capstone_case-study_1

Abhirup

2022-08-27

Google Data Analytics (Cyclistic) Capstone Project

#Introduction

This is my version of the Google Data Analytics Capstone - Case Study 1. The full document to the case study can be found in the Google Data Analytics Capstone: Complete a Case Study course.

For this project this steps will be followed to ensure its completion:

It will follow the steps of the data analysis process: Ask, prepare, process, analyze, share, and act. Each step will follow its own roadmap with: **Code, if needed on the step.** Guiding questions, with answers. **Key tasks, as a checklist.** Deliverable, as a checklist.

#Process

This step will prepare the data for analysis. All the csv files will be merged into one file to improve workflow

#Code

##Dependences The main dependencie for the project will be tidyverse.

```
# Install packages
# if (!require(package)) install.packages('package')
```

#Load Libraries

#Data

The data is on an AWS server where it is easily downloadable and named correctly. I downloaded the previous 12 months data and stored it locally for the next steps in the analysis processes. It is organized by year and Fiscal Quarters. The data is reliable and original since it comes from the company.It is comprehensive, current, and cited. The data source is the company so everything about the users personal information is hidden or kept private to the company only.

Note: that data-privacy issues prohibit you from using riders’ personally identifiable information. This means that you won’t be able to connect pass purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

###Data Set URL: https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html)

#STEP 1: COLLECT DATA

```
#load original .csv files, a years worth of data from August 2020 to July 2021
Q1_2020_df <- read_csv("cyclistic_data/Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426881 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, en...
##
## 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.
```

```
apr_2020_df <- read_csv("cyclistic_data/202004-divvy-tripdata.csv")
```

```
## Rows: 84776 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, en...
##
## 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.
```

```
may_2020_df <- read_csv("cyclistic_data/202005-divvy-tripdata.csv")
```

```
## Rows: 200274 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jun_2020_df <- read_csv("cyclistic_data/202006-divvy-tripdata.csv")
```

```
## Rows: 343005 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jul_2020_df <- read_csv("cyclistic_data/202007-divvy-tripdata.csv")
```

```
## Rows: 551480 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
aug_2020_df <- read_csv("cyclistic_data/202008-divvy-tripdata.csv")
```

```
## Rows: 622361 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sep_2020_df <- read_csv("cyclistic_data/202009-divvy-tripdata.csv")
```

```
## Rows: 532958 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
oct_2020_df <- read_csv("cyclistic_data/202010-divvy-tripdata.csv")
```

```
## Rows: 388653 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nov_2020_df <- read_csv("cyclistic_data/202011-divvy-tripdata.csv")
```

```
## Rows: 259716 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dec_2020_df <- read_csv("cyclistic_data/202012-divvy-tripdata.csv")
```

```
## Rows: 131573 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jan_2021_df <- read_csv("cyclistic_data/202101-divvy-tripdata.csv")
```

```
## Rows: 96834 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
feb_2021_df <- read_csv("cyclistic_data/202102-divvy-tripdata.csv")
```

```
## Rows: 49622 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mar_2021_df <- read_csv("cyclistic_data/202103-divvy-tripdata.csv")
```

```
## Rows: 228496 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
apr_2021_df <- read_csv("cyclistic_data/202104-divvy-tripdata.csv")
```

```
## Rows: 337230 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
may_2021_df <- read_csv("cyclistic_data/202105-divvy-tripdata.csv")
```

```
## Rows: 531633 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jun_2021_df <- read_csv("cyclistic_data/202106-divvy-tripdata.csv")
```

```
## Rows: 729595 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jul_2021_df <- read_csv("cyclistic_data/202107-divvy-tripdata.csv")
```

```
## Rows: 822410 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
aug_2021_df <- read_csv("cyclistic_data/202108-divvy-tripdata.csv")
```

```
## Rows: 804352 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sep_2021_df <- read_csv("cyclistic_data/202109-divvy-tripdata.csv")
```

```
## Rows: 756147 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
oct_2021_df <- read_csv("cyclistic_data/202110-divvy-tripdata.csv")
```

```
## Rows: 631226 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nov_2021_df <- read_csv("cyclistic_data/202111-divvy-tripdata.csv")
```

```
## Rows: 359978 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dec_2021_df <- read_csv("cyclistic_data/202112-divvy-tripdata.csv")
```

```
## Rows: 247540 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jan_2022_df <- read_csv("cyclistic_data/202201-divvy-tripdata.csv")
```

```
## Rows: 103770 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
feb_2022_df <- read_csv("cyclistic_data/202202-divvy-tripdata.csv")
```

```
## Rows: 115609 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mar_2022_df <- read_csv("cyclistic_data/202203-divvy-tripdata.csv")
```

```
## Rows: 284042 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
apr_2022_df <- read_csv("cyclistic_data/202204-divvy-tripdata.csv")
```

```
## Rows: 371249 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
may_2022_df <- read_csv("cyclistic_data/202205-divvy-tripdata.csv")
```

```
## Rows: 634858 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jun_2022_df <- read_csv("cyclistic_data/202206-divvy-tripdata.csv")
```

```
## Rows: 769204 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jul_2022_df <- read_csv("cyclistic_data/202207-divvy-tripdata.csv")
```

```
## Rows: 823488 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#fixing datatypes
# This step was necessary because the below imported CSV data was not in correct date-time fo
rmat.
# Wrong format could through error in later part.

Q1_2020_df <- Q1_2020_df %>% mutate(started_at = dmy_hm(started_at))
Q1_2020_df <- Q1_2020_df %>% mutate(ended_at = dmy_hm(ended_at))

apr_2020_df <- apr_2020_df %>% mutate(started_at = dmy_hm(started_at))
apr_2020_df <- apr_2020_df %>% mutate(ended_at = dmy_hm(ended_at))
```

#WRANGLE DATA AND COMBINE INTO A SINGLE FILE

```
#merge all of the data frames into one year view
all_trips <- rbind(Q1_2020_df,
                   apr_2020_df,
                   may_2020_df,
                   jun_2020_df,
                   jul_2020_df,
                   aug_2020_df,
                   sep_2020_df,
                   oct_2020_df,
                   nov_2020_df,
                   dec_2020_df,

                   jan_2021_df,
                   feb_2021_df,
                   mar_2021_df,
                   apr_2021_df,
                   may_2021_df,
                   jun_2021_df,
                   jul_2021_df,
                   aug_2021_df,
                   sep_2021_df,
                   oct_2021_df,
                   nov_2021_df,
                   dec_2021_df,

                   jan_2022_df,
                   feb_2022_df,
                   mar_2022_df,
                   apr_2022_df,
                   may_2022_df,
                   jun_2022_df,
                   jul_2022_df)
```

#Remove temorary data frames

```
#remove individual month data frames to clear up space in the environment
remove(Q1_2020_df,
       apr_2020_df,
       may_2020_df,
       jun_2020_df,
       jul_2020_df,
       aug_2020_df,
       sep_2020_df,
       oct_2020_df,
       nov_2020_df,
       dec_2020_df,

       jan_2021_df,
       feb_2021_df,
       mar_2021_df,
       apr_2021_df,
       may_2021_df,
       jun_2021_df,
       jul_2021_df,
       aug_2021_df,
       sep_2021_df,
       oct_2021_df,
       nov_2021_df,
       dec_2021_df,

       jan_2022_df,
       feb_2022_df,
       mar_2022_df,
       apr_2022_df,
       may_2022_df,
       jun_2022_df,
       jul_2022_df)
```

