Bike Rides - Google Data Analytics Capstone

2022-05-07

```
getwd()
```

```
## [1] "/Users/jw3/Downloads/Cycle Case Study"
```

```
setwd("~/Downloads/Cycle Case Study/CSV")
```

Background Information

The marketing analyst team wants to understand the riding habits of casual riders and annual members to design a new marketing strategy that will convert casual riders into annual members.

This case study will answer the following key business question:

How do annual members and casual riders use bikes differently?

The six steps of the data analysis process will be used as a guideline : Ask, Prepare, Process, Analyze, Share and Act.

1. ASK

A statement of the business task: Because annual members are more profitable than casual riders, it is necessary to develop a marketing strategies that will convert casual riders into annual members.

The key stakeholders are: 1-Lily Moreno: Manager and director of marketing. 2-The executive team: The executives must approve of the recommendations from this case study.

2. PREPARE

12 months of bike share data (05/2021-04/2022) was extracted as 12 .csv files.

Data Organization & Description: File naming convention: YYYY_MM File Type: csv format

3. PROCESS

I used RStudio for data verification and cleaning because of the file size.

Setting Up the Environment - Loading library

```
library(janitor)
```

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

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

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## 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(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(latexpdf)
```

Create dataframes

12 months of bike share data (05/2021-04/2022) was extracted as 12 .csv files.

Data Organization & Description: File naming convention: YYYY_MM File Type: csv format

```
DF1 <- read_csv("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.
```

```
DF2 <- read_csv("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.
```

```
DF3 <- read_csv("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.
```

```
DF4 <- read_csv("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.
```

```
DF5 <- read_csv("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.
```

```
DF6 <- read_csv("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.
```

```
DF7 <- read_csv("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.
```

```
DF8 <- read_csv("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.
```

```
DF9 <- read_csv("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.
```

```
DF10 <- read_csv("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.
```

```
DF11 <- read_csv("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.
```

```
DF12 <- read_csv("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.
```

Combine dataframes into one dataset

```
bike_rides <- rbind(DF1,DF2,DF3,DF4,DF5,DF6,DF7,DF8,DF9,DF10,DF11,DF12)
```

Summary of dataset

```
summary(bike_rides)
```

```
##    ride_id          rideable_type        started_at
##  Length:5757551     Length:5757551     Min.   :2021-05-01 00:00:11.00
##  Class :character   Class :character   1st Qu.:2021-07-07 14:52:45.00
##  Mode  :character   Mode  :character   Median :2021-08-31 17:17:20.00
##                                        Mean   :2021-09-18 18:21:45.73
##                                        3rd Qu.:2021-11-03 20:25:37.50
##                                        Max.   :2022-04-30 23:59:54.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-05-01 00:03:26.00   Length:5757551     Length:5757551
##  1st Qu.:2021-07-07 15:16:14.50   Class :character   Class :character
##  Median :2021-08-31 17:34:09.00   Mode  :character   Mode  :character
##  Mean   :2021-09-18 18:42:54.02
##  3rd Qu.:2021-11-03 20:38:44.50
##  Max.   :2022-05-02 00:35:01.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5757551     Length:5757551     Min.   :41.64   Min.   :-87.84
##  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:5757551
##  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.   :42.17   Max.   :-87.49
##  NA's   :4766    NA's   :4766
```

```
 colnames(bike_rides)  #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"
```

```
 dim(bike_rides)  #Dimensions of the data frame
```

```
## [1] 5757551      13
```

```
 head(bike_rides)  #See the first 6 rows of data frame
```

```
## # A tibble: 6 x 13
```

```
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 C809ED~ electric_bike 2021-05-30 11:58:15 2021-05-30 12:10:39 <NA>
## 2 DD59FD~ electric_bike 2021-05-30 11:29:14 2021-05-30 12:14:09 <NA>
## 3 0AB83C~ electric_bike 2021-05-30 14:24:01 2021-05-30 14:25:13 <NA>
## 4 7881AC~ electric_bike 2021-05-30 14:25:51 2021-05-30 14:41:04 <NA>
## 5 853FA7~ electric_bike 2021-05-30 18:15:39 2021-05-30 18:22:32 <NA>
## 6 F5E63D~ electric_bike 2021-05-30 11:33:41 2021-05-30 11:57:17 <NA>
## # ... with 8 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>
```

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

