cyclistic_data_analysis

Joan Kusuma

2022-04-28

Capstone Project

Case Study: How Does a Bike-Share Navigate Speedy Success? This is a Capstone Project completed for the Google Data Analytics Professional Certificate offered by Google.

To access the case study, click this link

Ask

- **business task**

We need to answer the following 3 questions, which will guide the future marketing program:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual membership?
3. How can Cyclistic use digital media to influence casual riders to become members?

- **Key stakeholders**

Primary Stakeholder: Lily Moreno, the Marketing Director, and the Cyclistic Executive Team

Secondary Stakeholder: Cyclistic marketing analytics team

- **A clear statement of the business task**

Identify and analyze key differences between casual riders and annual members to create an effective marketing campaign that will convert casual riders into annual members

Prepare

- **Where is your data located?**

the data is located here https://divvy-tripdata.s3.amazonaws.com/index.html

- **How is the data organized?**

the data is organized in a two-dimensional table format, with rows and columns in a csv file by year and month, and then compressed into a zip file.

- **Are there issues with bias or credibility in this data? Does your data ROCCC?**

Data is collected by historical trip data provided by Motivate International Inc, owned by the city of Chicago, available to the public under this license(https://ride.divvybikes.com/data-license-agreement)

- **How are you addressing licensing, privacy, security, and accessibility?**

License grants you a non-exclusive, royalty-free, limited, perpetual license to access, reproduce, analyze, copy, modify, distribute in your product or service and use the Data for any lawful purpose

- **How did you verify the data's integrity?**

By making sure that data is complete, accurate, consistent, and valid. For this purpose, I made sure this data is in compliance with the data-cleaning verfication checklist from Chapter 4 of this course (Process Data from Dirty to Clean). These include:

- data is sourced from a legible source.
- data contain missing values but since we have enough data to omit the wrong data, we may perform the analysis without the missing values.
- addresses in data are identified as correct when compared to actual address
- date and time on some columns were modified when imported on Excel, but I was able to format it to the correct date-time standard.
- **How does it help you answer your question?**

data contains ride time and bike type from each membership type, which may help in calculation of their respective activities.

- **Are there any problems with the data?**

Data contains some missing values, incorrect data format during data import which was corrected, and ride length contains 0.

Process

Guiding Questions

- **What tools are you choosing and why?**

I am choosing Excel Spreadsheet and RStudio IDE. Excel spreadsheet to do simple calculation and data cleaning, and RStudio to aggregate data as the data is too big to be done on Excel alone. Data analysis will be done in RStudio.

- **Have you ensured your data's integrity?**

yes, see Prepare section, also more data cleaning once data has been aggregated as can be seen on the steps taken in RStudio below

- **What steps have you taken to ensure that your data is clean?**

(see Process and Prepare section)

- **How can you verify that your data is clean and ready to analyze?**

By running them through the data-cleaning checklist (available on Chapter 4: Process Data from Dirty to Clean)

- **Have you documented your cleaning process so you can review and share those results?**

Yes.

Deliverables

- **Documentation of any cleaning or manipulation of data**

Processing starts by having separate folders to store the original files and edited files.

The next step include cleaning and manipulating data on an Excel spreadsheet. For the purpose of this project, I used the data from the last twelve months from March 2021 to March 2022 for relevance.

Data cleaning and manipulation done on Excel include the following:

- formatting the columns "started_at" and "ended_at" from general into date-time format (yyyy-mm-dd hh:mm:ss)

- adding the column “ride_length” to calculate the length of each ride. Calculation is done by subtracting the “started_at” column from the “ended_at” column
- formatting the column “ride_length” as time format (hh:mm:ss)
- adding the column “day_of_week” to configure the day of the week that each ride started based on the “started_at” column
- adding the column “ride_time” to configure the most popular time to ride a bike based on the “started_at” column
- changing several column names to make data easier to understand, such as:

1. “rideable_type” to “bike_type”
2. “started_at” to “start_time”
3. “ended_at” to “end_time”
4. “member_casual” to “user_type”

Data processing then continue on RStudio, the following steps were done during this stage:

- **Install necessary packages to start this project**

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

- **set working directory to where file is located, then import the files in the directory**

```
setwd("/Users/joankusuma/working_directory/capstone_1/trip_data")
tripdata_202104 <- read_csv("202104.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 337230 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202105 <- read_csv("202105.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 531633 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202106 <- read_csv("202106.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 729595 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202107 <- read_csv("202107.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 822410 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202108 <- read_csv("202108.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 804352 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202109 <- read_csv("202109.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 756147 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202110 <- read_csv("202110.csv")
```

```
## Rows: 631226 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202111 <- read_csv("202111.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 359978 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202112 <- read_csv("202112.csv")
```

```
## Rows: 247540 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202201 <- read_csv("202201.csv")
```

```
## Rows: 103770 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202202 <- read_csv("202202.csv")
```

```
## Rows: 115609 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

```
tripdata_202203 <- read_csv("202203.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 284042 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): ride_id, bike_type, start_station_name, start_station_id, end_stat...
## dbl  (1): day_of_week
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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.
```

- **To make sure that data will combine successfully, compare_df_cols will be used**

```
compare_df_cols(tripdata_202104, tripdata_202105, tripdata_202106,
                tripdata_202107, tripdata_202108, tripdata_202109,
                tripdata_202110, tripdata_202111, tripdata_202112,
```

```
               tripdata_202201, tripdata_202202, tripdata_202203,
               return = "mismatch",
               bind_method = "bind_rows")
```

```
##  [1] column_name     tripdata_202104 tripdata_202105 tripdata_202106
##  [5] tripdata_202107 tripdata_202108 tripdata_202109 tripdata_202110
##  [9] tripdata_202111 tripdata_202112 tripdata_202201 tripdata_202202
## [13] tripdata_202203
## <0 rows> (or 0-length row.names)
```

- **After making sure that there is no mismatch and data will combine successfully, aggregate the data by seasons and year round**

```
spring_data <- bind_rows(tripdata_202203, tripdata_202104, tripdata_202105)
summer_data <- bind_rows(tripdata_202106, tripdata_202107, tripdata_202108)
fall_data <- bind_rows(tripdata_202109, tripdata_202110, tripdata_202111)
winter_data <- bind_rows(tripdata_202112, tripdata_202201, tripdata_202202)
year_round_data <- bind_rows(tripdata_202104, tripdata_202105, tripdata_202106,
                          tripdata_202107, tripdata_202108, tripdata_202109,
                          tripdata_202110, tripdata_202111, tripdata_202112,
                          tripdata_202201, tripdata_202202, tripdata_202203)
```

- **The following code is run to skim through data to check for any missing values, whitespace, misspellings, correct data format, and misleading column labels**

```
head(year_round_data)
```

```
## # A tibble: 6 x 12
##   ride_id     bike_type start_time          end_time            start_station_n~
##   <chr>       <chr>     <dttm>              <dttm>              <chr>
## 1 6C992BD37A~ classic_~ 2021-04-12 18:25:36 2021-04-12 18:56:55 State St & Pear~
## 2 1E0145613A~ docked_b~ 2021-04-27 17:27:11 2021-04-27 18:31:29 Dorchester Ave ~
## 3 E498E15508~ docked_b~ 2021-04-03 12:42:45 2021-04-07 11:40:24 Loomis Blvd & 8~
## 4 1887262AD1~ classic_~ 2021-04-17 09:17:42 2021-04-17 09:42:48 Honore St & Div~
## 5 C123548CAB~ docked_b~ 2021-04-03 12:42:25 2021-04-03 14:13:42 Loomis Blvd & 8~
## 6 097E76F365~ classic_~ 2021-04-25 18:43:18 2021-04-25 18:43:59 Clinton St & Po~
## # ... with 7 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, user_type <chr>, ride_length <time>,
## #   day_of_week <dbl>, ride_time <chr>
```

```
str(year_round_data)
```

```
## spec_tbl_df [5,723,532 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5723532] "6C992BD37A98A63F" "1E0145613A209000" "E498E15508A80BAD" "188
##  $ bike_type         : chr [1:5723532] "classic_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ start_time        : POSIXct[1:5723532], format: "2021-04-12 18:25:36" "2021-04-27 17:27:11" ...
##  $ end_time          : POSIXct[1:5723532], format: "2021-04-12 18:56:55" "2021-04-27 18:31:29" ...
##  $ start_station_name: chr [1:5723532] "State St & Pearson St" "Dorchester Ave & 49th St" "Loomis Bl
##  $ start_station_id  : chr [1:5723532] "TA1307000061" "KA1503000069" "20121" "TA1305000034" ...
##  $ end_station_name  : chr [1:5723532] "Southport Ave & Waveland Ave" "Dorchester Ave & 49th St" "Lo
##  $ end_station_id    : chr [1:5723532] "13235" "KA1503000069" "20121" "13235" ...
##  $ user_type         : chr [1:5723532] "member" "casual" "casual" "member" ...
##  $ ride_length       : 'hms' num [1:5723532] 00:31:19 01:04:18 22:57:39 00:25:06 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week       : num [1:5723532] 2 3 7 7 7 1 7 3 2 7 ...
##  $ ride_time         : chr [1:5723532] "18" "17" "12" "09" ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   bike_type = col_character(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   user_type = col_character(),
##   ..   ride_length = col_time(format = ""),
##   ..   day_of_week = col_double(),
##   ..   ride_time = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
glimpse(year_round_data)
```

```
## Rows: 5,723,532
## Columns: 12
## $ ride_id            <chr> "6C992BD37A98A63F", "1E0145613A209000", "E498E15508~
## $ bike_type          <chr> "classic_bike", "docked_bike", "docked_bike", "clas~
## $ start_time         <dttm> 2021-04-12 18:25:36, 2021-04-27 17:27:11, 2021-04-~
## $ end_time           <dttm> 2021-04-12 18:56:55, 2021-04-27 18:31:29, 2021-04-~
## $ start_station_name <chr> "State St & Pearson St", "Dorchester Ave & 49th St"~
## $ start_station_id   <chr> "TA1307000061", "KA1503000069", "20121", "TA1305000~
## $ end_station_name   <chr> "Southport Ave & Waveland Ave", "Dorchester Ave & 4~
## $ end_station_id     <chr> "13235", "KA1503000069", "20121", "13235", "20121",~
## $ user_type          <chr> "member", "casual", "casual", "member", "casual", "~
## $ ride_length        <time> 00:31:19, 01:04:18, 22:57:39, 00:25:06, 01:31:17, ~
## $ day_of_week        <dbl> 2, 3, 7, 7, 7, 1, 7, 3, 2, 7, 7, 7, 3, 1, 3, 5, 3, ~
## $ ride_time          <chr> "18", "17", "12", "09", "12", "18", "16", "16", "15~
```

```
skim_without_charts(year_round_data)
```

Table 1: Data summary

Name	year_round_data
Number of rows	5723532
Number of columns	12
Column type frequency:	
character	8
difftime	1
numeric	1
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	8	16	0	5721539	0
bike_type	0	1.00	11	13	0	3	0
start_station_name	745376	0.87	3	53	0	861	0
start_station_id	745373	0.87	2	44	0	851	0
end_station_name	796247	0.86	10	53	0	860	0
end_station_id	796247	0.86	2	44	0	851	0
user_type	0	1.00	6	6	0	2	0
ride_time	0	1.00	2	2	0	24	0

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
ride_length	145	1	0 secs	86397 secs	00:11:43	24945

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
day_of_week	0	1	4.09	2.06	1	2	4	6	7

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
start_time	0	1	2021-04-01 00:03:18	2022-03-31 23:59:47	2021-08-17 18:25:49	4793191
end_time	0	1	2021-04-01 00:14:29	2022-04-01 22:10:12	2021-08-17 18:44:32	4787114

- **further cleaning: drop na values, filter 0s, and mutate day of week into week names for ease of understanding**