#Summerize Data

```
#Summerize Data
colnames(all_trips)  #List of column names
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
nrow(all_trips)  #How many rows are in data frame?
```

```
## [1] 12238960
```

```
dim(all_trips)  #Dimensions of the data frame?
```

```
## [1] 12238960       13
```

```
head(all_trips)  #See the first 6 rows of data frame.  Also tail(qs_raw)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 1068AB1B8F12F… docked… 2020-01-01 00:04:00 2020-01-01 00:17:00 Sheffi… 115
## 2 DCF74A0EB3284… docked… 2020-01-01 00:10:00 2020-01-01 00:10:00 Daley … 81
## 3 4DE50A4FC7687… docked… 2020-01-01 00:11:00 2020-01-01 00:15:00 Daley … 81
## 4 1C78B5F337CBF… docked… 2020-01-01 00:11:00 2020-01-01 00:13:00 Sherid… 240
## 5 D231CE7990A3A… docked… 2020-01-01 00:12:00 2020-01-01 00:14:00 Delano… 626
## 6 EB21BB139ABF6… docked… 2020-01-01 00:21:00 2020-01-01 00:29:00 Clark … 326
## # … with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

```
str(all_trips)  #See list of columns and data types (numeric, character, etc)
```

```
## tibble [12,238,960 × 13] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:12238960] "1068AB1B8F12FE23" "DCF74A0EB3284B3E" "4DE50A4FC76
87A0D" "1C78B5F337CBFC93" ...
##  $ rideable_type     : chr [1:12238960] "docked_bike" "docked_bike" "docked_bike" "docked_
bike" ...
##  $ started_at        : POSIXct[1:12238960], format: "2020-01-01 00:04:00" "2020-01-01 00:1
0:00" ...
##  $ ended_at          : POSIXct[1:12238960], format: "2020-01-01 00:17:00" "2020-01-01 00:1
0:00" ...
##  $ start_station_name: chr [1:12238960] "Sheffield Ave & Wellington Ave" "Daley Center Pla
za" "Daley Center Plaza" "Sheridan Rd & Irving Park Rd" ...
##  $ start_station_id  : chr [1:12238960] "115" "81" "81" "240" ...
##  $ end_station_name  : chr [1:12238960] "Ashland Ave & Belle Plaine Ave" "Daley Center Pla
za" "Dearborn St & Van Buren St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr [1:12238960] "246" "81" "624" "256" ...
##  $ start_lat         : num [1:12238960] 41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num [1:12238960] -87.7 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:12238960] 42 41.9 41.9 42 41.9 ...
##  $ end_lng           : num [1:12238960] -87.7 -87.6 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:12238960] "casual" "member" "member" "member" ...
```

```
summary(all_trips)  #Statistical summary of data. Mainly for numerics
```

```
##    ride_id          rideable_type        started_at                   
##  Length:12238960    Length:12238960    Min.   :2020-01-01 00:04:00.00  
##  Class :character   Class :character   1st Qu.:2020-10-22 14:58:22.25  
##  Mode  :character   Mode  :character   Median :2021-07-24 12:27:47.00  
##                                        Mean   :2021-06-29 09:33:47.16  
##                                        3rd Qu.:2022-01-13 21:21:54.75  
##                                        Max.   :2022-07-31 23:59:58.00  
##                                                                         
##     ended_at                     start_station_name start_station_id  
##  Min.   :2020-01-01 00:10:00.00   Length:12238960    Length:12238960   
##  1st Qu.:2020-10-22 15:17:23.75   Class :character   Class :character  
##  Median :2021-07-24 12:55:01.50   Mode  :character   Mode  :character  
##  Mean   :2021-06-29 09:56:11.25                                        
##  3rd Qu.:2022-01-13 21:37:43.00                                        
##  Max.   :2022-08-04 13:53:01.00                                        
##                                                                         
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:12238960    Length:12238960    Min.   :41.64   Min.   :-87.87  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :45.64   Max.   :-73.80  
##                                                                         
##     end_lat         end_lng       member_casual     
##  Min.   :41.39   Min.   :-88.97   Length:12238960   
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.65                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.37   Max.   :-87.44                     
##  NA's   :12496   NA's   :12496
```

```
glimpse(all_trips) #summary
```

```
## Rows: 12,238,960
## Columns: 13
## $ ride_id            <chr> "1068AB1B8F12FE23", "DCF74A0EB3284B3E", "4DE50A4FC7…
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke…
## $ started_at         <dttm> 2020-01-01 00:04:00, 2020-01-01 00:10:00, 2020-01-…
## $ ended_at           <dttm> 2020-01-01 00:17:00, 2020-01-01 00:10:00, 2020-01-…
## $ start_station_name <chr> "Sheffield Ave & Wellington Ave", "Daley Center Pla…
## $ start_station_id   <chr> "115", "81", "81", "240", "626", "326", "326", "347…
## $ end_station_name   <chr> "Ashland Ave & Belle Plaine Ave", "Daley Center Pla…
## $ end_station_id     <chr> "246", "81", "624", "256", "59", "460", "460", "153…
## $ start_lat          <dbl> 41.9363, 41.8842, 41.8842, 41.9542, 41.8675, 41.967…
## $ start_lng          <dbl> -87.6527, -87.6296, -87.6296, -87.6544, -87.6322, -…
## $ end_lat            <dbl> 41.9561, 41.8842, 41.8763, 41.9528, 41.8672, 41.983…
## $ end_lng            <dbl> -87.6688, -87.6296, -87.6292, -87.6500, -87.6260, -…
## $ member_casual      <chr> "casual", "member", "member", "member", "member", "…
```

#Create dataframe for Date-Time Analysis

```
#create new data frame to contain new columns
cyclistic_datetime <- all_trips
```

```
#calculate ride length by subtracting ended_at time from started_at time and converted it to minutes
cyclistic_datetime$ride_length <- difftime(all_trips$ended_at, all_trips$started_at, units = "mins")
```

```
#create columns: day of week, month, day, year, time, hour
cyclistic_datetime$date <- as.Date(cyclistic_datetime$started_at) #default format is yyyy-mm-dd, use start date
cyclistic_datetime$day_of_week <- wday(all_trips$started_at) #calculate the day of the week
cyclistic_datetime$day_of_week <- format(as.Date(cyclistic_datetime$date), "%a")
cyclistic_datetime$month <- format(as.Date(cyclistic_datetime$date), "%m")#create column for month
cyclistic_datetime$day <- format(as.Date(cyclistic_datetime$date), "%d") #create column for day
cyclistic_datetime$year <- format(as.Date(cyclistic_datetime$date), "%Y") #create column for year
cyclistic_datetime$time <- format(as.Date(cyclistic_datetime$date), "%H:%M:%S") #format time as HH:MM:SS
cyclistic_datetime$time <- as_hms((cyclistic_datetime$started_at)) #create new column for time
cyclistic_datetime$hour <- hour(cyclistic_datetime$time) #create new column for hour
```

```
#create column for different seasons: Spring, Summer, Fall, Winter
cyclistic_datetime <-cyclistic_datetime %>% mutate(season =
                                             case_when(month == "03" ~ "Spring",
                                                       month == "04" ~ "Spring",
                                                       month == "05" ~ "Spring",
                                                       month == "06"  ~ "Summer",
                                                       month == "07"  ~ "Summer",
                                                       month == "08"  ~ "Summer",
                                                       month == "09" ~ "Fall",
                                                       month == "10" ~ "Fall",
                                                       month == "11" ~ "Fall",
                                                       month == "12" ~ "Winter",
                                                       month == "01" ~ "Winter",
                                                       month == "02" ~ "Winter")
                                                    )
```