```
## spec_tbl_df [5,757,551 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5757551] "C809ED75D6160B2A" "DD59FDCE0ACACAF3" "0AB83CB88C43EFC2" "788
##  $ rideable_type     : chr [1:5757551] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:5757551], format: "2021-05-30 11:58:15" "2021-05-30 11:29:14" ...
##  $ ended_at          : POSIXct[1:5757551], format: "2021-05-30 12:10:39" "2021-05-30 12:14:09" ...
##  $ start_station_name: chr [1:5757551] NA NA NA NA ...
##  $ start_station_id  : chr [1:5757551] NA NA NA NA ...
##  $ end_station_name  : chr [1:5757551] NA NA NA NA ...
##  $ end_station_id    : chr [1:5757551] NA NA NA NA ...
##  $ start_lat         : num [1:5757551] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5757551] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5757551] 41.9 41.8 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5757551] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5757551] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Cleaning Data

Remove Duplicate Riders by Ride ID and Print the Results

```
bike_rides_v2 <- bike_rides[!duplicated(bike_rides$ride_id), ]
print(paste(nrow(bike_rides) - nrow(bike_rides_v2), "duplicated rows removed"))
```

```
## [1] "0 duplicated rows removed"
```

Date/Time Data Cleaning

```
bike_rides_v2$started_at <- as.POSIXct(bike_rides_v2$started_at, "%Y-%m-%d %H:%M:%S", TZ="GMT")
```

```
bike_rides_v2$ended_at <- as.POSIXct(bike_rides_v2$ended_at, "%Y-%m-%d %H:%M:%S", TZ="GMT")
```

```
bike_rides_v2 <- bike_rides_v2 %>%
mutate(ride_time_m = as.numeric(bike_rides_v2$ended_at - bike_rides_v2$started_at) / 60)
summary(bike_rides_v2$ride_time_m)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   -58.03     6.45    11.52    21.14    20.95 55944.15
```

