Google Data Analytics Capstone Project

Case Study 1: How Does a Bike-Share Navigate Speedy Success?

Scenario

You work as a junior data analyst in the marketing analyst team of **Cyclistic**, a Chicago-based bike-share firm. The company's future prosperity, according to the director of marketing, hinges on increasing the number of yearly subscribers. As a result, your team is interested in learning how casual riders and annual members use Cyclistic bikes. As a result of these findings,

Your group will devise a new marketing approach to convert casual riders into annual subscribers. However, Cyclistic executives must first be addressed. Your recommendations must be backed up by **compelling data insights and professional data visualizations** in order for them to be approved.

Business Tasks

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would a non-cyclist purchase a Cyclistic annual membership?
- Design marketing strategies aimed at converting casual riders into annual members.
- what impact digital media might have on their marketing strategies

Data Source

- Divvy Data

Setting up my environment

- Importing library `tidyverse` and loading datasets

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

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

Loading Data sets

```r
tripdata_202004 <- read_csv("202004-divvy-tripdata.csv")
```

```
## Rows: 84776 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.
```

```r
tripdata_202005 <- read_csv("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.
```

```r
tripdata_202006 <- read_csv("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.
```

```r
tripdata_202007 <- read_csv("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.
```

```
tripdata_202008 <- read_csv("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.
```

```
tripdata_202009 <- read_csv("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.
```

```
tripdata_202010 <- read_csv("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.
```

```r
tripdata_202011 <- read_csv("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.
```

```r
tripdata_202012 <- read_csv("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.
```

```r
tripdata_202101 <- read_csv("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.
```

```r
tripdata_202102 <- read_csv("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.
```

```
tripdata_202103 <- read_csv("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.
```

```
tripdata_202104 <- read_csv("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.
```

```
tripdata_202105 <- 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.
```

Combining all the data together

```
tripdata <- rbind(tripdata_202004,
                  tripdata_202005,
                  tripdata_202006,
                  tripdata_202007,
                  tripdata_202008,
                  tripdata_202009,
                  tripdata_202010,
                  tripdata_202011,
                  tripdata_202012,
                  tripdata_202101,
                  tripdata_202102,
                  tripdata_202103,
                  tripdata_202104,
                  tripdata_202105)
```

view the data

```
glimpse(tripdata)
```

```
## Rows: 4,358,611
## Columns: 13
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-04-26 17:45:14, 2020-04-17 17:08:54, 2020-04--~
## $ ended_at           <dttm> 2020-04-26 18:12:03, 2020-04-17 17:17:03, 2020-04--~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
## $ start_station_id   <chr> "86", "503", "142", "216", "125", "173", "35", "434~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <chr> "152", "499", "255", "657", "323", "35", "635", "38~
## $ start_lat          <dbl> 41.8964, 41.9244, 41.8945, 41.9030, 41.8902, 41.896~
## $ start_lng          <dbl> -87.6610, -87.7154, -87.6179, -87.6975, -87.6262, --~
## $ end_lat            <dbl> 41.9322, 41.9306, 41.8679, 41.8992, 41.9695, 41.892~
## $ end_lng            <dbl> -87.6586, -87.7238, -87.6230, -87.6722, -87.6547, --~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

Data Exploration

```
table(is.na(tripdata))
```

```
##
##    FALSE     TRUE
## 55786772   875171
```

Columns with missing values

```
colSums(is.na(tripdata))
```

```
##            ride_id      rideable_type         started_at           ended_at 
##                  0                  0                  0                  0 
## start_station_name   start_station_id   end_station_name     end_station_id 
##             201975             202601             229610             230071 
##          start_lat          start_lng            end_lat            end_lng 
##                  0                  0               5457               5457 
##      member_casual 
##                  0
```

Missing data would be removed.

```
tripdata_cleaned <- tripdata[complete.cases(tripdata), ]
```

checking the data again

```
colSums(is.na(tripdata_cleaned))
```

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

Data with a "started at" value greater than "ended at" will be removed.

```
tripdata_cleaned <- tripdata_cleaned %>%
  filter(tripdata_cleaned$started_at < tripdata_cleaned$ended_at)
```

```
glimpse(tripdata_cleaned)
```

```
## Rows: 4,033,125
## Columns: 13
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-04-26 17:45:14, 2020-04-17 17:08:54, 2020-04--~
## $ ended_at           <dttm> 2020-04-26 18:12:03, 2020-04-17 17:17:03, 2020-04--~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
## $ start_station_id   <chr> "86", "503", "142", "216", "125", "173", "35", "434~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <chr> "152", "499", "255", "657", "323", "35", "635", "38~
## $ start_lat          <dbl> 41.8964, 41.9244, 41.8945, 41.9030, 41.8902, 41.896~
## $ start_lng          <dbl> -87.6610, -87.7154, -87.6179, -87.6975, -87.6262, -~
## $ end_lat            <dbl> 41.9322, 41.9306, 41.8679, 41.8992, 41.9695, 41.892~
## $ end_lng            <dbl> -87.6586, -87.7238, -87.6230, -87.6722, -87.6547, -~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