```
#create column for different time_of_day: Night, Morning, Afternoon, Evening
cyclistic_datetime <-cyclistic_datetime %>% mutate(time_of_day =
                                             case_when(hour == "0" ~ "Night",
                                                       hour == "1" ~ "Night",
                                                       hour == "2" ~ "Night",
                                                       hour == "3" ~ "Night",
                                                       hour == "4" ~ "Night",
                                                       hour == "5" ~ "Night",
                                                       hour == "6" ~ "Morning",
                                                       hour == "7" ~ "Morning",
                                                       hour == "8" ~ "Morning",
                                                       hour == "9" ~ "Morning",
                                                       hour == "10" ~ "Morning",
                                                       hour == "11" ~ "Morning",
                                                       hour == "12" ~ "Afternoon",
                                                       hour == "13" ~ "Afternoon",
                                                       hour == "14" ~ "Afternoon",
                                                       hour == "15" ~ "Afternoon",
                                                       hour == "16" ~ "Afternoon",
                                                       hour == "17" ~ "Afternoon",
                                                       hour == "18" ~ "Evening",
                                                       hour == "19" ~ "Evening",
                                                       hour == "20" ~ "Evening",
                                                       hour == "21" ~ "Evening",
                                                       hour == "22" ~ "Evening",
                                                       hour == "23" ~ "Evening")
                                                      )
```

```
#create a column for the month using the full month name
cyclistic_datetime <-cyclistic_datetime %>% mutate(month =
                                             case_when(month == "01" ~ "January",
                                                       month == "02" ~ "February",
                                                       month == "03" ~ "March",
                                                       month == "04" ~ "April",
                                                       month == "05" ~ "May",
                                                       month == "06" ~ "June",
                                                       month == "07" ~ "July",
                                                       month == "08" ~ "August",
                                                       month == "09" ~ "September",
                                                       month == "10" ~ "October",
                                                       month == "11" ~ "November",
                                                       month == "12" ~ "December"
                                                       )
                                            )
```

#View Uncleaned Data

```
#----------View UnCleaned Data
head(cyclistic_datetime)
```

```
## # A tibble: 6 × 23
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 1068AB1B8F12F… docked… 2020-01-01 00:04:00 2020-01-01 00:17:00 Sheffi… 115
## 2 DCF74A0EB3284… docked… 2020-01-01 00:10:00 2020-01-01 00:10:00 Daley … 81
## 3 4DE50A4FC7687… docked… 2020-01-01 00:11:00 2020-01-01 00:15:00 Daley … 81
## 4 1C78B5F337CBF… docked… 2020-01-01 00:11:00 2020-01-01 00:13:00 Sherid… 240
## 5 D231CE7990A3A… docked… 2020-01-01 00:12:00 2020-01-01 00:14:00 Delano… 626
## 6 EB21BB139ABF6… docked… 2020-01-01 00:21:00 2020-01-01 00:29:00 Clark … 326
## # … with 17 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, ride_length <drtn>, date <date>, day_of_week <chr>,
## #   month <chr>, day <chr>, year <chr>, time <time>, hour <int>, season <chr>,
## #   time_of_day <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

#Data cleaning

```
#clean the data
cyclistic_datetime <- cyclistic_datetime %>% rename(bike_type = rideable_type) #Renaming Colu
mns for more understandability
cyclistic_datetime <- na.omit(cyclistic_datetime) #remove rows with NA values
cyclistic_datetime <- distinct(cyclistic_datetime) #remove duplicate rows
cyclistic_datetime <- cyclistic_datetime[!(cyclistic_datetime$ride_length <=0),] #remove wher
e ride_length is 0 or negative
cyclistic_datetime <- cyclistic_datetime %>%  #remove columns not needed: ride_id, start_stat
ion_id, end_station_id, start_lat, start_long, end_lat, end_lng
  select(-c(ride_id, start_station_id, end_station_id))
#Data cleaned Up
message("Cleaned Up ", nrow(all_trips)-nrow(cyclistic_datetime), " Rows")
```

```
## Cleaned Up 1855610 Rows
```

#Final Data

```
#view the final data
str(cyclistic_datetime)
```

```
## tibble [10,383,350 × 20] (S3: tbl_df/tbl/data.frame)
##  $ bike_type         : chr [1:10383350] "docked_bike" "docked_bike" "docked_bike" "docked_
bike" ...
##  $ started_at        : POSIXct[1:10383350], format: "2020-01-01 00:04:00" "2020-01-01 00:1
1:00" ...
##  $ ended_at          : POSIXct[1:10383350], format: "2020-01-01 00:17:00" "2020-01-01 00:1
5:00" ...
##  $ start_station_name: chr [1:10383350] "Sheffield Ave & Wellington Ave" "Daley Center Pla
za" "Sheridan Rd & Irving Park Rd" "Delano Ct & Roosevelt Rd" ...
##  $ end_station_name  : chr [1:10383350] "Ashland Ave & Belle Plaine Ave" "Dearborn St & Va
n Buren St" "Broadway & Sheridan Rd" "Wabash Ave & Roosevelt Rd" ...
##  $ start_lat         : num [1:10383350] 41.9 41.9 42 41.9 42 ...
##  $ start_lng         : num [1:10383350] -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ end_lat           : num [1:10383350] 42 41.9 42 41.9 42 ...
##  $ end_lng           : num [1:10383350] -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ member_casual     : chr [1:10383350] "casual" "member" "member" "member" ...
##  $ ride_length       : 'difftime' num [1:10383350] 13 4 2 2 ...
##   ..- attr(*, "units")= chr "mins"
##  $ date              : Date[1:10383350], format: "2020-01-01" "2020-01-01" ...
##  $ day_of_week       : chr [1:10383350] "Wed" "Wed" "Wed" "Wed" ...
##  $ month             : chr [1:10383350] "January" "January" "January" "January" ...
##  $ day               : chr [1:10383350] "01" "01" "01" "01" ...
##  $ year              : chr [1:10383350] "2020" "2020" "2020" "2020" ...
##  $ time              : 'hms' num [1:10383350] 00:04:00 00:11:00 00:11:00 00:12:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ hour              : int [1:10383350] 0 0 0 0 0 0 0 0 0 0 ...
##  $ season            : chr [1:10383350] "Winter" "Winter" "Winter" "Winter" ...
##  $ time_of_day       : chr [1:10383350] "Night" "Night" "Night" "Night" ...
##  - attr(*, "na.action")= 'omit' Named int [1:1837859] 391206 428166 430966 431017 432071 4
32282 432425 433347 440617 441891 ...
##   ..- attr(*, "names")= chr [1:1837859] "391206" "428166" "430966" "431017" ...
```

```
glimpse(cyclistic_datetime)
```

```
## Rows: 10,383,350
## Columns: 20
## $ bike_type          <chr> "docked_bike", "docked_bike", "docked_bike", "docke…
## $ started_at         <dttm> 2020-01-01 00:04:00, 2020-01-01 00:11:00, 2020-01-…
## $ ended_at           <dttm> 2020-01-01 00:17:00, 2020-01-01 00:15:00, 2020-01-…
## $ start_station_name <chr> "Sheffield Ave & Wellington Ave", "Daley Center Pla…
## $ end_station_name   <chr> "Ashland Ave & Belle Plaine Ave", "Dearborn St & Va…
## $ start_lat          <dbl> 41.9363, 41.8842, 41.9542, 41.8675, 41.9671, 41.967…
## $ start_lng          <dbl> -87.6527, -87.6296, -87.6544, -87.6322, -87.6674, -…
## $ end_lat            <dbl> 41.9561, 41.8763, 41.9528, 41.8672, 41.9836, 41.983…
## $ end_lng            <dbl> -87.6688, -87.6292, -87.6500, -87.6260, -87.6692, -…
## $ member_casual      <chr> "casual", "member", "member", "member", "member", "…
## $ ride_length        <drtn> 13 mins, 4 mins, 2 mins, 2 mins, 8 mins, 8 mins, 1…
## $ date               <date> 2020-01-01, 2020-01-01, 2020-01-01, 2020-01-01, 20…
## $ day_of_week        <chr> "Wed", "Wed", "Wed", "Wed", "Wed", "Wed", "Wed", "W…
## $ month              <chr> "January", "January", "January", "January", "Januar…
## $ day                <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01…
## $ year               <chr> "2020", "2020", "2020", "2020", "2020", "2020", "20…
## $ time               <time> 00:04:00, 00:11:00, 00:11:00, 00:12:00, 00:21:00, …
## $ hour               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ season             <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "…
## $ time_of_day        <chr> "Night", "Night", "Night", "Night", "Night", "Night…
```