Get rid of scientific notation

```
options(scipen=999)
```

4. ANALYZE

Casuals v Members Distribution

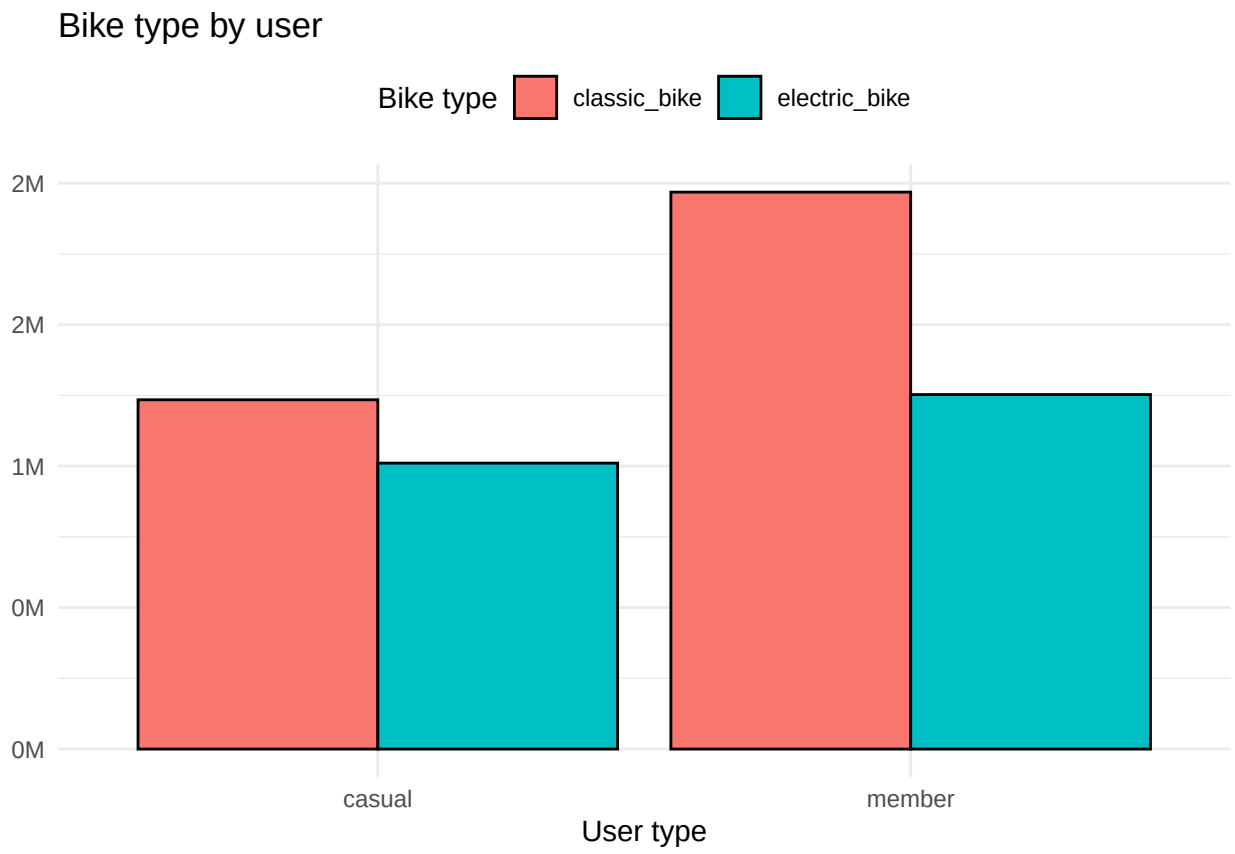
```
ggplot(bike_rides_v2, aes(member_casual, fill=member_casual)) +
    geom_bar(color = "black") +
    labs(x="Casual vs Members",y=NULL, title="Casual vs Members Distribution") +
    theme_minimal() +
    scale_y_continuous(labels  =
    label_number(scale = 1e-6, prefix = "", suffix = "M", accuracy = 1))
```

Type of Bike

Create data frame filtering the "bike type" column:

```
with_bike_type <- bike_rides_v2 %>% filter(rideable_type=="classic_bike" | rideable_type=="electric_bik
```

```
with_bike_type %>%
    group_by(member_casual,rideable_type) %>%
    summarise(totals=n(), .groups="drop")  %>%
ggplot()+
    geom_col(color = "black", aes(x=member_casual,y=totals,fill=rideable_type), position = "dodge") +
    labs(title = "Bike type by user",x="User type",y=NULL, fill="Bike type") +
    theme_minimal() +
    theme(legend.position="top") +
    scale_y_continuous(labels  =
    label_number(scale = 1e-6, prefix = "", suffix = "M", accuracy = 1))
```