```
spring_df <- spring_data%>%
  drop_na%>%
  filter(ride_length > 0)%>%
  mutate(day_of_week = recode(day_of_week,
                              "1" = "Sunday",
                              "2" = "Monday",
                              "3" = "Tuesday",
                              "4" = "Wednesday",
                              "5" = "Thursday",
                              "6" = "Friday",
                              "7" =" Saturday"))

summer_df <- summer_data%>%
  drop_na%>%
  filter(ride_length > 0)%>%
  mutate(day_of_week = recode(day_of_week,
                              "1" = "Sunday",
                              "2" = "Monday",
```

```
                                "3" = "Tuesday",
                                "4" = "Wednesday",
                                "5" = "Thursday",
                                "6" = "Friday",
                                "7" =" Saturday"))

fall_df <- fall_data%>%
  drop_na%>%
  filter(ride_length > 0)%>%
  mutate(day_of_week = recode(day_of_week,
                                "1" = "Sunday",
                                "2" = "Monday",
                                "3" = "Tuesday",
                                "4" = "Wednesday",
                                "5" = "Thursday",
                                "6" = "Friday",
                                "7" =" Saturday"))

winter_df <- winter_data%>%
  drop_na%>%
  filter(ride_length > 0)%>%
  mutate(day_of_week = recode(day_of_week,
                                "1" = "Sunday",
                                "2" = "Monday",
                                "3" = "Tuesday",
                                "4" = "Wednesday",
                                "5" = "Thursday",
                                "6" = "Friday",
                                "7" =" Saturday"))

year_round_df <- year_round_data%>%
  drop_na%>%
  filter(ride_length > 0)%>%
  mutate(day_of_week = recode(day_of_week,
                                "1" = "Sunday",
                                "2" = "Monday",
                                "3" = "Tuesday",
                                "4" = "Wednesday",
                                "5" = "Thursday",
                                "6" = "Friday",
                                "7" =" Saturday"))
```

Analyze

- **load libraries and added mode function to start analysis**

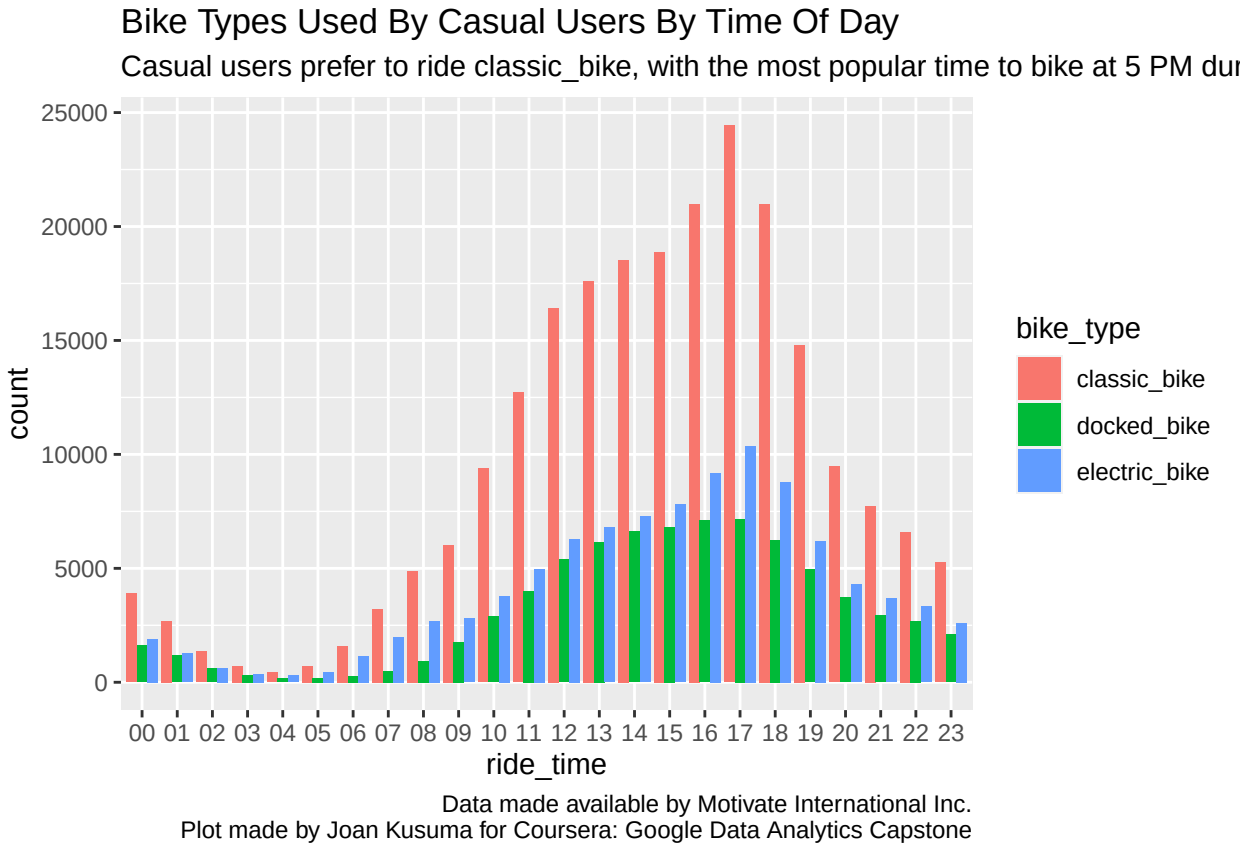
mode function available through this link

```
library(Tmisc)
library(SimDesign)
mode <- function(x) {
  unique_val <- unique(x)
  counts <- vector()
  for (i in 1: length(unique_val)) {
```

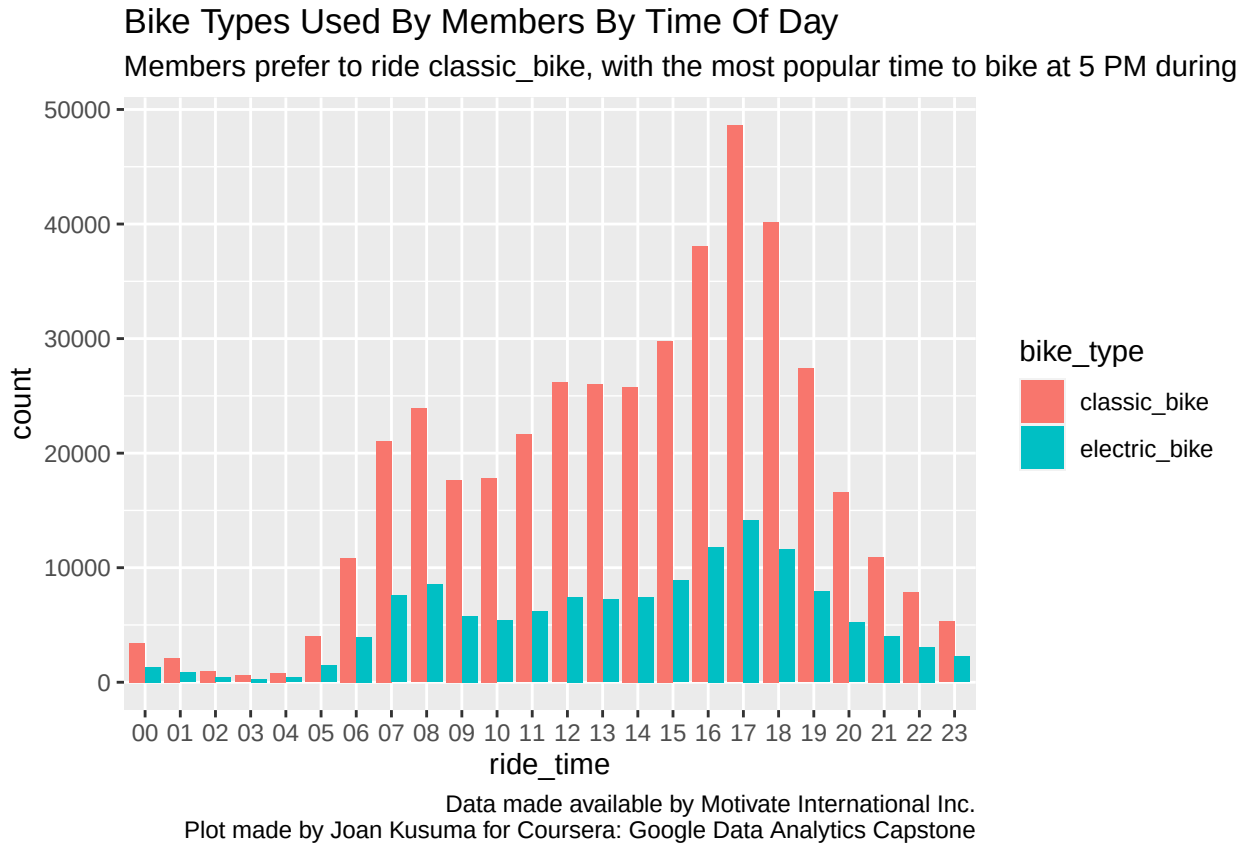
```
    counts[i] <- length(which(x==unique_val[i]))
  }
  position <- c(which(counts==max(counts)))
  if (length(unique_val)== length (x))
    mode_x <- 'Mode does not exist'
  else
    mode_x <- unique_val[position]
  return(mode_x)
}
```

- **Spring Analysis**

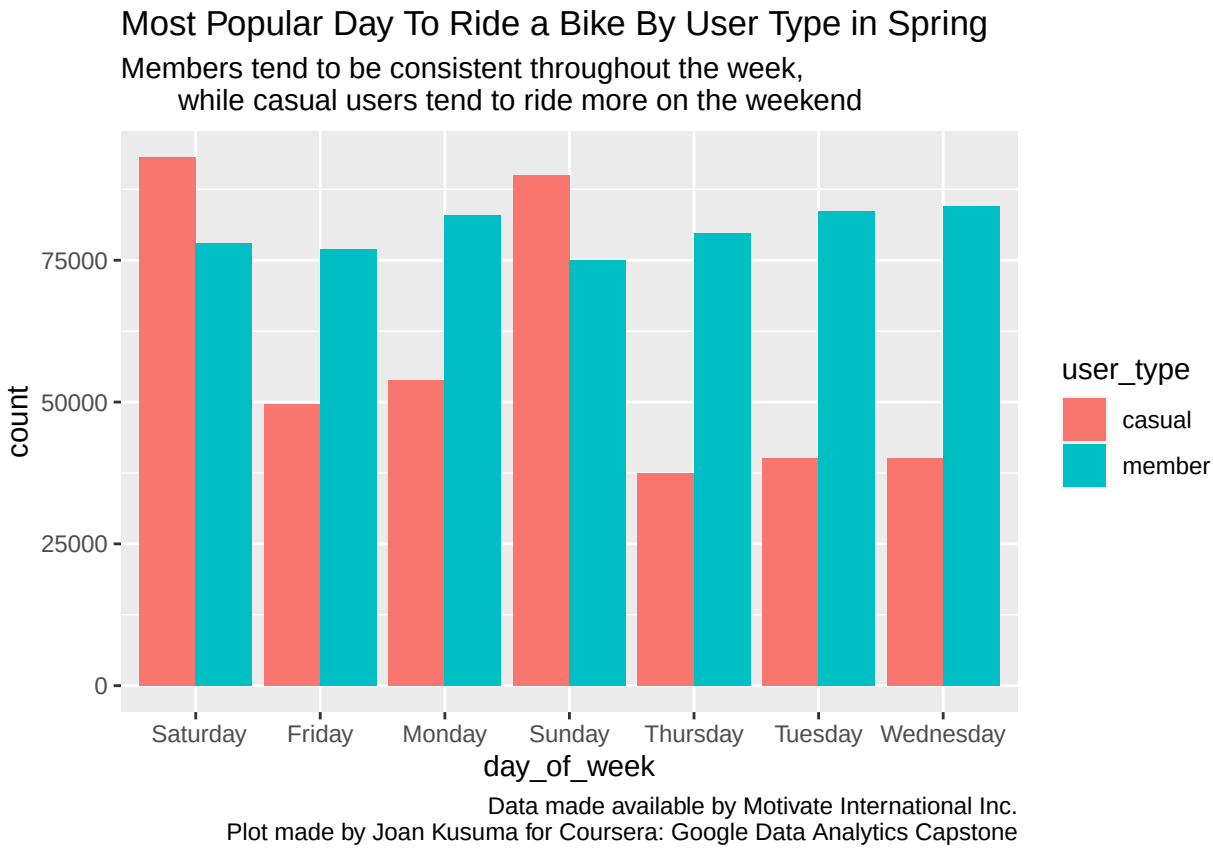
```
spring_analysis <- spring_df%>%
  group_by(user_type)%>%
  summarize(average_ride_length=seconds_to_period(mean(ride_length)),
            mode_day_of_week = mode(day_of_week),
            mode_start_station_name = mode(start_station_name),
            mode_end_station_name = mode(end_station_name),
            mode_ride_time = mode(ride_time),
            total = n())