#ANALYSIS

```
#-----------------------------------------TOTAL RIDES--------------------------------------

#total number of rides
nrow(cyclistic_datetime)
```

```
## [1] 10383350
```

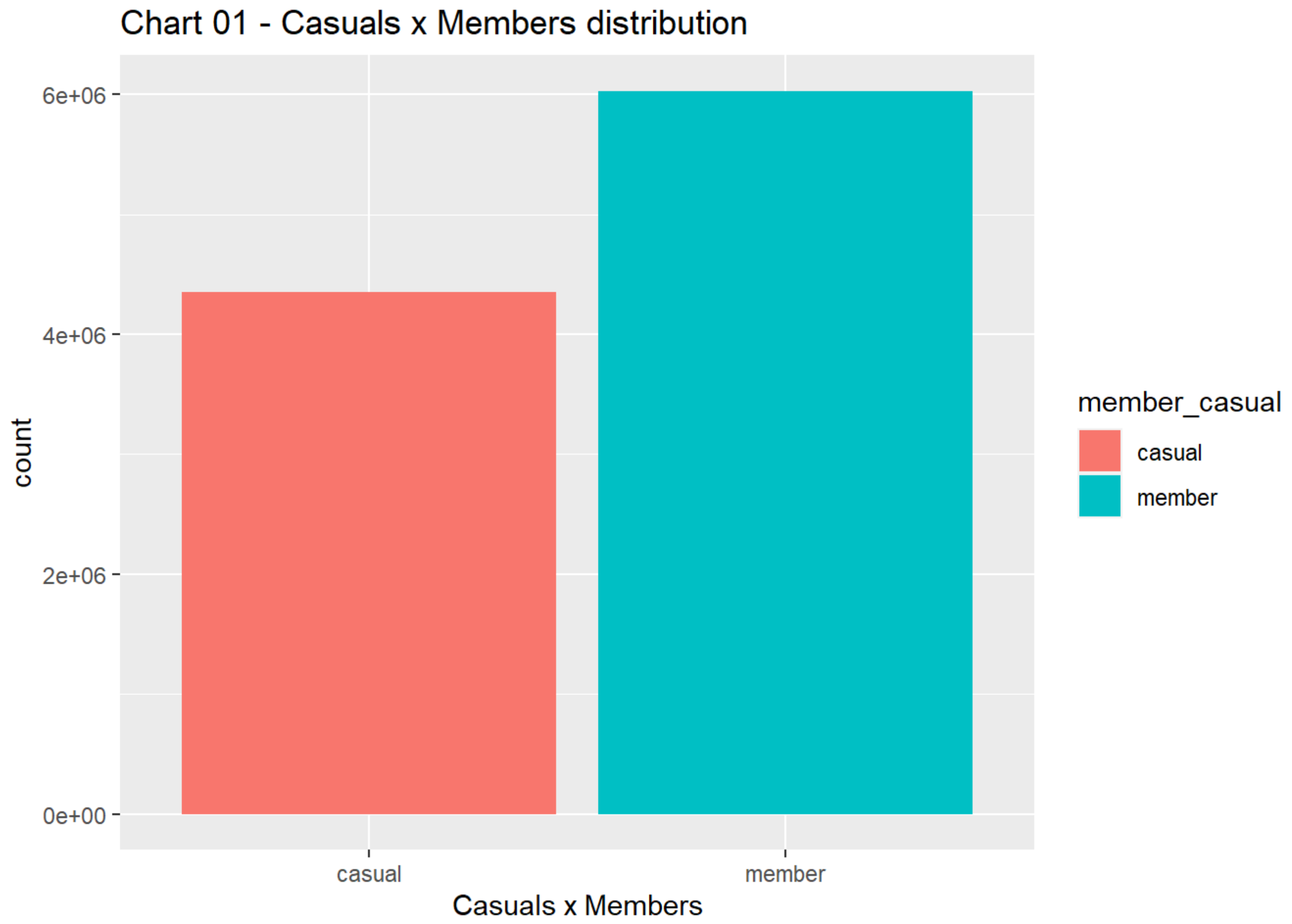
#Data distribution Here we want to try to answer the most basic questions about how the data is distributed.

##Casuals vs members How much of the data is about members and how much is about casuals?

```
#-----------------MEMBER TYPE---------------------
cyclistic_datetime %>%
  group_by(member_casual) %>%
  count(member_casual)
```

```
## # A tibble: 2 × 2
## # Groups:   member_casual [2]
##   member_casual       n
##   <chr>           <int>
## 1 casual        4352711
## 2 member        6030639
```

```
#-----------------Plot MEMBER TYPE---------------------
ggplot(cyclistic_datetime, aes(member_casual, fill=member_casual)) +
  geom_bar() +
  labs(x="Casuals x Members", title="Chart 01 - Casuals x Members distribution")
```



```
#----------------TYPE OF BIKE---------------------

#total rides by member type
cyclistic_datetime %>%
  group_by(member_casual, bike_type) %>%
  count(bike_type)
```

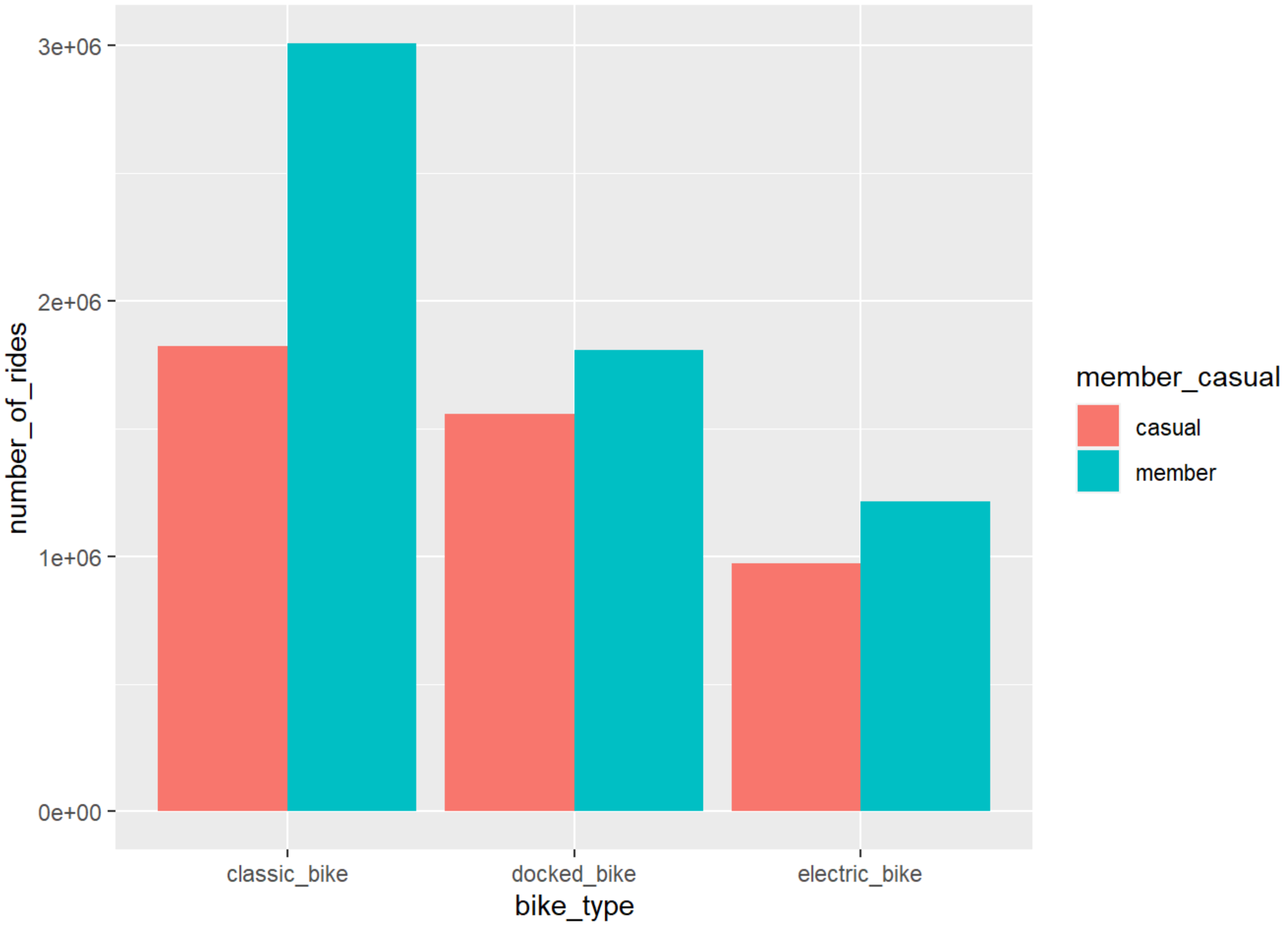
```
## # A tibble: 6 × 3
## # Groups:   member_casual, bike_type [6]
##   member_casual bike_type           n
##   <chr>         <chr>           <int>
## 1 casual        classic_bike  1821173
## 2 casual        docked_bike   1558503
## 3 casual        electric_bike  973035
## 4 member        classic_bike  3008012
## 5 member        docked_bike   1807845
## 6 member        electric_bike 1214782
```