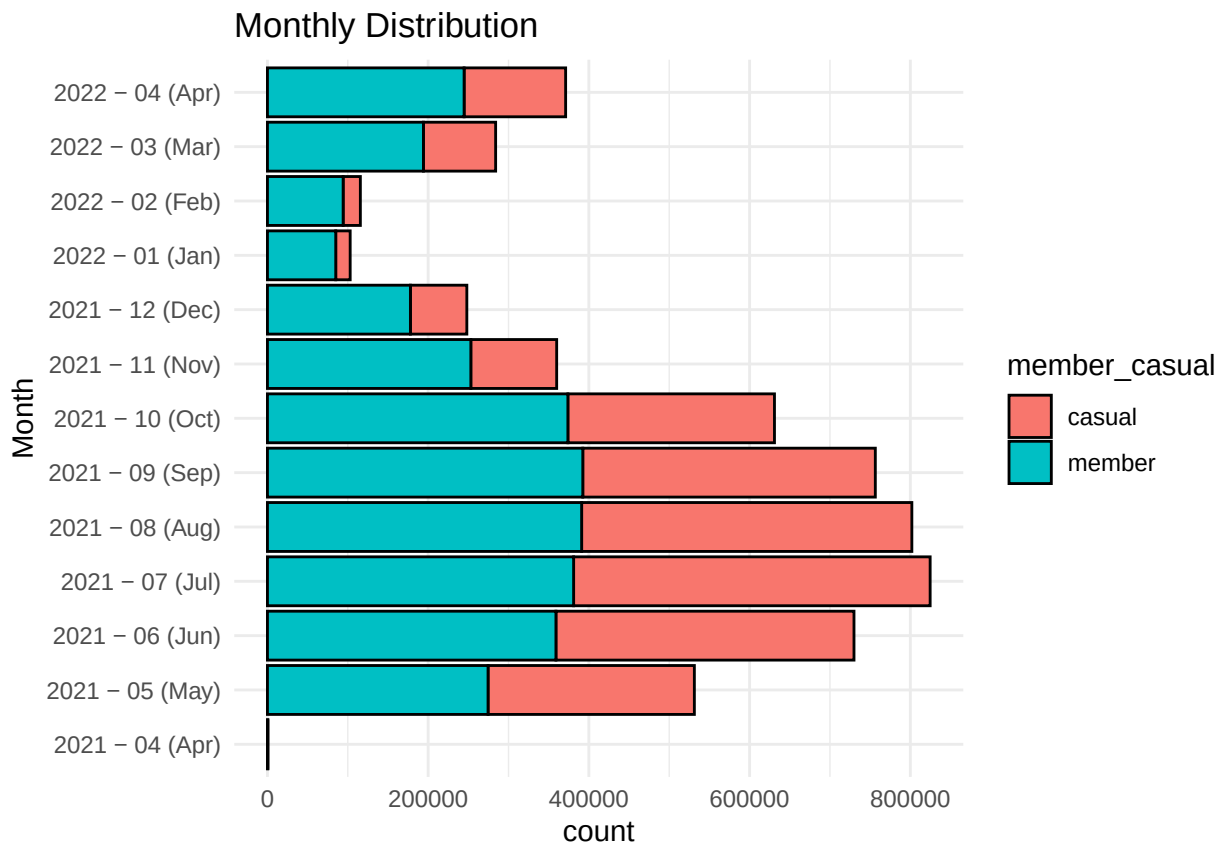
Year Month Data Manipultation

```
bike_rides_v2 <- bike_rides_v2  %>%
  mutate(year_month = paste(strftime(bike_rides_v2$started_at, "%Y"),
      "-",
      strftime(bike_rides_v2$started_at, "%m"),
      paste("(",strftime(bike_rides_v2$started_at, "%b"), ")", sep="")))
unique(bike_rides_v2$year_month)
```

