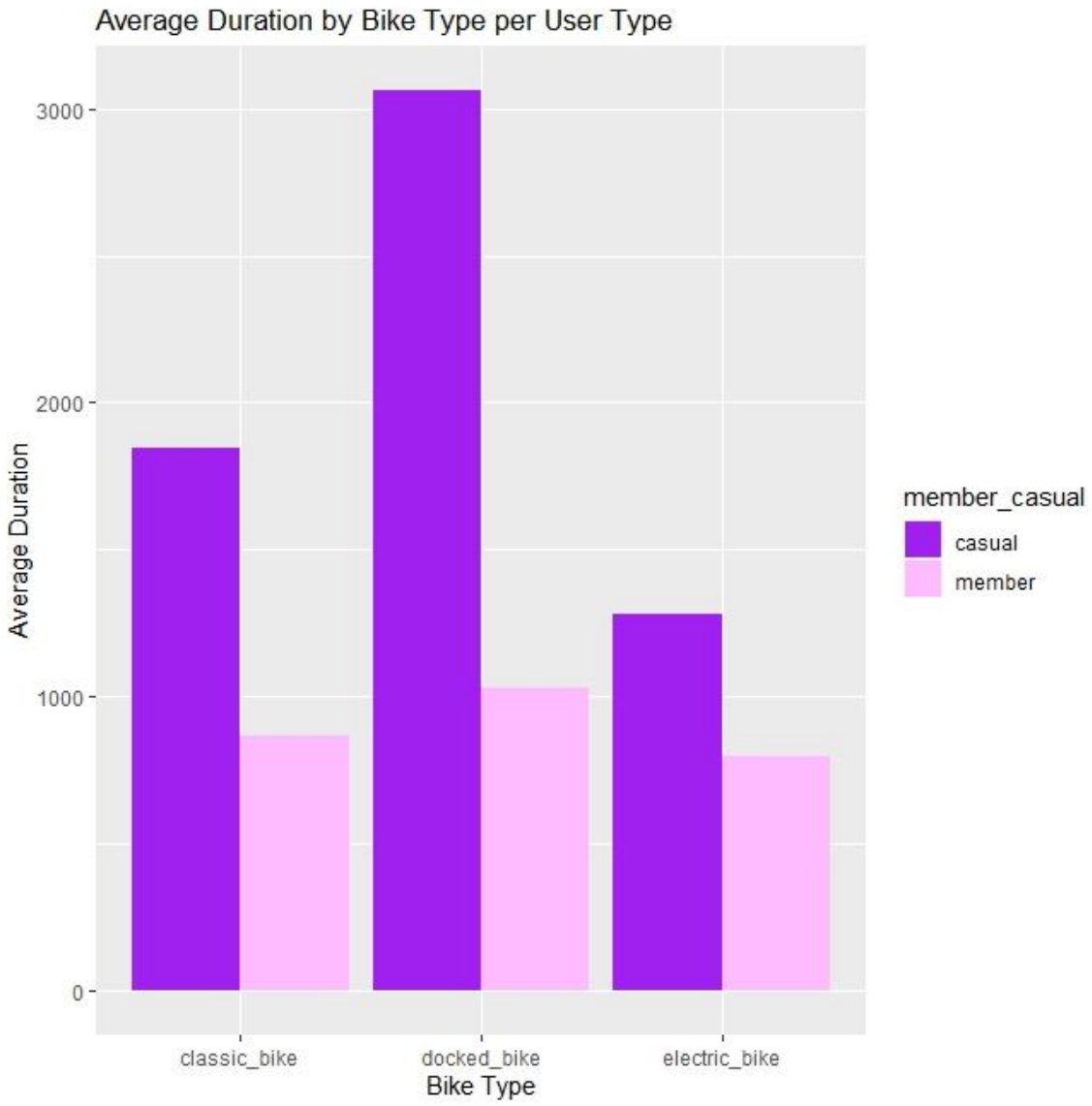
```
ggplot(data = number_of_rides_and_average_duration_per_usertype,
       aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
geom_col(position = "dodge") +
scale_fill_manual(values = c("red", "blue"))+
labs(title = "Number of rides by weekdays per User Type", x = "Weekday", y =
"Number of rides")
```



Compared to the previous section, here, annual members have the highest number of rides compared to casual riders. Specifically, they are mostly active on Saturday (323,120 trips) and Friday (306,398 trips), When it comes to casual riders they are mostly active on Saturday (335,059 trips), and Sunday (262,271 trips).

Here, I plot Average Duration against bike type for each rider type.