#Grouped barchart ###Ref: https://r-graph-gallery.com/48-grouped-barplot-with-ggplot2.html (https://r-graph-gallery.com/48-grouped-barplot-with-ggplot2.html)

Plotting Bar chart by Grouping bike_types w.r.t Member_casual

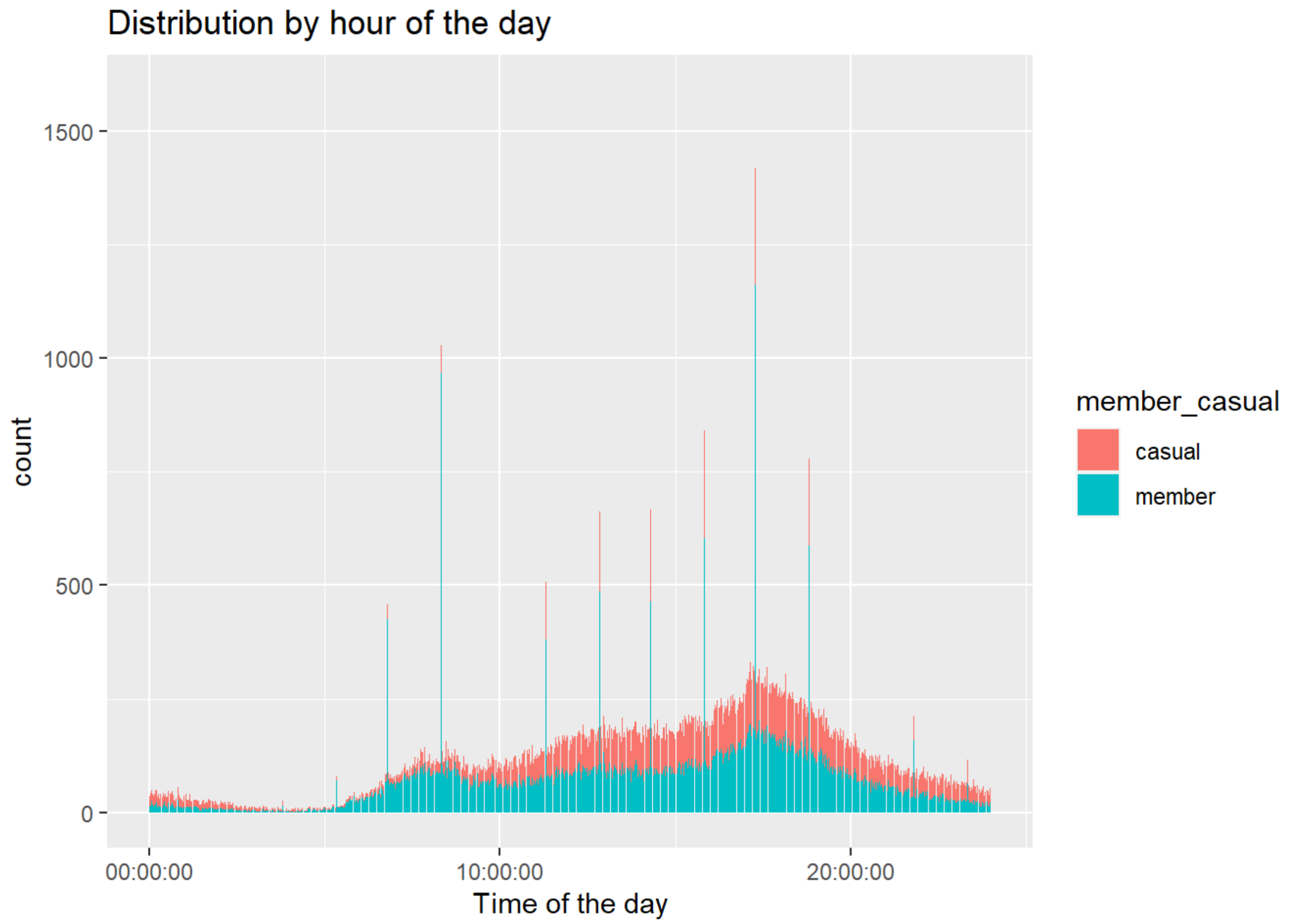
```
#Plot
cyclistic_datetime %>%
  group_by(member_casual, bike_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = bike_type, y = number_of_rides, fill = member_casual)) +
  geom_bar(position = "dodge", stat='identity')
```