View(spring_analysis)
write.csv(spring_analysis, file='spring_analysis.csv')

##spring_casual_users
spring_casual <- spring_df%>%
  filter(user_type=="casual")
View(spring_casual)

##spring_casual_start_station_analysis
spring_casual_start_station_analysis <- spring_casual%>%
  group_by(user_type)%>%
  count(start_station_name=spring_casual$start_station_name)

View(spring_casual_start_station_analysis)

##spring_casual_end_station_analysis
spring_casual_end_station_analysis <- spring_casual%>%
  group_by(user_type)%>%
  count(end_station_name=spring_casual$end_station_name)

View(spring_casual_end_station_analysis)

##plot:spring_casual_ridetime_and_biketype
ggplot(data=spring_casual, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike Types Used By Casual Users By Time Of Day",
       subtitle="Casual users prefer to ride classic_bike, with the most popular time to bike at 5 PM du
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



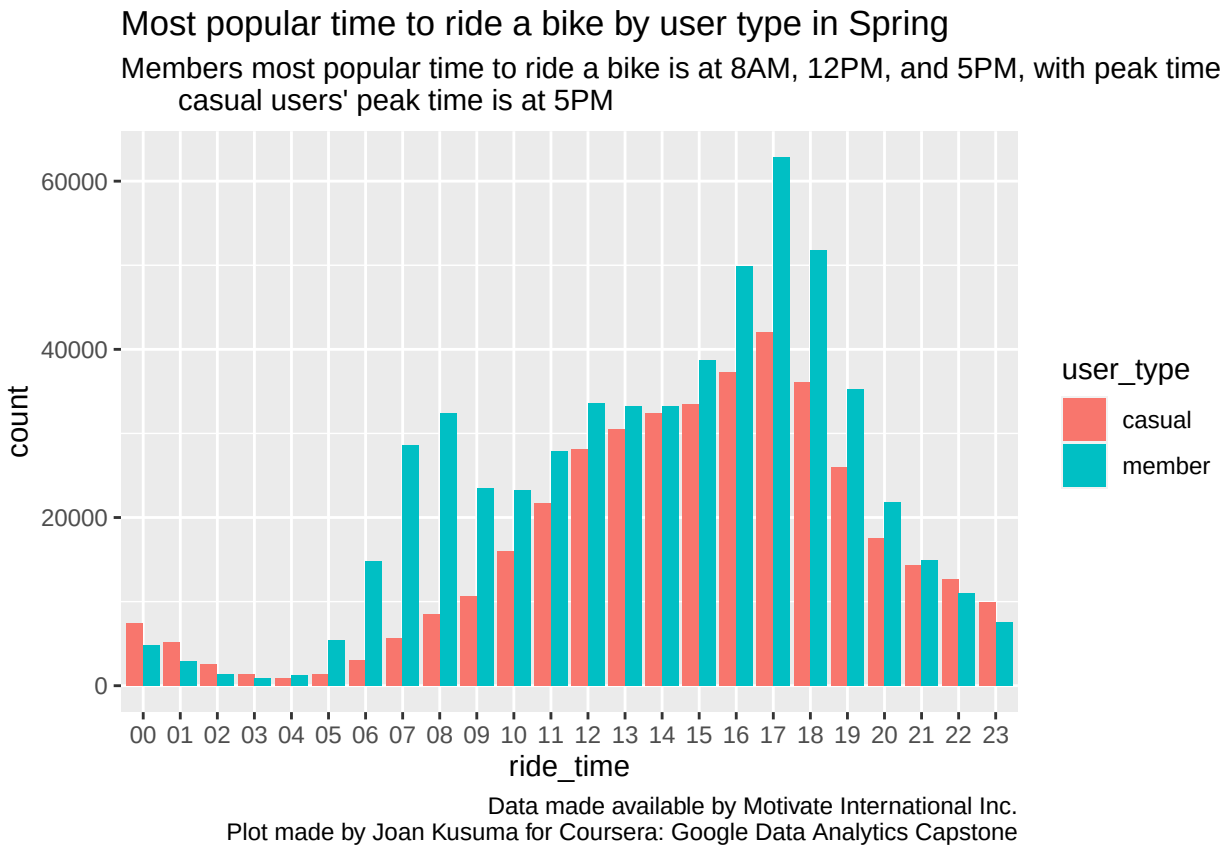

```
##spring_member_users
spring_member <- spring_df%>%
  filter(user_type=="member")
View(spring_member)

##spring_members_start_station_analysis
spring_member_start_station_analysis <- spring_member%>%
  group_by(user_type)%>%
  count(start_station_name)

View(spring_member_start_station_analysis)

##spring_members_end_station_analysis
spring_member_end_station_analysis <- spring_member%>%
  group_by(user_type)%>%
  count(end_station_name)

View(spring_member_end_station_analysis)

##plot:spring_member_ridetime_and_biketype
ggplot(data=spring_member, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike Types Used By Members By Time Of Day",
       subtitle="Members prefer to ride classic_bike, with the most popular time to bike at 5 PM during
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot:spring_dayofweek_by_usertype
ggplot(data=spring_df, aes(x=day_of_week, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most Popular Day To Ride a Bike By User Type in Spring",
       subtitle="Members tend to be consistent throughout the week,
       while casual users tend to ride more on the weekend",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



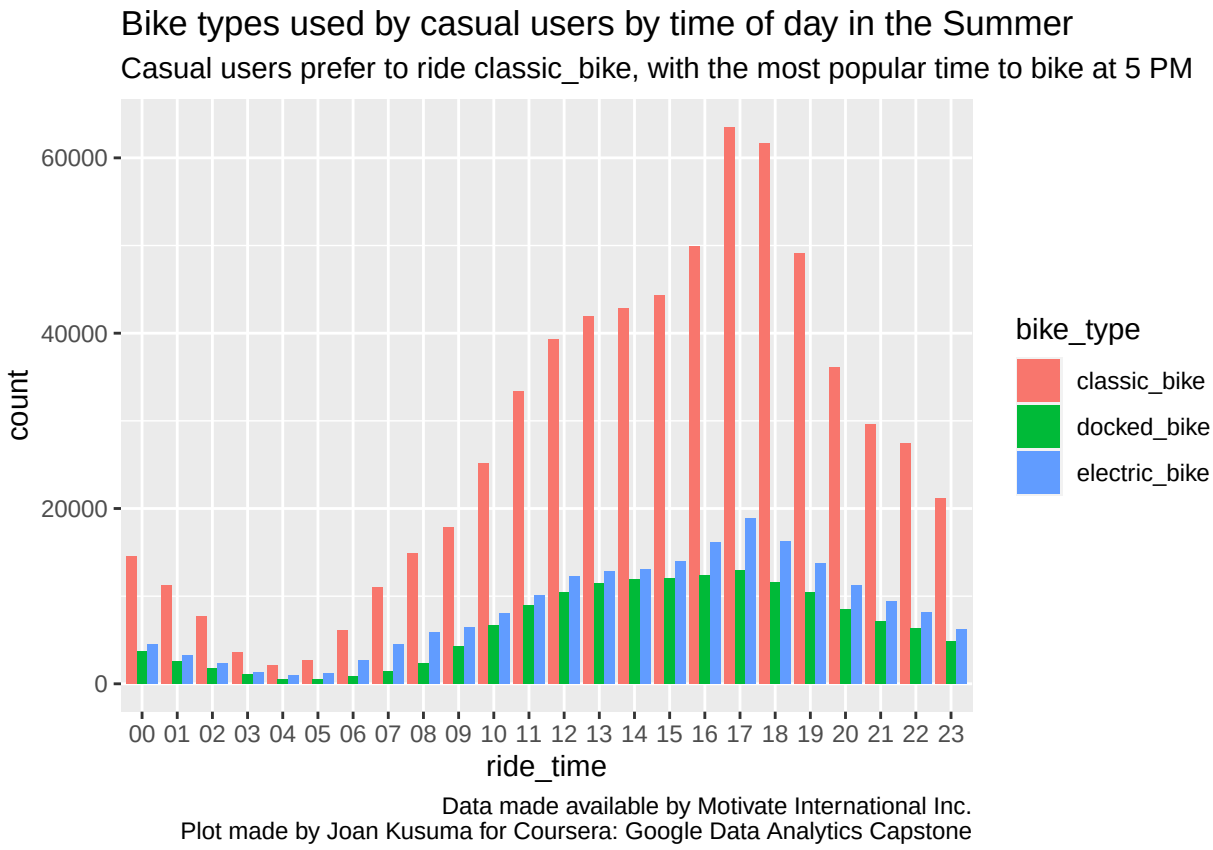

```
##plot: spring_peaktime_by_usertype
ggplot(data=spring_df, aes(x=ride_time, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most popular time to ride a bike by user type in Spring",
       subtitle="Members most popular time to ride a bike is at 8AM, 12PM, and 5PM, with peak time at 5
       casual users' peak time is at 5PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