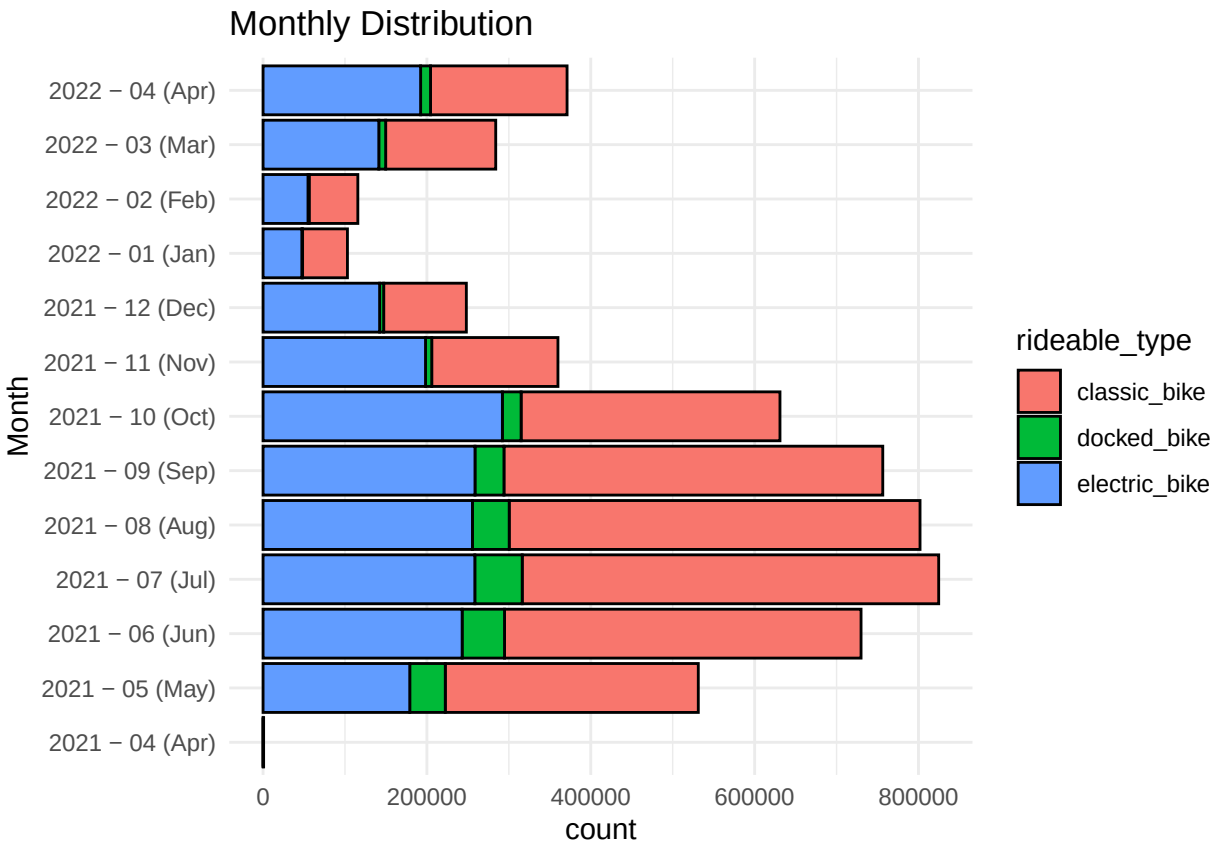
```
##  [1] "2021 - 05 (May)" "2021 - 04 (Apr)" "2021 - 06 (Jun)" "2021 - 07 (Jul)"
##  [5] "2021 - 08 (Aug)" "2021 - 09 (Sep)" "2021 - 10 (Oct)" "2021 - 11 (Nov)"
##  [9] "2021 - 12 (Dec)" "2022 - 01 (Jan)" "2022 - 02 (Feb)" "2022 - 03 (Mar)"
## [13] "2022 - 04 (Apr)"
```

Monthly Distribution Member v Casual

```
bike_rides_v2 %>%
  ggplot(aes(year_month, fill=member_casual)) +
  geom_bar(color = "black") +
  labs(x="Month", title="Monthly Distribution") +
  coord_flip()+
  theme_minimal()
```




```
bike_rides_v2 %>%
  group_by(member_casual,rideable_type) %>%
  ggplot(aes(year_month, fill=rideable_type)) +
  geom_bar(color = "black") +
  labs(x="Month", title="Monthly Distribution") +
  coord_flip()+
  theme_minimal()
```