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

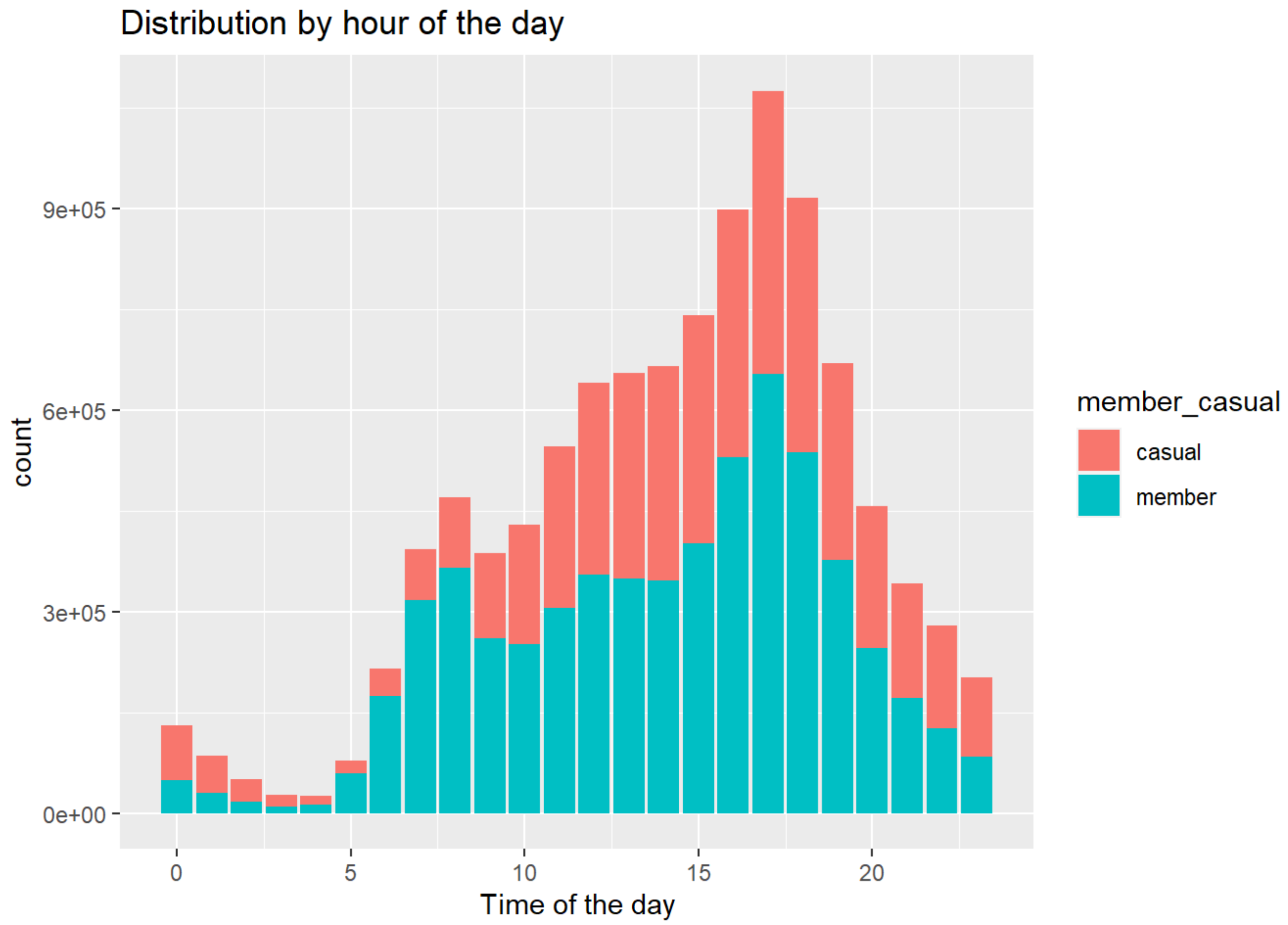


#Plotting - Distribution by time of the day

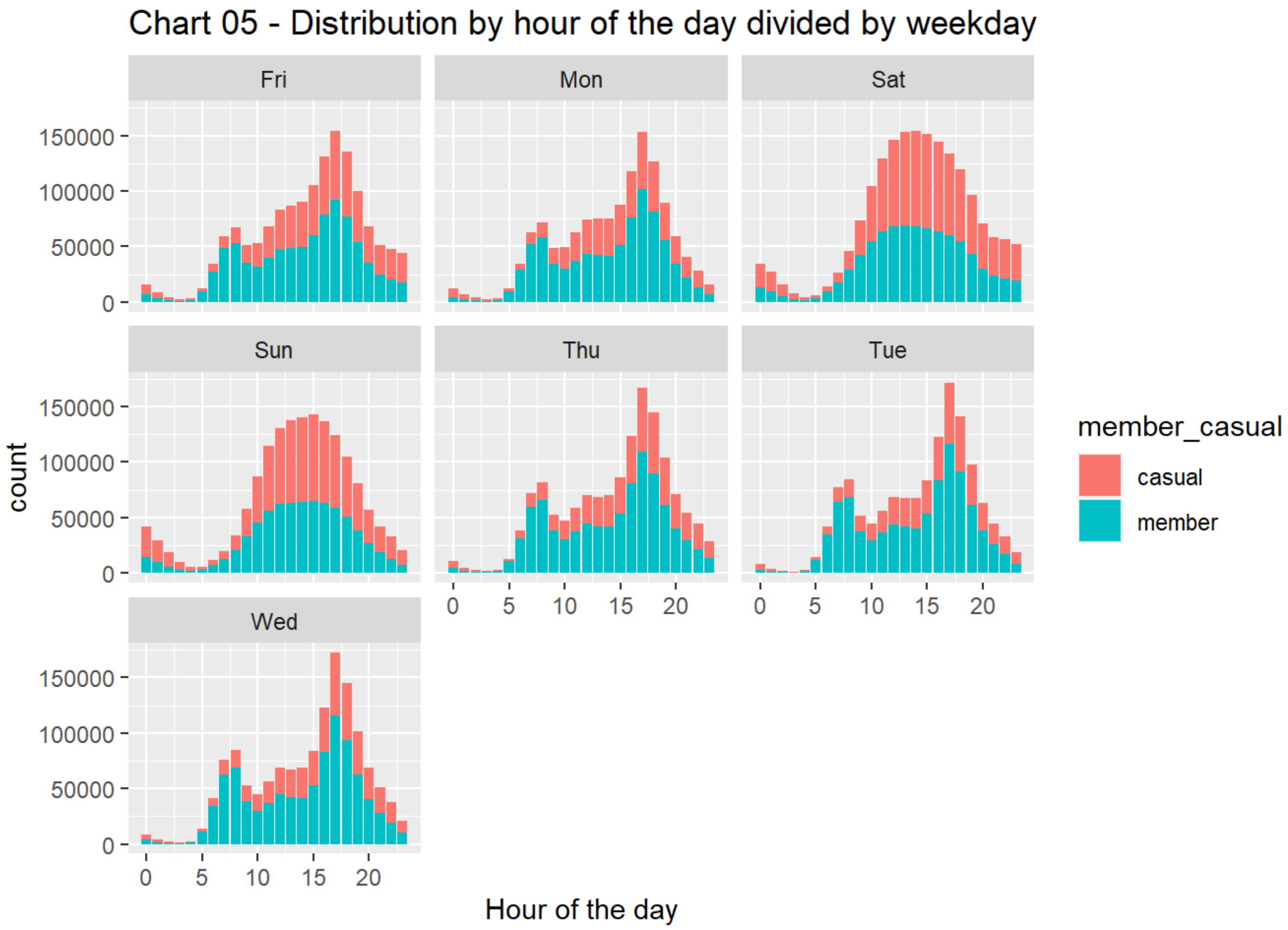
```
cyclistic_datetime %>%
    ggplot(aes(time, fill=member_casual)) +
    labs(x="Time of the day", title="Distribution by hour of the day") +
    geom_bar()
```



```
cyclistic_datetime %>%
    ggplot(aes(hour, fill=member_casual)) +
    labs(x="Time of the day", title="Distribution by hour of the day") +
    geom_bar()
```