- **Summer Analysis**

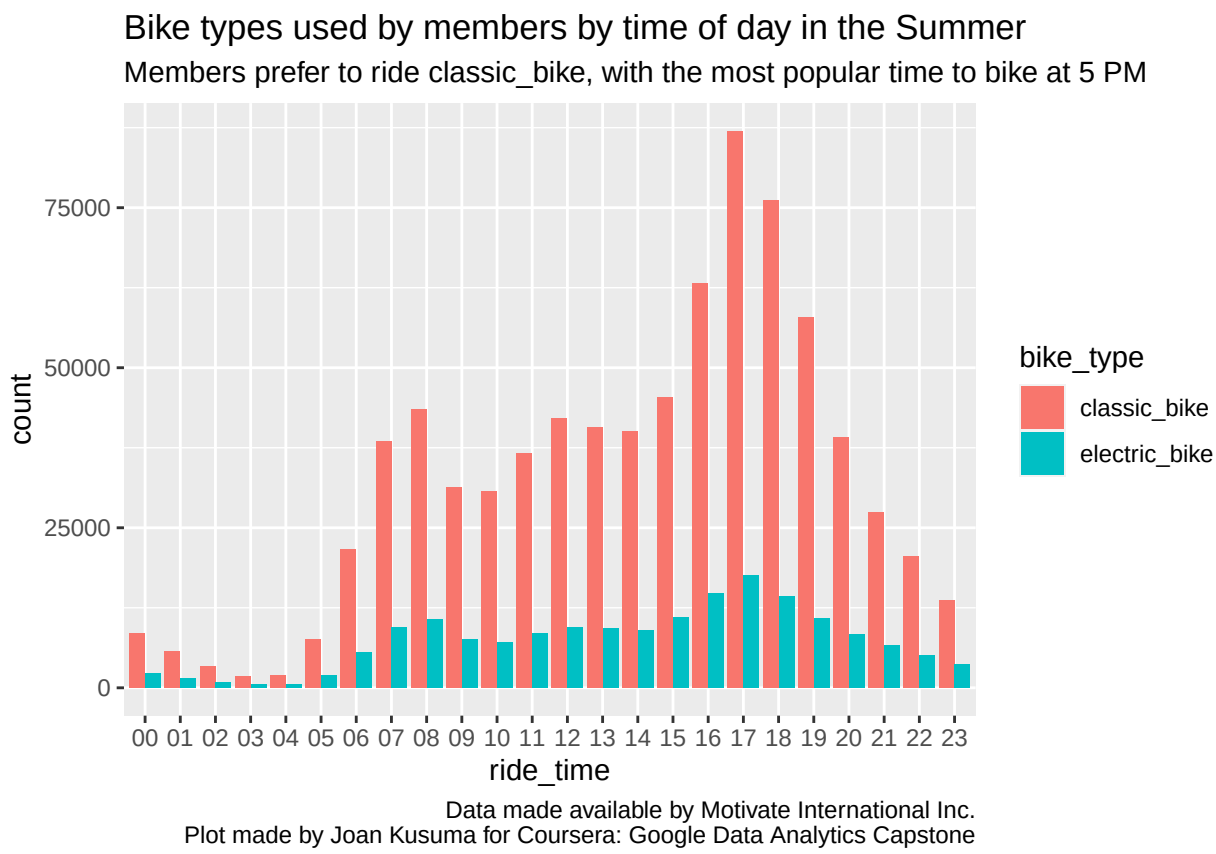
```
summer_analysis <- summer_df%>%
  group_by(user_type)%>%
  summarize(average_ride_length=seconds_to_period(mean(ride_length)),
            mode_day_of_week = mode(day_of_week),
            mode_ride_time = mode(ride_time),
            total = n())
View(summer_analysis)

##summer_casual_users
summer_casual <- summer_df%>%
  filter(user_type=="casual")
View(summer_casual)

##summer_casual_start_station_analysis
summer_casual_start_station_analysis <- summer_casual%>%
  group_by(user_type)%>%
  count(start_station_name)

View(summer_casual_start_station_analysis)

##summer_casual_end_station_analysis
summer_casual_end_station_analysis <- summer_casual%>%
  group_by(user_type)%>%
  count(end_station_name)
```

```
View(summer_casual_end_station_analysis)

##plot:summer_casual_ridetime_and_biketype
ggplot(data=summer_casual, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by casual users by time of day in the Summer",
       subtitle="Casual users prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```

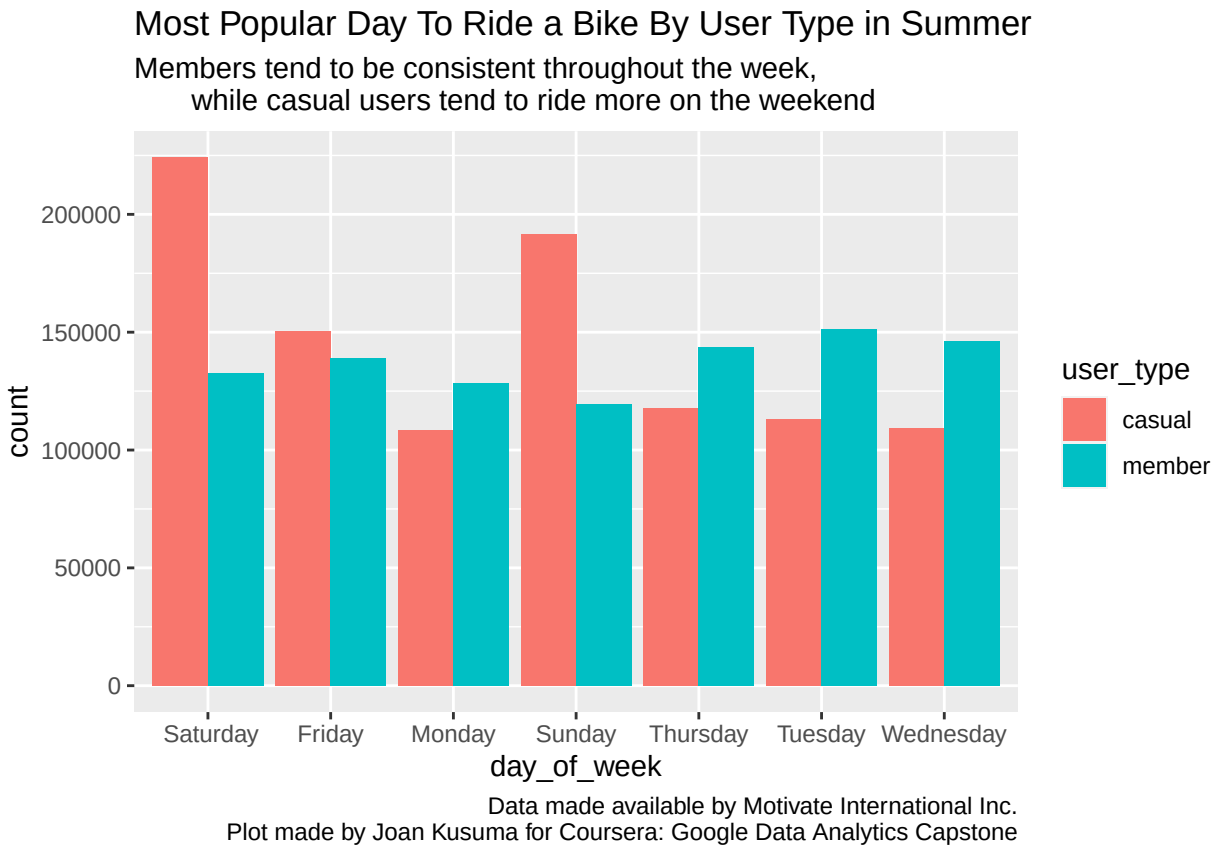



```
##summer_member_users
summer_member <- summer_df%>%
  filter(user_type=="member")
View(summer_member)

##summer_member_start_station_analysis
summer_member_start_station_analysis <- summer_member%>%
  group_by(user_type)%>%
  count(start_station_name)

View(summer_member_start_station_analysis)

##summer_member_end_station_analysis
summer_member_end_station_analysis <- summer_member%>%
  group_by(user_type)%>%
  count(end_station_name)
```

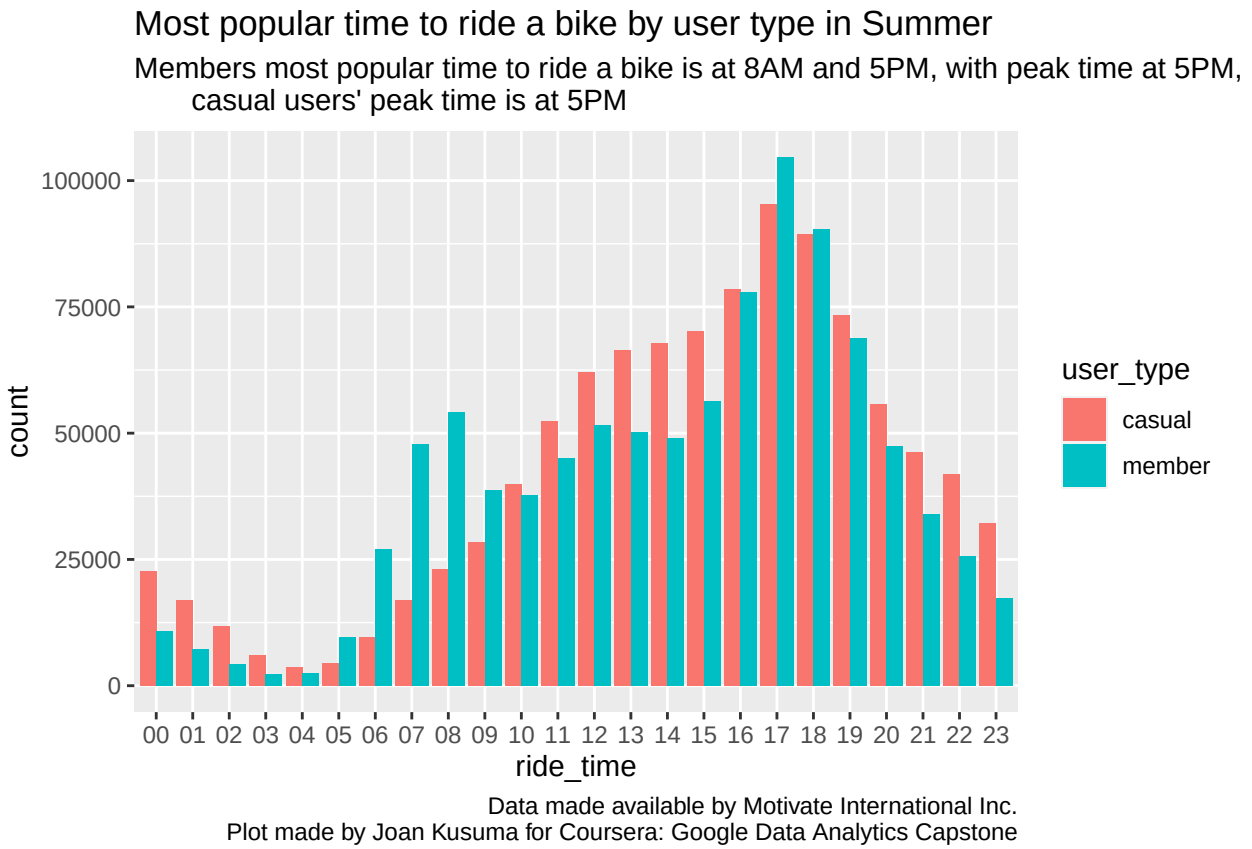
```
View(summer_member_end_station_analysis)

##plot:summer_member_ridetime_and_biketype
ggplot(data=summer_member, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by members by time of day in the Summer",
       subtitle="Members prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot:summer_dayofweek_by_usertype
ggplot(data=summer_df, aes(x=day_of_week, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most Popular Day To Ride a Bike By User Type in Summer",
       subtitle="Members tend to be consistent throughout the week,
       while casual users tend to ride more on the weekend",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