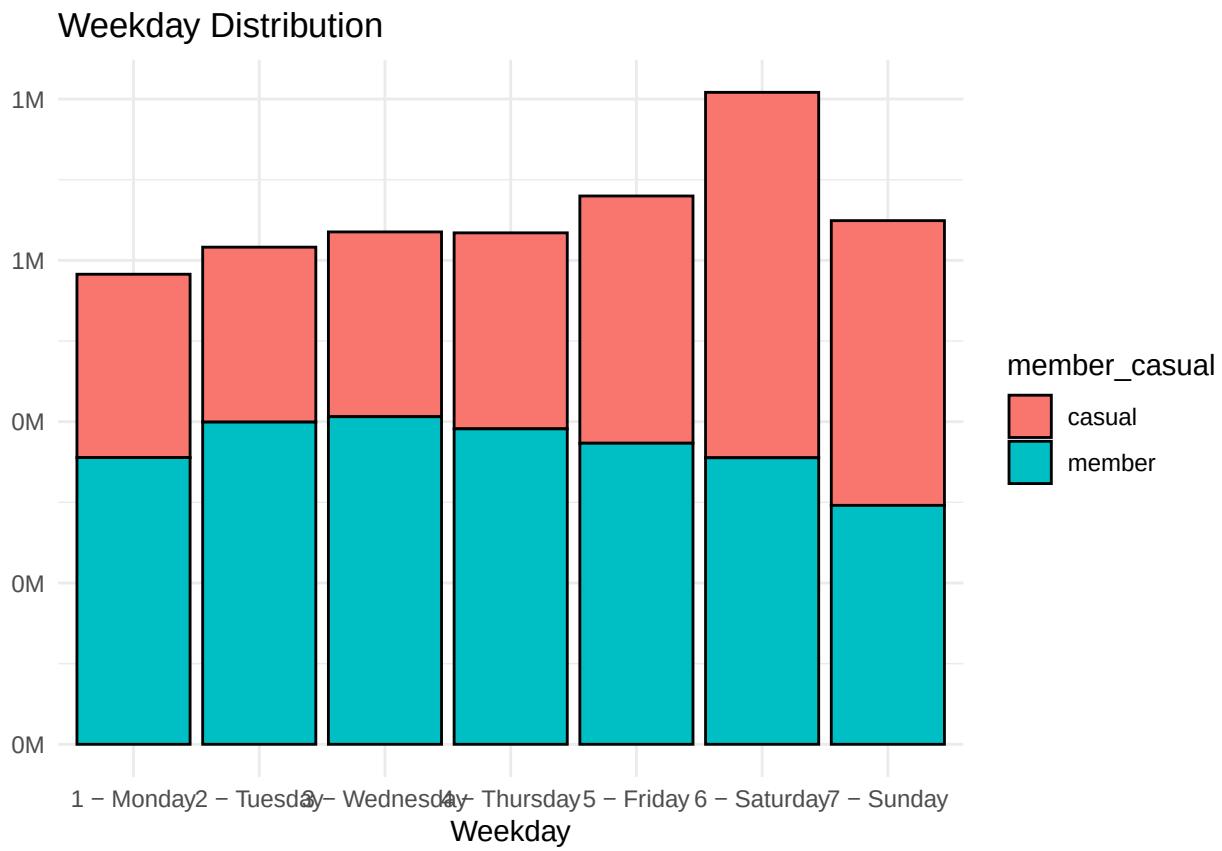
Weekday Data Manipulation

```
bike_rides_v2 <- bike_rides_v2 %>%
  mutate(weekday = paste(strftime(bike_rides_v2$ended_at, "%u"), "-", strftime(bike_rides_v2$ended_at,
unique(bike_rides_v2$weekday)
```

```
## [1] "7 - Sunday"    "3 - Wednesday" "2 - Tuesday"   "1 - Monday"
## [5] "6 - Saturday"  "4 - Thursday"  "5 - Friday"
```

Weekday Distribution

```
ggplot(bike_rides_v2, aes(weekday, fill=member_casual)) +
 geom_bar(color = "black") +
 labs(x="Weekday", y= NULL, title="Weekday Distribution") +
 theme_minimal() +
 scale_y_continuous(labels  =
 label_number(scale = 1e-6, prefix = "", suffix = "M", accuracy = 1))
```