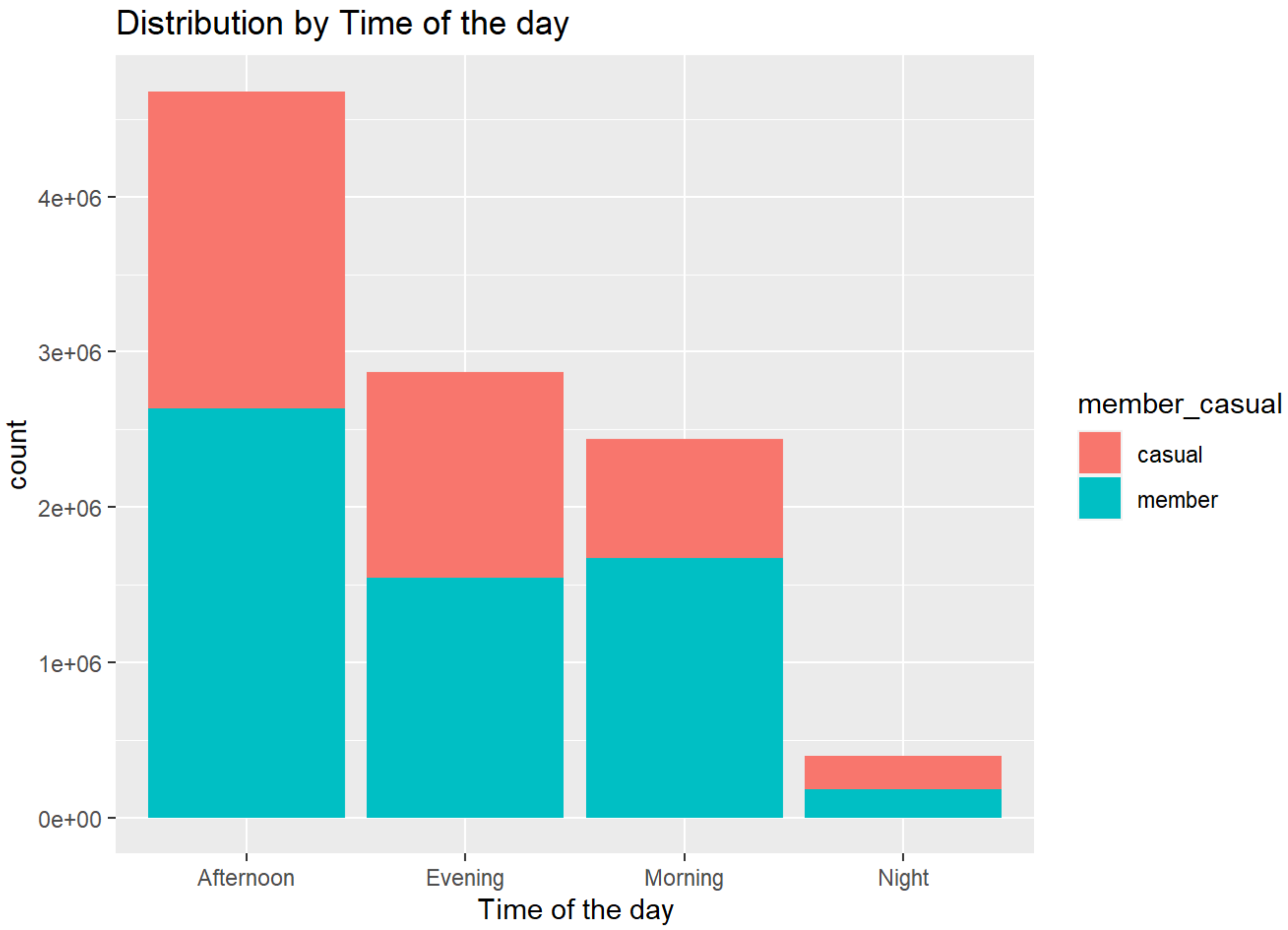
```
cyclistic_datetime %>%
    ggplot(aes(hour, fill=member_casual)) +
    geom_bar() +
    labs(x="Hour of the day", title="Chart 05 - Distribution by hour of the day divided by we
ekday") +
    facet_wrap(~ day_of_week)
```



```
#----------------------TIME OF DAY----------------------

#total rides by member type
cyclistic_datetime %>%
  group_by(member_casual) %>%
  count(time_of_day)
```

```
## # A tibble: 8 × 3
## # Groups:   member_casual [2]
##   member_casual time_of_day       n
##   <chr>         <chr>         <int>
## 1 casual        Afternoon   2041858
## 2 casual        Evening     1324899
## 3 casual        Morning      768221
## 4 casual        Night        217733
## 5 member        Afternoon   2637332
## 6 member        Evening     1541822
## 7 member        Morning     1672328
## 8 member        Night        179157
```

```
cyclistic_datetime %>%
    ggplot(aes(time_of_day, fill=member_casual)) +
    labs(x="Time of the day", title="Distribution by Time of the day") +
    geom_bar()
```



```
#----------------DAY OF THE WEEK------------------

#total rides by member type
cyclistic_datetime %>%
  group_by(member_casual) %>%
  count(day_of_week)
```

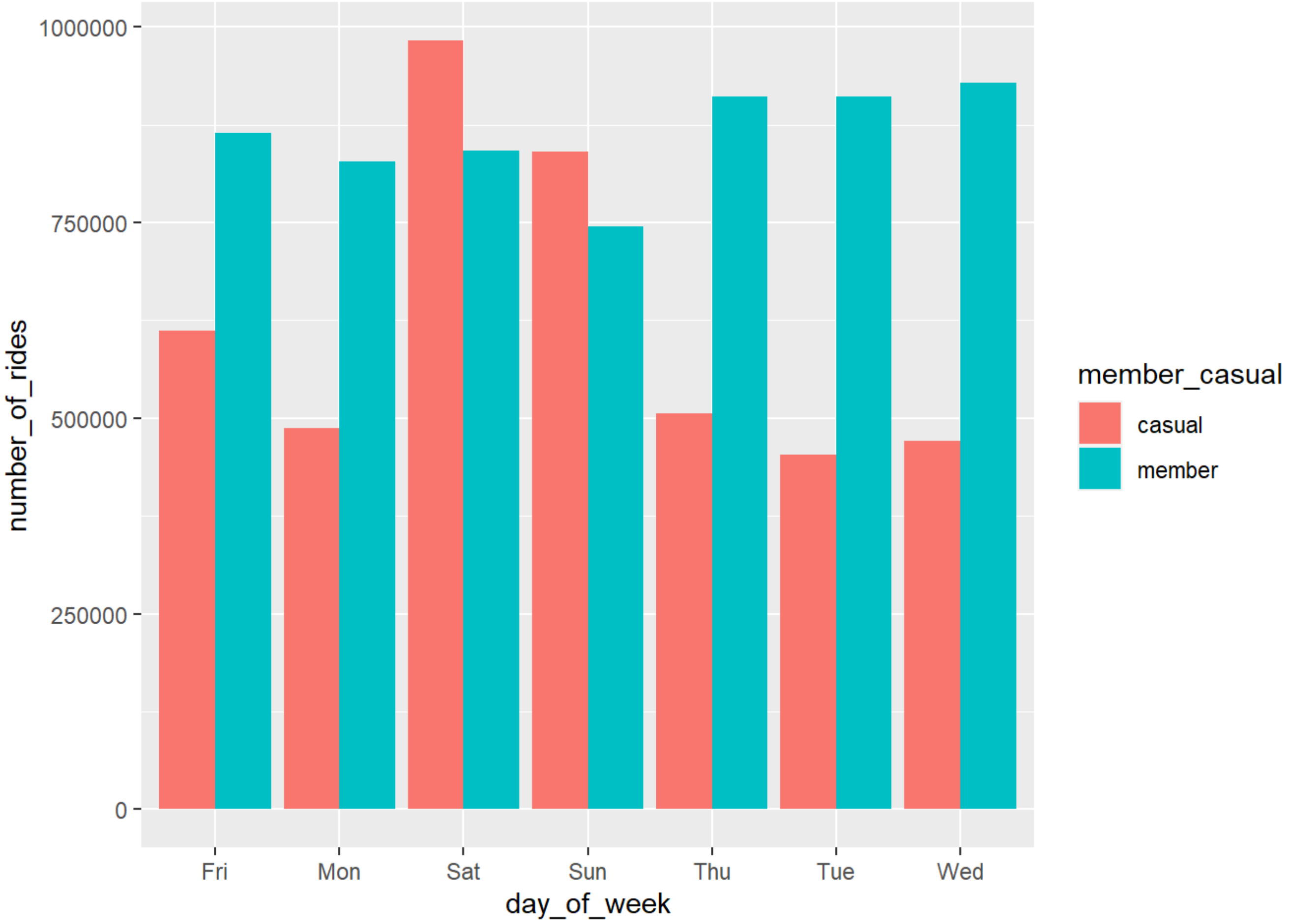
```
## # A tibble: 14 × 3
## # Groups:   member_casual [2]
##    member_casual day_of_week      n
##    <chr>         <chr>        <int>
##  1 casual        Fri         611913
##  2 casual        Mon         486624
##  3 casual        Sat         982576
##  4 casual        Sun         840629
##  5 casual        Thu         506702
##  6 casual        Tue         453683
##  7 casual        Wed         470584
##  8 member        Fri         864933
##  9 member        Mon         828313
## 10 member        Sat         842151
## 11 member        Sun         744485
## 12 member        Thu         910957
## 13 member        Tue         911500
## 14 member        Wed         928300
```

```
#total rides
cyclistic_datetime %>%
  count(day_of_week)
```

```
## # A tibble: 7 × 2
##   day_of_week       n
##   <chr>         <int>
## 1 Fri         1476846
## 2 Mon         1314937
## 3 Sat         1824727
## 4 Sun         1585114
## 5 Thu         1417659
## 6 Tue         1365183
## 7 Wed         1398884
```

```
cyclistic_datetime %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(day_of_week)%>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

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