- *Create a column called* ***`ride_length`*** *Calculate the length of each ride by subtracting the column* ***`started_at`*** *from the column* ***`ended_at`***

```r
tripdata_cleaned$ride_length <- tripdata_cleaned$ended_at - tripdata_cleaned$started_at

head(tripdata_cleaned$ride_length, 20)
```

```
## Time differences in secs
##  [1] 1609  489  863  732 3175  324  313 4549  344 1039 1452  293  498  274 1441
## [16] 2131 1302  493  955  354
```

Import lubridate library

```r
library(lubridate)
```

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

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

Since they are in numeric term we have to convert it.

```r
tripdata_cleaned$ride_length <- hms::hms(seconds_to_period(tripdata_cleaned$ride_length))
head(tripdata_cleaned$ride_length)
```

```
## 00:26:49
## 00:08:09
## 00:14:23
## 00:12:12
## 00:52:55
## 00:05:24
```

Create a column called "day_of_week," and calculate the day of the week that each ride started

- wday(x, label = FALSE, abbr = TRUE, ordered = FALSE)

```r
tripdata_cleaned <- tripdata_cleaned %>% mutate(day_of_week = wday(started_at, label = F))
```

```
head(tripdata_cleaned)
```

```
## # A tibble: 6 x 15
##   ride_id          rideable_type started_at          ended_at            start_station_n~
##   <chr>            <chr>         <dttm>              <dttm>              <chr>
## 1 A847FADBBC638E45 docked_bike   2020-04-26 17:45:14 2020-04-26 18:12:03 Eckhart Park
## 2 5405B80E996FF60D docked_bike   2020-04-17 17:08:54 2020-04-17 17:17:03 Drake Ave & Ful~
## 3 5DD24A79A4E006F4 docked_bike   2020-04-01 17:54:13 2020-04-01 18:08:36 McClurg Ct & Er~
## 4 2A59BBDF5CDBA725 docked_bike   2020-04-07 12:50:19 2020-04-07 13:02:31 California Ave ~
## 5 27AD306C119C6158 docked_bike   2020-04-18 10:22:59 2020-04-18 11:15:54 Rush St & Hubba~
## 6 356216E875132F61 docked_bike   2020-04-30 17:55:47 2020-04-30 18:01:11 Mies van der Ro~
## # ... with 10 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>, ride_length <time>, day_of_week <dbl>
```

Calculation

- Calculate the mean of ride_length
- Calculate the max ride_length
- Calculate the mode of day_of_week

```
library(DescTools)
```

```
tripdata_summary <- tripdata_cleaned %>% summarise(ride_length_mean = mean(ride_length), ride_length_ma

tripdata_summary
```

```
## # A tibble: 1 x 3
##   ride_length_mean ride_length_max mode_day_of_week
##   <drtn>           <drtn>                     <dbl>
## 1 1654.869 secs    3523202 secs                   7
```

- Calculate the average ride_length for members and casual riders, group by `member_casual`

```
average_ride_length <- tripdata_cleaned %>% group_by(member_casual) %>% summarise(average_ride_length =

average_ride_length
```

```
## # A tibble: 2 x 2
##   member_casual average_ride_length
##   <chr>         <drtn>
## 1 casual        2652.3596 secs
## 2 member         938.9665 secs
```

- Calculate the average ride_length for users by day_of_week. group by `day_of_week`

```
avg_ride_length <- tripdata_cleaned %>% group_by(day_of_week) %>% summarise(avg_ride_length = mean(ride

avg_ride_length
```

```
## # A tibble: 7 x 2
##   day_of_week avg_ride_length
##         <dbl> <drtn>
## 1           1 2064.147 secs
## 2           2 1538.681 secs
## 3           3 1402.635 secs
## 4           4 1397.767 secs
## 5           5 1450.453 secs
## 6           6 1579.200 secs
## 7           7 1926.302 secs
```

- Calculate the number of rides for users by day_of_week

```
count_rides <- tripdata_cleaned %>% group_by(day_of_week) %>% summarise(number_rides = n())

count_rides
```

```
## # A tibble: 7 x 2
##   day_of_week number_rides
##         <dbl>        <int>
## 1           1       635599
## 2           2       497245
## 3           3       494317
## 4           4       522219
## 5           5       523978
## 6           6       591740
## 7           7       768027
```

```
tripdata_cleaned %>% select(ride_length,member_casual,day_of_week) %>% group_by(day_of_week,member_casu
```