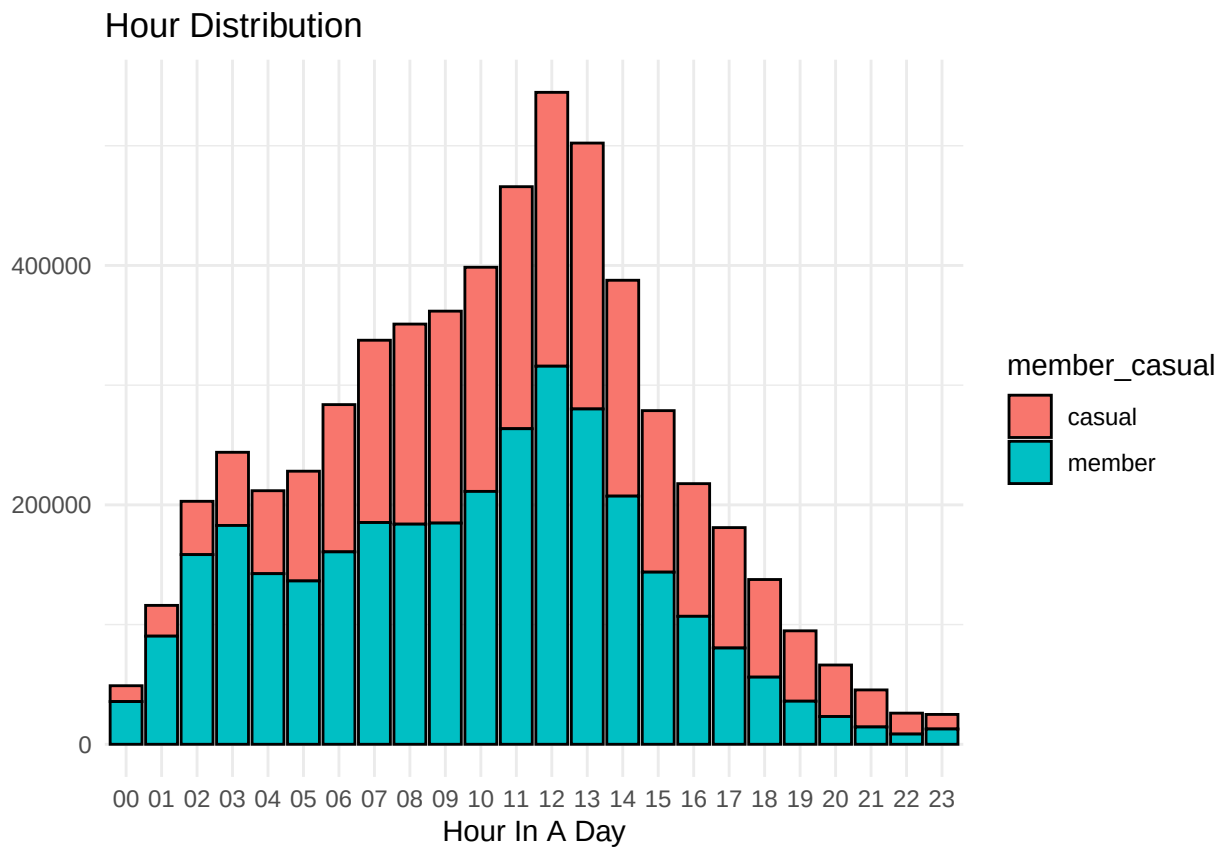
Start Hour Data Manipulation

```
 bike_rides_v2 <- bike_rides_v2 %>%
  mutate(start_hour = strftime(bike_rides_v2$ended_at, "%H"))
unique(bike_rides_v2$start_hour)
```