```
#---------------------MONTH-----------------------

#total rides by member type
cyclistic_datetime %>%
  group_by(member_casual) %>%
  count(month) %>%
  print(n = 24) #lets you view the entire tibble
```

```
## # A tibble: 24 × 3
## # Groups:   member_casual [2]
##    member_casual month          n
##    <chr>         <chr>      <int>
##  1 casual        April     235803
##  2 casual        August    623408
##  3 casual        December   69568
##  4 casual        February   36082
##  5 casual        January    35049
##  6 casual        July      949153
##  7 casual        June      750566
##  8 casual        March     167505
##  9 casual        May       523748
## 10 casual        November  142805
## 11 casual        October   311432
## 12 casual        September 507592
## 13 member        April     418977
## 14 member        August    656621
## 15 member        December  220340
## 16 member        February  234265
## 17 member        January   271673
## 18 member        July      934386
## 19 member        June      820548
## 20 member        March     393685
## 21 member        May       629518
## 22 member        November  334978
## 23 member        October   503909
## 24 member        September 611739
```

```
#total rides
cyclistic_datetime %>%
  count(month)
```

```
## # A tibble: 12 × 2
##    month           n
##    <chr>       <int>
##  1 April      654780
##  2 August    1280029
##  3 December   289908
##  4 February   270347
##  5 January    306722
##  6 July      1883539
##  7 June      1571114
##  8 March      561190
##  9 May       1153266
## 10 November   477783
## 11 October    815341
## 12 September 1119331
```

#Plotting Season Data

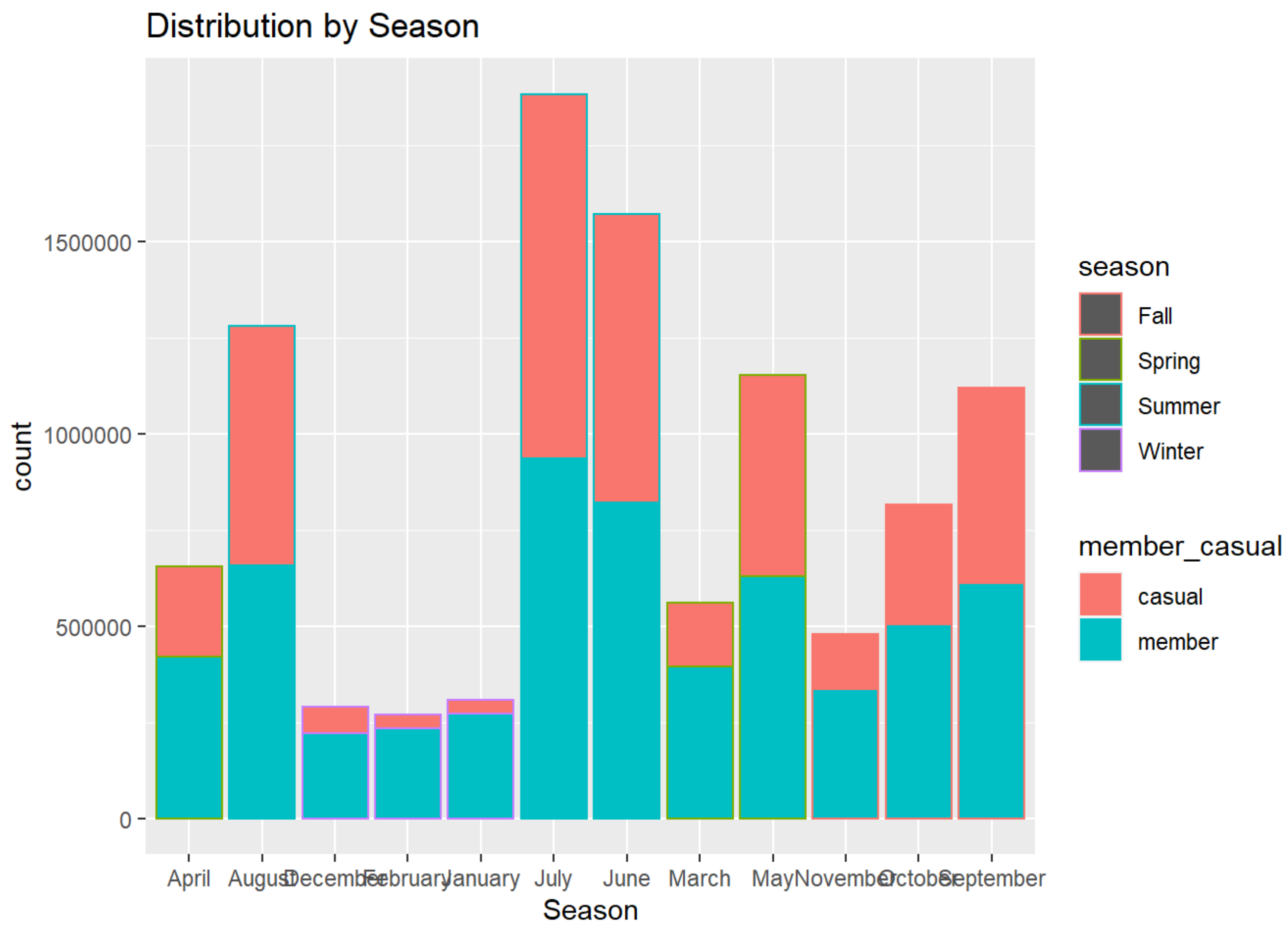
```
#--------------------SEASON----------------------

#-----all seasons-------

#total rides by member type
cyclistic_datetime %>%
  group_by(season, member_casual) %>%
  count(season)
```

```
## # A tibble: 8 × 3
## # Groups:   season, member_casual [8]
##   season member_casual       n
##   <chr>  <chr>           <int>
## 1 Fall   casual         961829
## 2 Fall   member        1450626
## 3 Spring casual         927056
## 4 Spring member        1442180
## 5 Summer casual        2323127
## 6 Summer member        2411555
## 7 Winter casual         140699
## 8 Winter member         726278
```

```
#total rides
cyclistic_datetime %>%
  group_by(season) %>%
  count(season)
```

```
## # A tibble: 4 × 2
## # Groups:   season [4]
##   season       n
##   <chr>    <int>
## 1 Fall   2412455
## 2 Spring 2369236
## 3 Summer 4734682
## 4 Winter  866977
```

```
cyclistic_datetime %>%
    ggplot(aes(month, fill=member_casual, color=season)) +
    labs(x="Season", title="Distribution by Season") +
    geom_bar()
```


Distribution by Season
count
April
August
December
February
January
July
June
March
May
November
October
September
Season
season
Fall
Spring
Summer
Winter
member_casual
casual
member