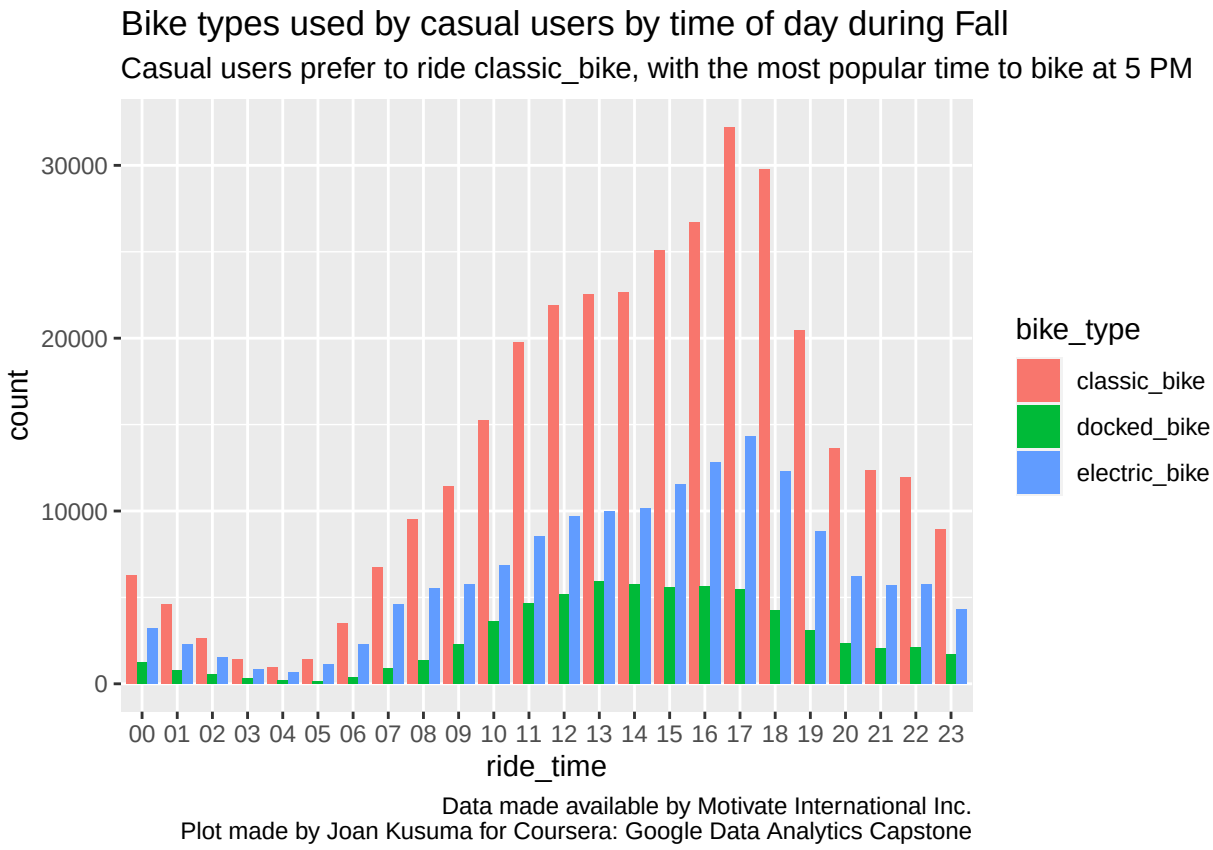

```
##plot: summer_peaktime_by_usertype
ggplot(data=summer_df, aes(x=ride_time, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most popular time to ride a bike by user type in Summer",
       subtitle="Members most popular time to ride a bike is at 8AM and 5PM, with peak time at 5PM,
       casual users' peak time is at 5PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



- **Fall Analysis**

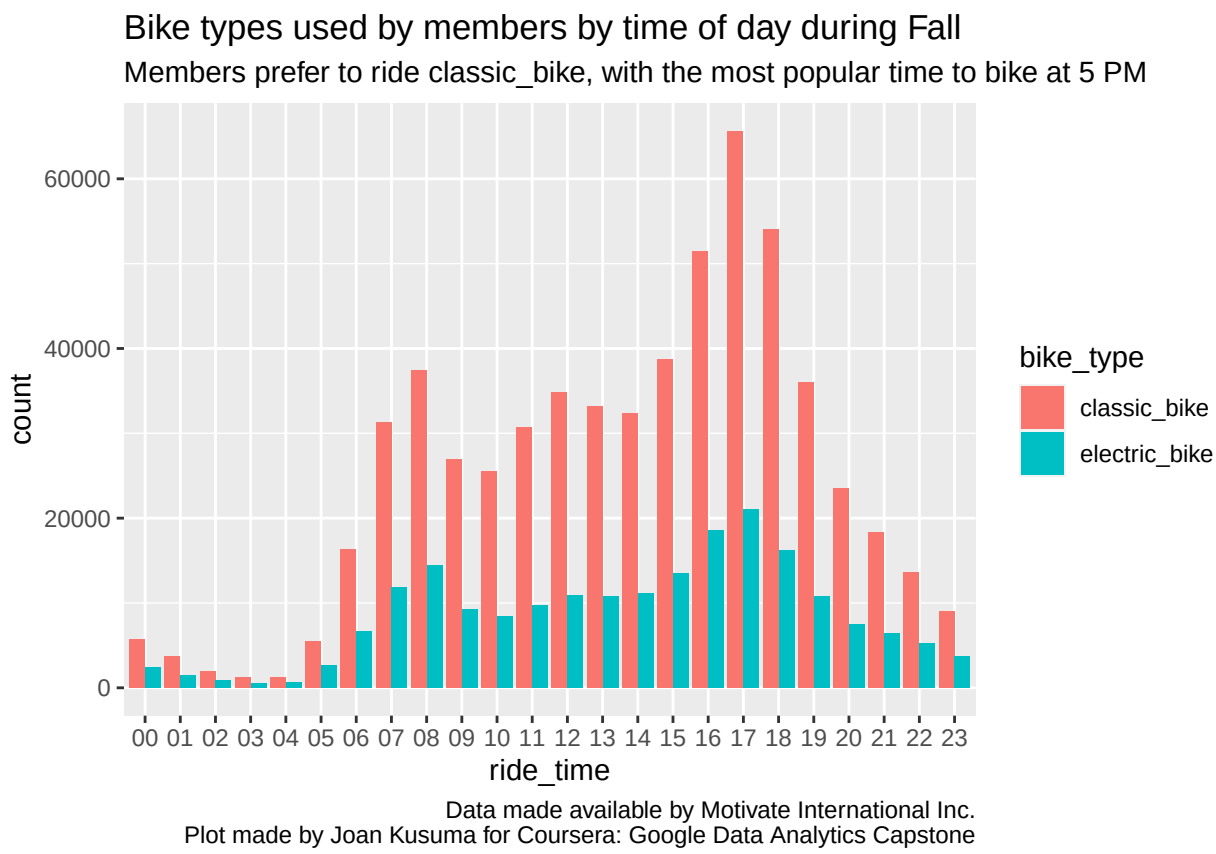
```
fall_analysis <- fall_df%>%
  group_by(user_type)%>%
  summarize(average_ride_length=seconds_to_period(mean(ride_length)),
            mode_day_of_week = mode(day_of_week),
            mode_ride_time = mode(ride_time),
            total = n())
View(fall_analysis)

##fall_casual_users
fall_casual <- fall_df%>%
  filter(user_type=="casual")
View(fall_casual)

##fall_casual_start_station_analysis
fall_casual_start_station_analysis <- fall_casual%>%
  group_by(user_type)%>%
  count(start_station_name)

View(fall_casual_start_station_analysis)

##fall_casual_end_station_analysis
fall_casual_end_station_analysis <- fall_casual%>%
  group_by(user_type)%>%
  count(end_station_name)
```

```
View(fall_casual_end_station_analysis)

##plot:fall_casual_ridetime_and_biketype
ggplot(data=fall_casual, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by casual users by time of day during Fall",
       subtitle="Casual users prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```

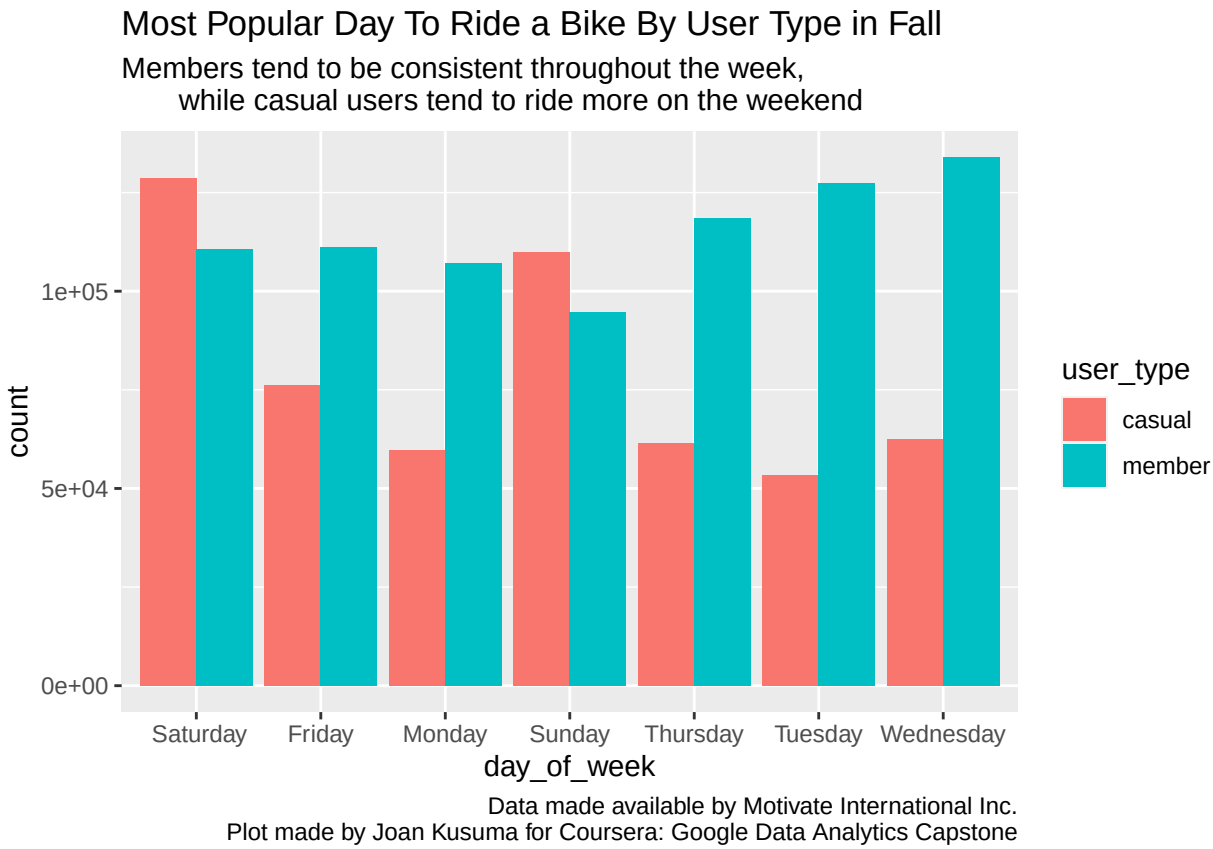



```
##fall_member_users
fall_member <- fall_df%>%
  filter(user_type=="member")
View(fall_member)

##fall_member_start_station_analysis
fall_member_start_station_analysis <- fall_member%>%
  group_by(user_type)%>%
  count(start_station_name)

View(fall_member_start_station_analysis)

##fall_member_end_station_analysis
fall_member_end_station_analysis <- fall_member%>%
  group_by(user_type)%>%
  count(end_station_name)
```