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

```
## # A tibble: 14 x 3
## # Groups:   day_of_week [7]
##    day_of_week member_casual mean
##          <dbl> <chr>         <drtn>
##  1           1 casual        3002.9704 secs
##  2           1 member        1070.9664 secs
##  3           2 casual        2628.8342 secs
##  4           2 member         892.2655 secs
##  5           3 casual        2402.9326 secs
##  6           3 member         882.3849 secs
##  7           4 casual        2391.8368 secs
##  8           4 member         885.4061 secs
##  9           5 casual        2489.3593 secs
## 10           5 member         882.7463 secs
## 11           6 casual        2546.9083 secs
## 12           6 member         913.4072 secs
## 13           7 casual        2736.9760 secs
## 14           7 member        1044.1187 secs
```

```
trip <- tripdata_cleaned %>% mutate(weekday = wday(started_at, label = T)) %>% group_by(member_casual,
```

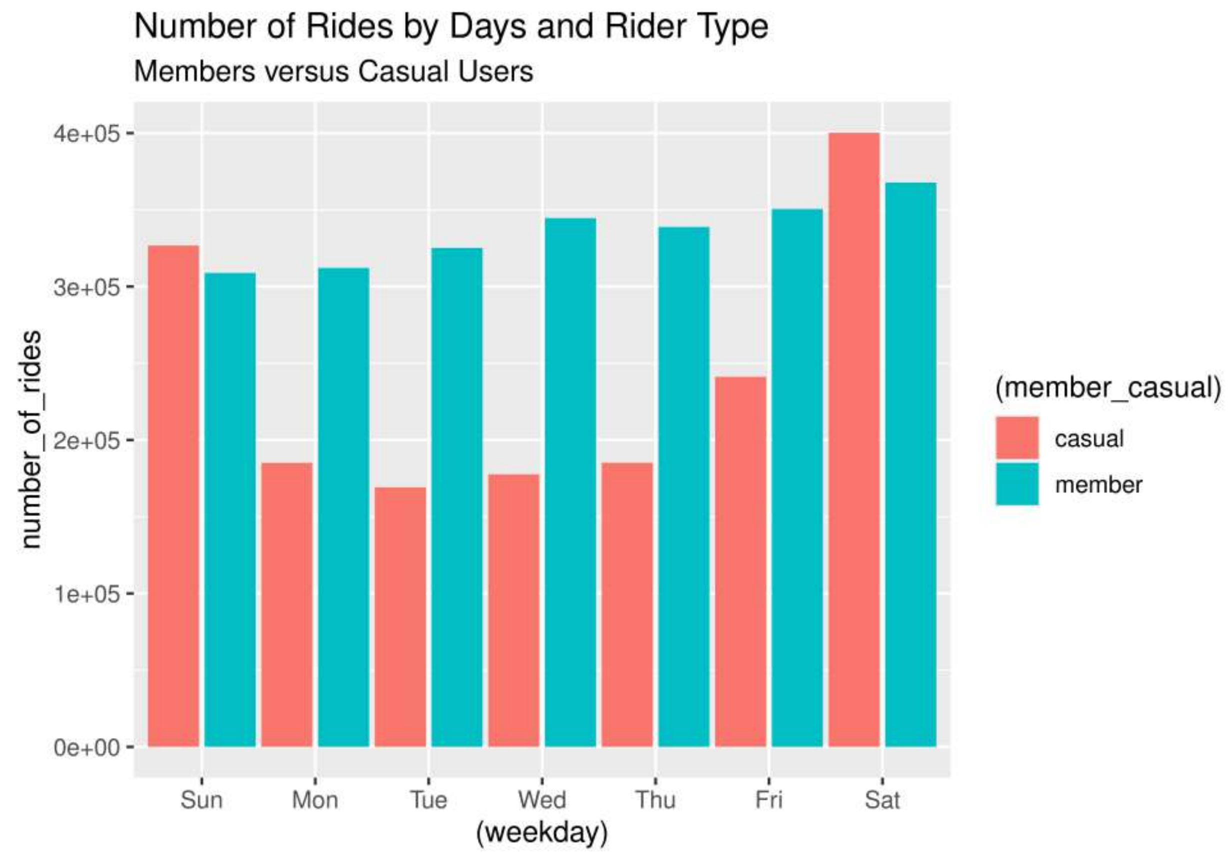
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
trip
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int> <drtn>
##  1 casual        Sun              326741 3002.9704 secs
##  2 casual        Mon              185093 2628.8342 secs
##  3 casual        Tue              169129 2402.9326 secs
##  4 casual        Wed              177615 2391.8368 secs
##  5 casual        Thu              185151 2489.3593 secs
##  6 casual        Fri              241185 2546.9083 secs
##  7 casual        Sat              400235 2736.9760 secs
##  8 member        Sun              308858 1070.9664 secs
##  9 member        Mon              312152  892.2655 secs
## 10 member        Tue              325188  882.3849 secs
## 11 member        Wed              344604  885.4061 secs
## 12 member        Thu              338827  882.7463 secs
## 13 member        Fri              350555  913.4072 secs
## 14 member        Sat              367792 1044.1187 secs
```

Visualizing number of rides by type