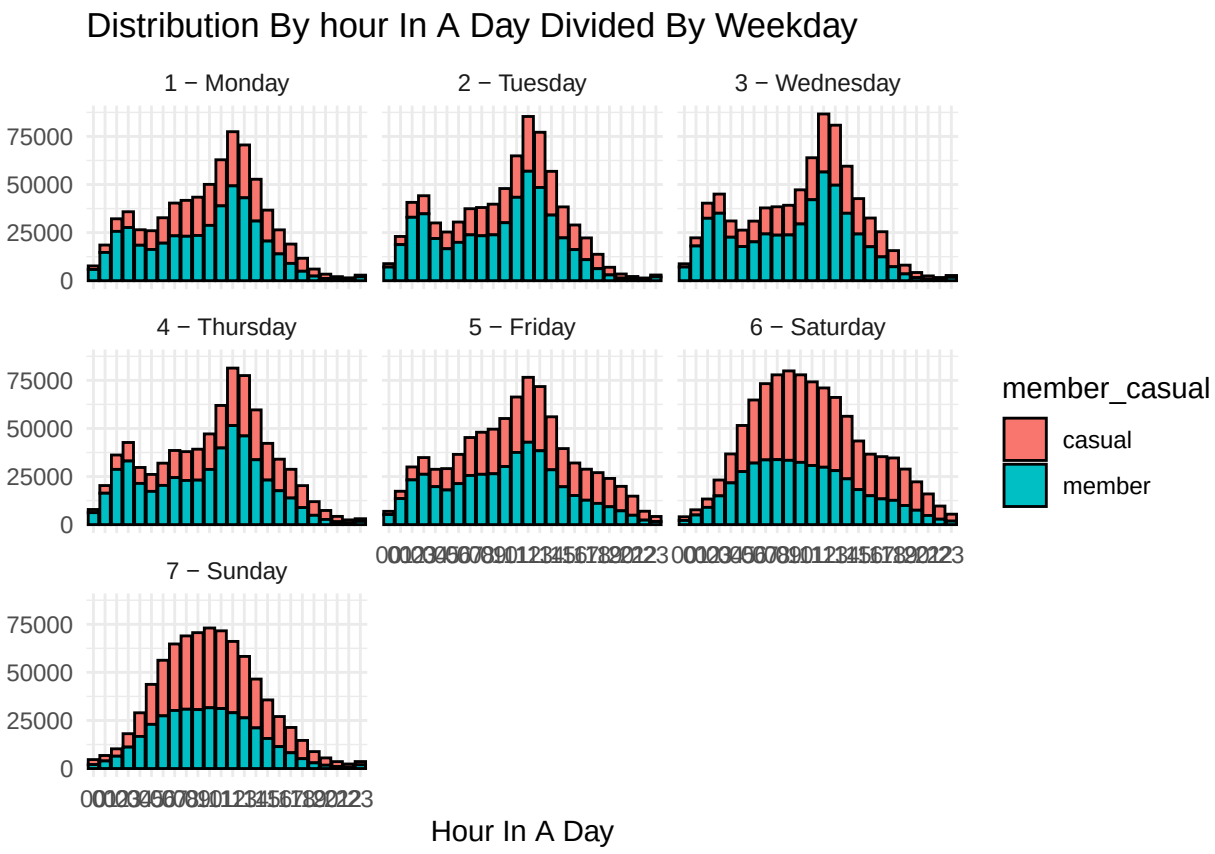
```
##  [1] "07" "09" "13" "06" "15" "11" "19" "17" "20" "10" "18" "12" "16" "04" "21"
## [16] "14" "08" "05" "03" "22" "02" "01" "23" "00"
```

Hours in a Day

```
bike_rides_v2 %>%
  ggplot(aes(start_hour, fill=member_casual)) +
  labs(x="Hour In A Day", y= NULL, title="Hour Distribution") +
  geom_bar(color = "black") +
  theme_minimal()
```




```
## Hour-Day / Weekday
bike_rides_v2 %>%
  ggplot(aes(start_hour, fill=member_casual)) +
  geom_bar(color = "black") +
  labs(x="Hour In A Day", y= NULL, title="Distribution By hour In A Day Divided By Weekday") +
  facet_wrap(~ weekday) +
   theme_minimal()
```



5. SHARE

The Monthly Distribution visualization shows that during winter there are significantly less casual riders than member riders. The Weekday Distribution visualization shows that casual riders use bikes less than members on weekdays. The Distribution By hour In A Day Divided By Weekday visualization confirms that casual riders use bikes less than members on weekdays espeicially during hours 00-06.

6. ACT

Marketing campaign: To convince casual riders to subscribe to a membership offer a discount for weekend rides. For example, casual riders save X% on rides during peak hours (07-15) on weekends by subscribing.