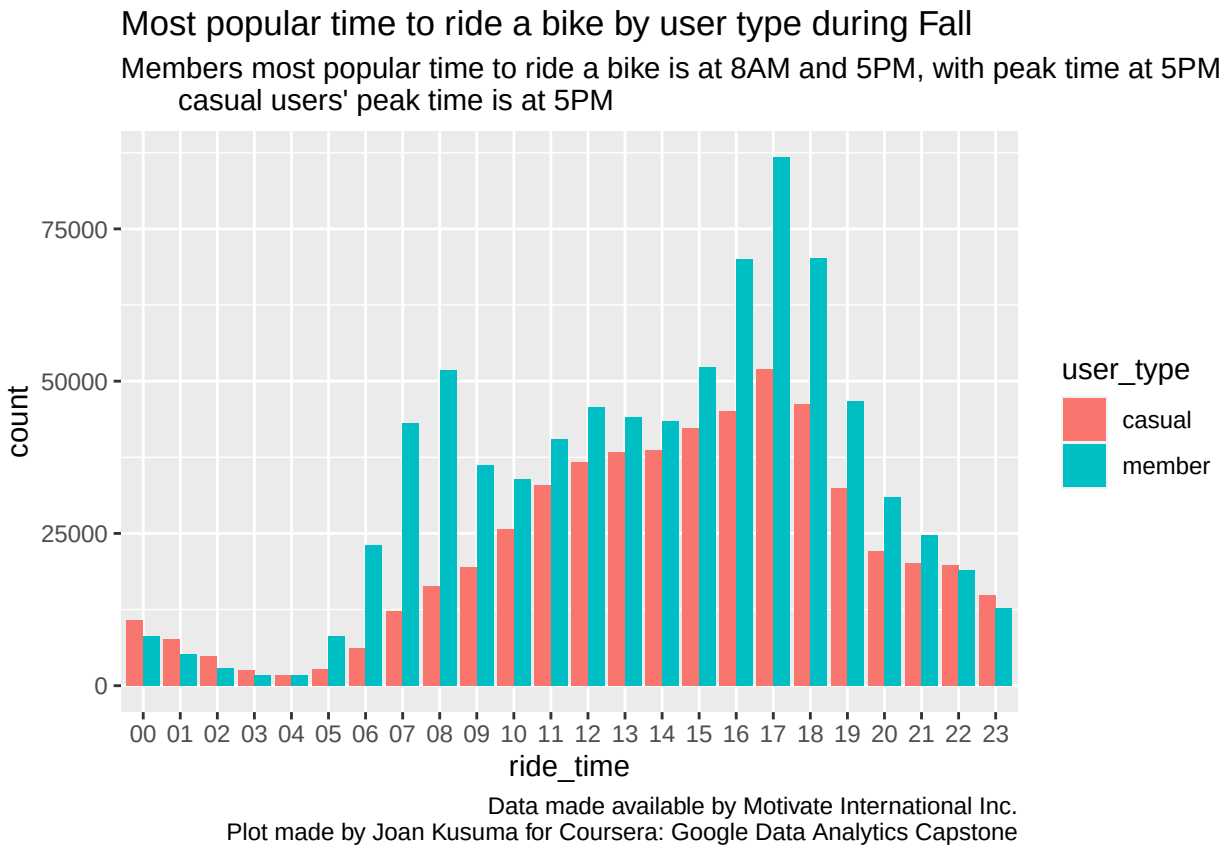
```
View(fall_member_end_station_analysis)

##plot:fall_member_ridetime_and_biketype
ggplot(data=fall_member, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by members by time of day during Fall",
       subtitle="Members prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot:fall_dayofweek_by_usertype
ggplot(data=fall_df, aes(x=day_of_week, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most Popular Day To Ride a Bike By User Type in Fall",
       subtitle="Members tend to be consistent throughout the week,
       while casual users tend to ride more on the weekend",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



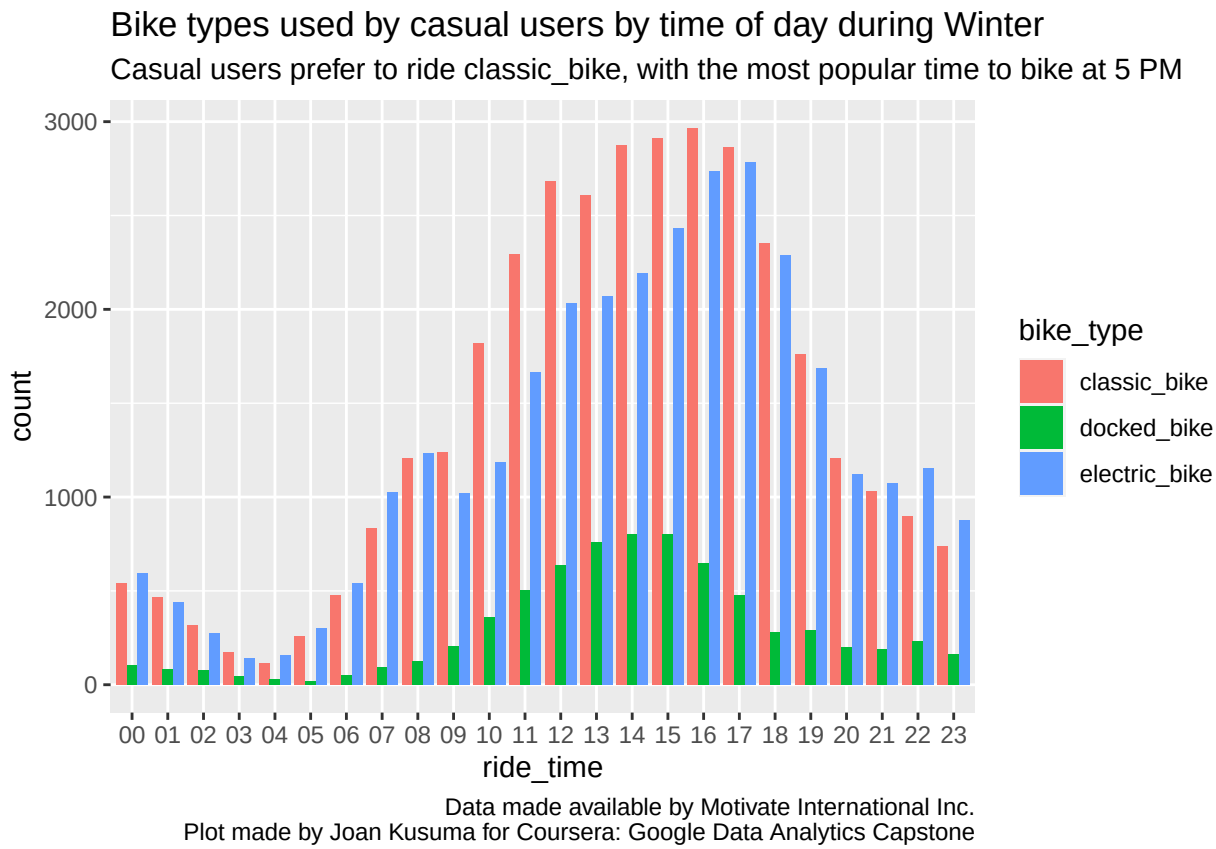

```
##plot: fall_peaktime_by_usertype
ggplot(data=fall_df, aes(x=ride_time, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most popular time to ride a bike by user type during Fall",
       subtitle="Members most popular time to ride a bike is at 8AM and 5PM, with peak time at 5PM,
       casual users' peak time is at 5PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



- **Winter Analysis**

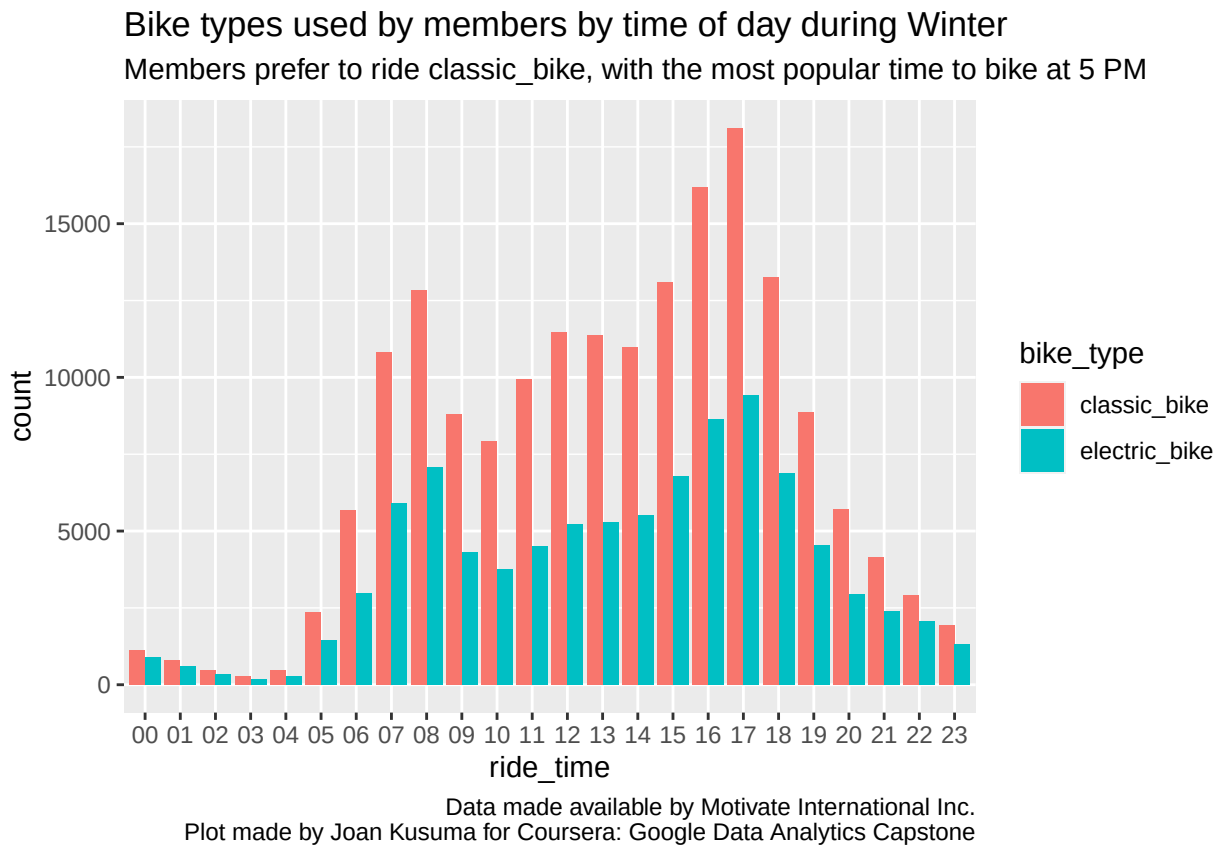
```
winter_analysis <- winter_df%>%
  group_by(user_type)%>%
  summarize(average_ride_length=seconds_to_period(mean(ride_length)),
            mode_day_of_week = mode(day_of_week),
            mode_ride_time = mode(ride_time),
            total = n())
View(winter_analysis)

##winter_casual_users
winter_casual <- winter_df%>%
  filter(user_type=="casual")
View(winter_casual)

##winter_casual_user_bike_type
winter_casual_bike_type <- winter_casual%>%
  group_by(bike_type)%>%
  summarize(total=n())

View(winter_casual_bike_type)

write.csv(winter_casual_bike_type, file='winter_casual_bike_type.csv')

##winter_casual_start_station_analysis
winter_casual_start_station_analysis <- winter_casual%>%
  group_by(user_type)%>%
```

```
  count(start_station_name=winter_casual$start_station_name)

View(winter_casual_start_station_analysis)

##winter_casual_end_station_analysis
winter_casual_end_station_analysis <- winter_casual%>%
  group_by(user_type)%>%
  count(end_station_name=winter_casual$end_station_name)

View(winter_casual_end_station_analysis)

##plot: winter_casual_ridetime_and_biketype
ggplot(data=winter_casual, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by casual users by time of day during Winter",
       subtitle="Casual users prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```

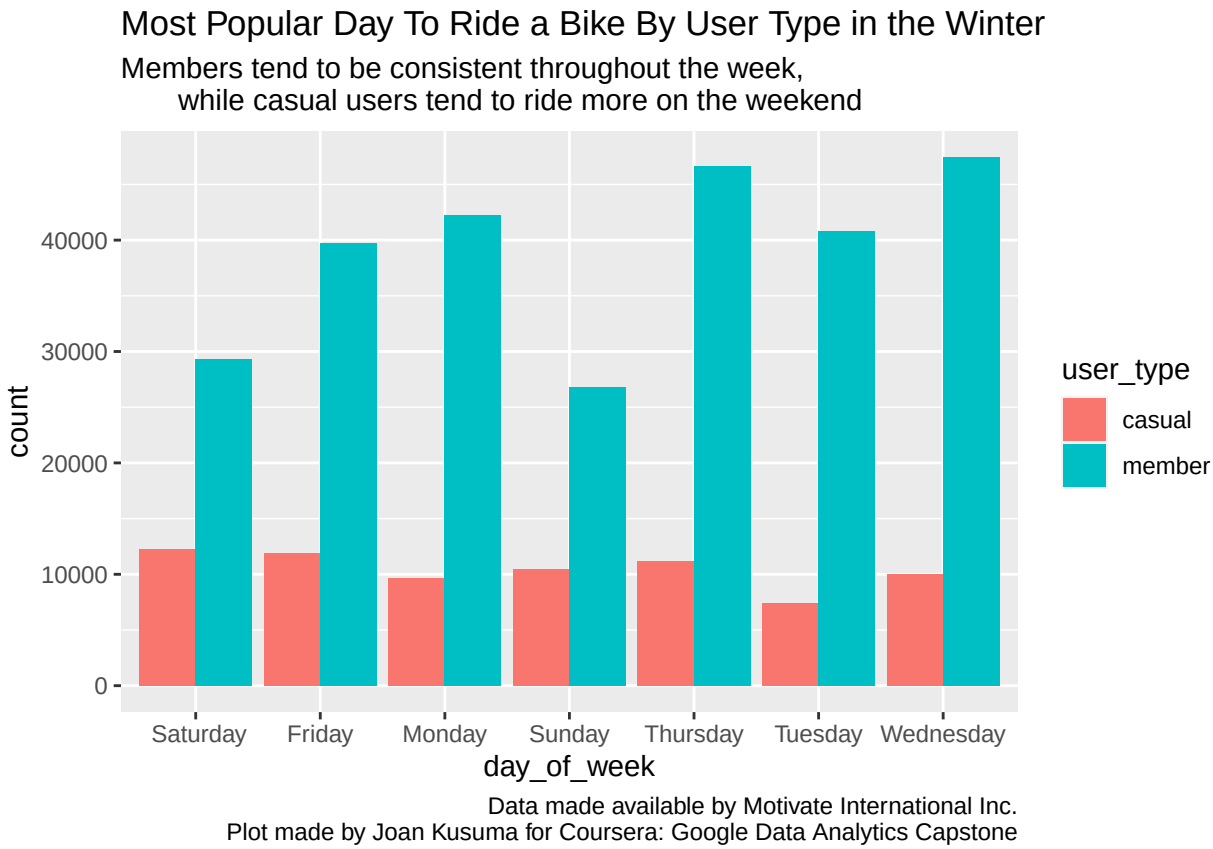