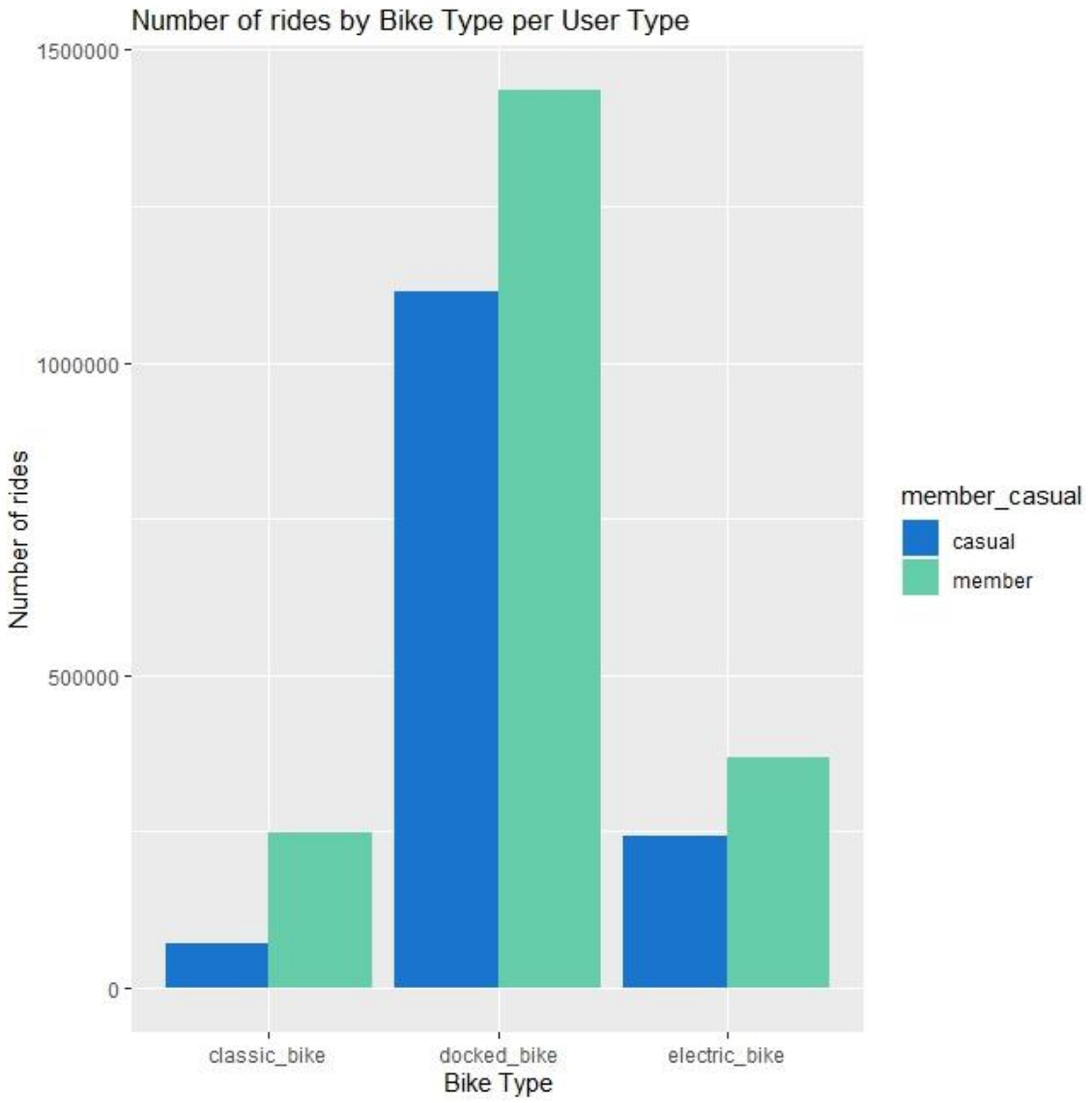
```
ggplot(data = number_of_rides_and_biketype_per_usertype,
       aes(x = rideable_type, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("purple", "plum1"))+
  labs(title = "Average Duration by Bike Type per User Type", x = "Bike
Type",
       y = "Average Duration")
```



We notice that although having the same preference for docked bikes, casual riders and annual members differ in terms of average ride length using that bike type. Indeed, casual riders have the highest duration (3061 seconds) compared to annual members (1027 seconds).

Here, I plot number of rides against bike type for each rider type.

```
ggplot(data = number_of_rides_and_biketype_per_usertype,
       aes(x = rideable_type, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("dodgerblue3", "mediumaquamarine"))+
  labs(title = "Number of rides by Bike Type per User Type", x = "Bike Type",
y = "Number of rides")
```



By contrast, here we can see that annual members have the highest number of rides (1,434,720) using docked bikes compared to casual riders (1,113,408).

Analysis Summary

- In general, annual members have the highest number of rides (2,052,075) compared to casual riders (1,427,121) while casual riders have the highest average ride length (18552.74 seconds) compared to annual members (6768.31 seconds).
- Annual members and casual riders are mostly active at weekends and differ in terms of number of trips on Saturday: casual riders (335,059 trips) and annual members (323,120 trips); average ride length on Sunday: casual riders (3044 seconds) and annual members (1093 seconds).
- Docked bike is most popular and used by annual members and casual riders. However, they differ in terms of highest ride length (3061 seconds for casual riders

against 1027 seconds for annual members) and number of rides (1,113,408 rides for casual riders and 1,434,720 rides for annual members and).

Act phase

High-level recommendations to the executive team

Based on previous analysis, the executive team should apply the following recommendations to design a new marketing strategy to convert casual riders into annual members:

- As casual riders have the highest ride lengths but least number of rides, this shows that they are interested in using the bikes for long time. Then, the marketing team should propose them a marketing offer which will include an attractive annual membership at affordable price compared to their single-ride passes and full-day passes.
- Design a special annual marketing offer which is affordable in terms of transport costs for casual riders that use the bikes to commute to work each day.
- As casual riders have a particular interest for docked bikes, then design a special package exclusively for docked bikes to convert them into annual members with a bonus of bringing a family member who will benefit of a three-month subscription for free.