```
##winter_member_users
winter_member <- winter_df%>%
  filter(user_type=="member")
View(winter_member)

##winter_member_start_station_analysis
winter_member_start_station_analysis <- winter_member%>%
  group_by(user_type)%>%
```

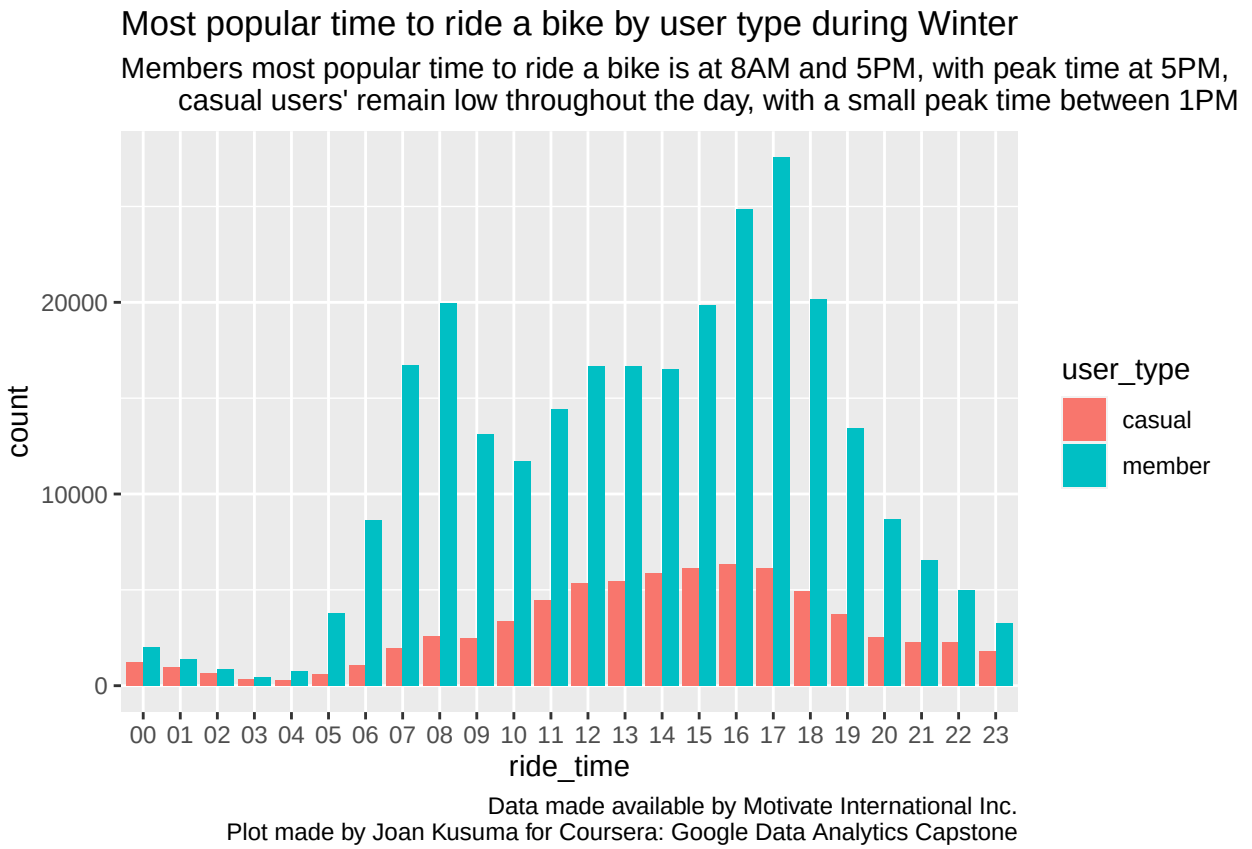
```
  count(start_station_name)

View(winter_member_start_station_analysis)

##winter_member_end_station_analysis
winter_member_end_station_analysis <- winter_member%>%
  group_by(user_type)%>%
  count(end_station_name)

View(winter_member_end_station_analysis)

##plot:winter_member_ridetime_and_biketype
ggplot(data=winter_member, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by members by time of day during Winter",
       subtitle="Members prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot:winter_dayofweek_by_usertype
ggplot(data=winter_df, aes(x=day_of_week, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most Popular Day To Ride a Bike By User Type in the Winter",
       subtitle="Members tend to be consistent throughout the week,
       while casual users tend to ride more on the weekend",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



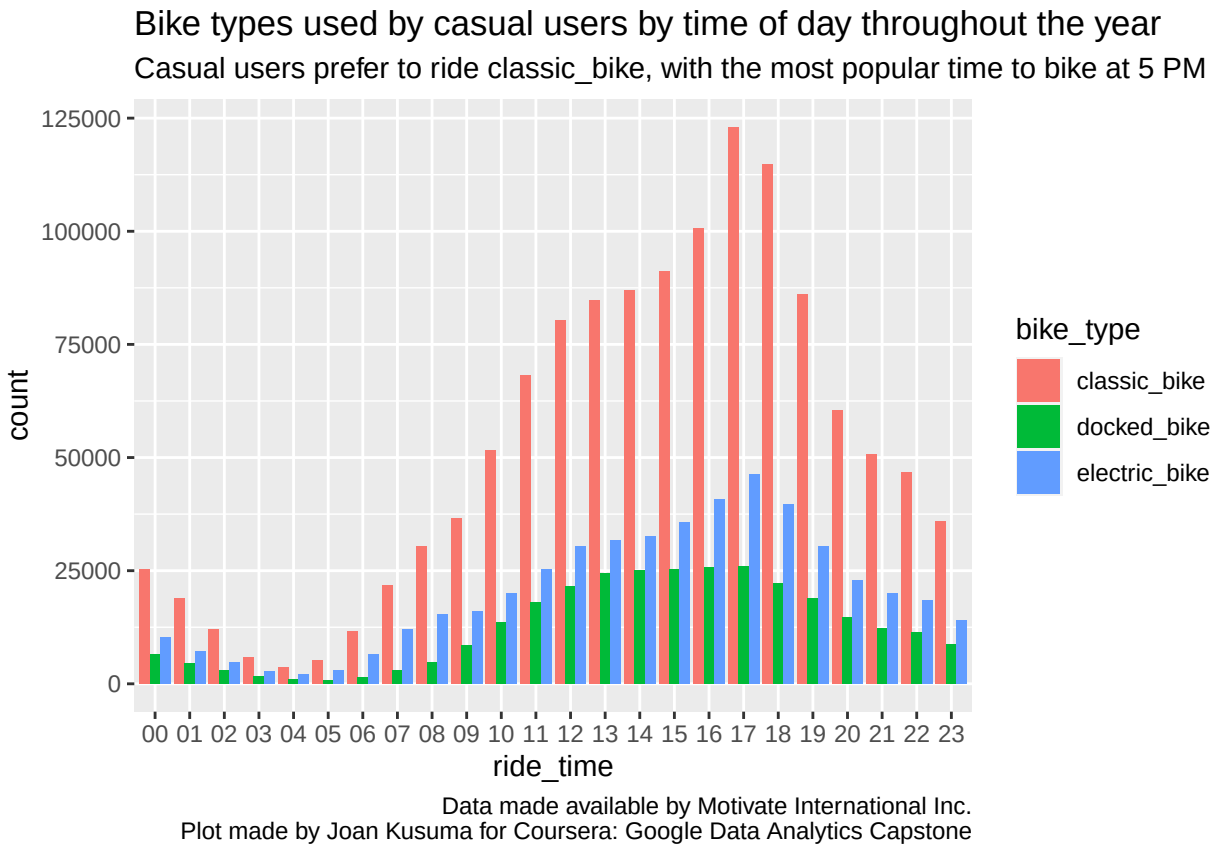

```
##plot: winter_peaktime_by_usertype
ggplot(data=winter_df, aes(x=ride_time, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most popular time to ride a bike by user type during Winter",
       subtitle="Members most popular time to ride a bike is at 8AM and 5PM, with peak time at 5PM,
       casual users' remain low throughout the day, with a small peak time between 1PM-5PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



- **Year-round analysis**

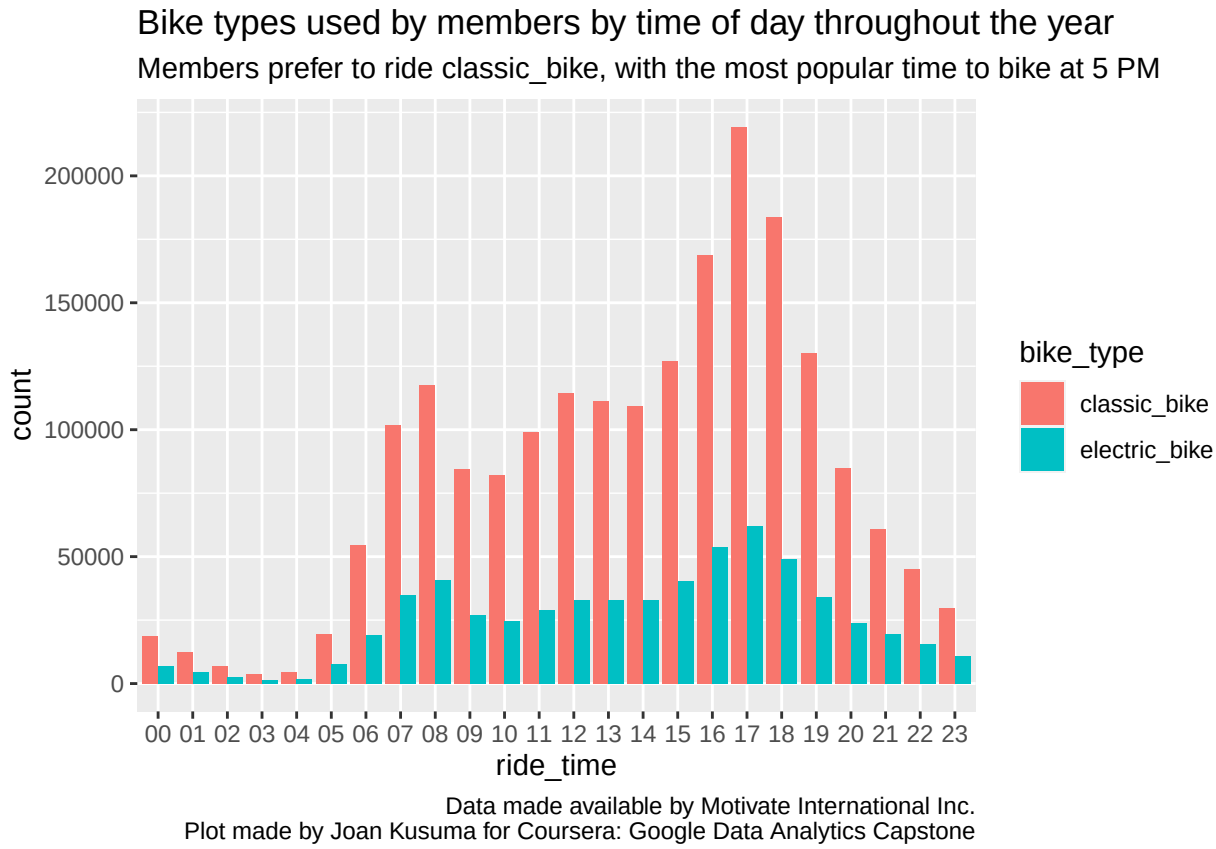
```
year_round_analysis <- year_round_df%>%
  group_by(user_type)%>%
  summarize(average_ride_length=seconds_to_period(mean(ride_length)),
            mode_day_of_week = mode(day_of_week),
            mode_ride_time = mode(ride_time),
            total = n())
View(year_round_analysis)

##year_roundcasual_users
year_round_casual <- year_round_df%>%
  filter(user_type=="casual")
View(year_round_casual)

##year_round_casual_start_and_end_station_analysis
year_round_casual_start_station_analysis <- year_round_casual%>%
  group_by(user_type)%>%
  count(start_station_name=year_round_casual$start_station_name)

View(year_round_casual_start_station_analysis)

##spring_casual_end_station_analysis
year_round_casual_end_station_analysis <- year_round_casual%>%
  group_by(user_type)%>%
  count(end_station_name=year_round_casual$end_station_name)
```

```
View(year_round_casual_end_station_analysis)

##plot: year_round_casual_ridetime_and_biketype
ggplot(data=year_round_casual, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by casual users by time of day throughout the year",
       subtitle="Casual users prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```

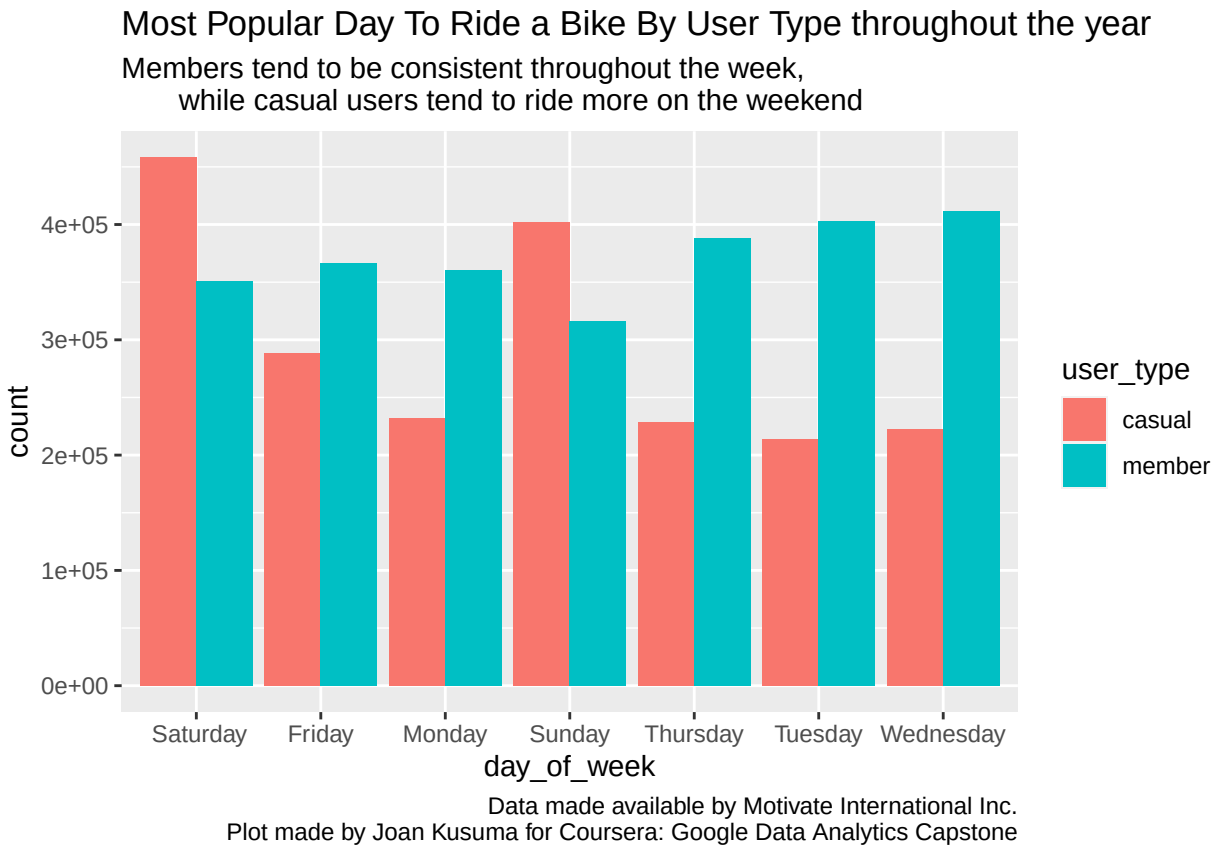



```
##year_round_member_users
year_round_member <- year_round_df%>%
  filter(user_type=="member")
View(year_round_member)

##year_round_member_start_station_analysis
year_round_member_start_station_analysis <- year_round_member%>%
  group_by(user_type)%>%
  count(start_station_name)

View(year_round_member_start_station_analysis)

write.csv(year_round_member_start_station_analysis, file='year_round_member_start_station.csv')

##year_round_member_end_station_analysis
year_round_member_end_station_analysis <- year_round_member%>%
  group_by(user_type)%>%
```

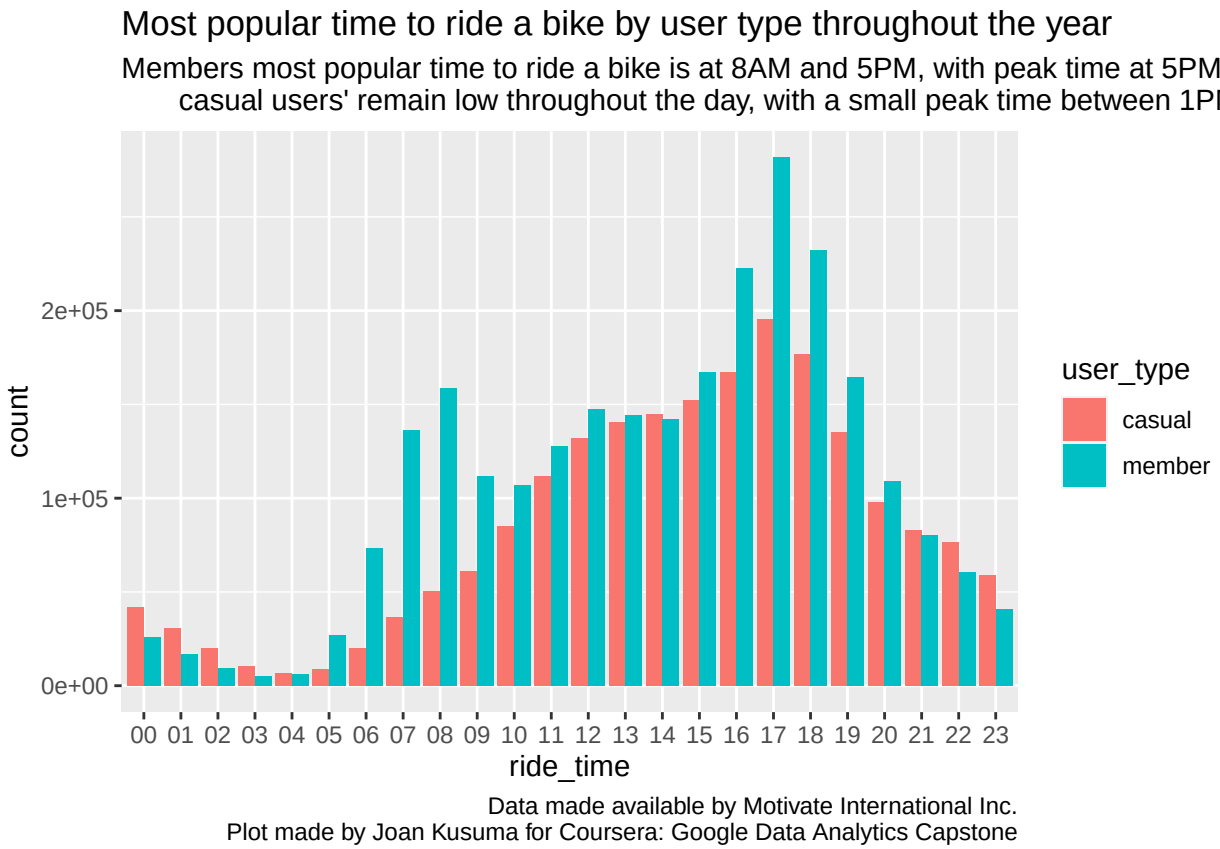
```
  count(end_station_name)

View(year_round_member_end_station_analysis)

##plot: year_round_member_ridetime_and_biketype
ggplot(data=year_round_member, aes(x=ride_time, fill=bike_type))+
  geom_bar(position="dodge")+
  labs(title="Bike types used by members by time of day throughout the year",
       subtitle="Members prefer to ride classic_bike, with the most popular time to bike at 5 PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot: year_round_dayofweek_by_usertype
ggplot(data=year_round_df, aes(x=day_of_week, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most Popular Day To Ride a Bike By User Type throughout the year",
       subtitle="Members tend to be consistent throughout the week,
       while casual users tend to ride more on the weekend",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```




```
##plot: year_round_peaktime_by_usertype
ggplot(data=year_round_df, aes(x=ride_time, fill=user_type))+
  geom_bar(position="dodge")+
  labs(title="Most popular time to ride a bike by user type throughout the year",
       subtitle="Members most popular time to ride a bike is at 8AM and 5PM, with peak time at 5PM,
       casual users' remain low throughout the day, with a small peak time between 1PM-5PM",
       caption="Data made available by Motivate International Inc.
       Plot made by Joan Kusuma for Coursera: Google Data Analytics Capstone")
```



Complete analysis and presentation is available through the following link