```
ggplot(data = trip) + aes( x = (weekday), y = number_of_rides, fill = (member_casual))+geom_bar(stat='i
       subtitle = "Members versus Casual Users")
```

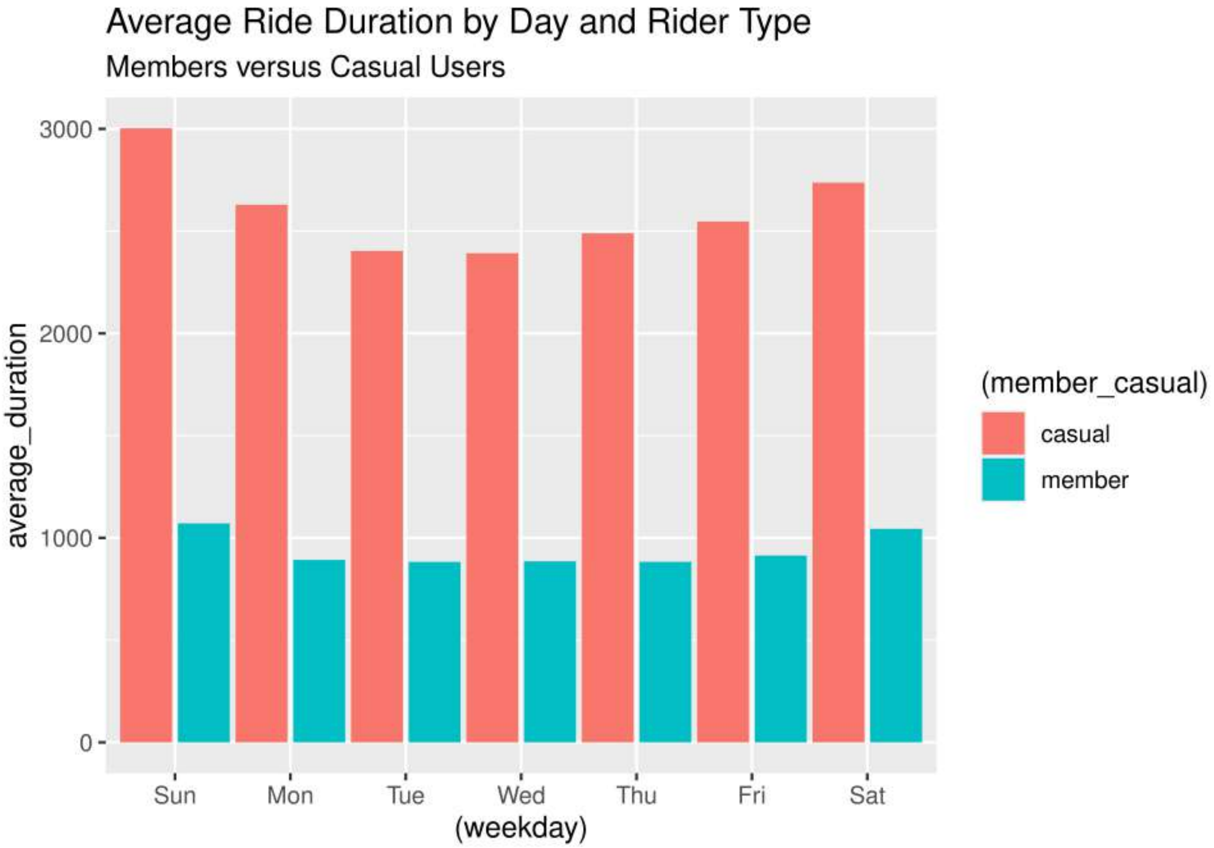


Saturday has the highest number of rides for both casual riders and members

Average Duration

```
ggplot(data = trip) + aes( x = (weekday), y = average_duration, fill = (member_casual))+geom_bar(stat='
       subtitle = "Members versus Casual Users")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



This shows that casual riders prefer longer rides

```
trip_add <- tripdata_cleaned %>% group_by(member_casual,rideable_type) %>% summarise(number_of_rides =
```

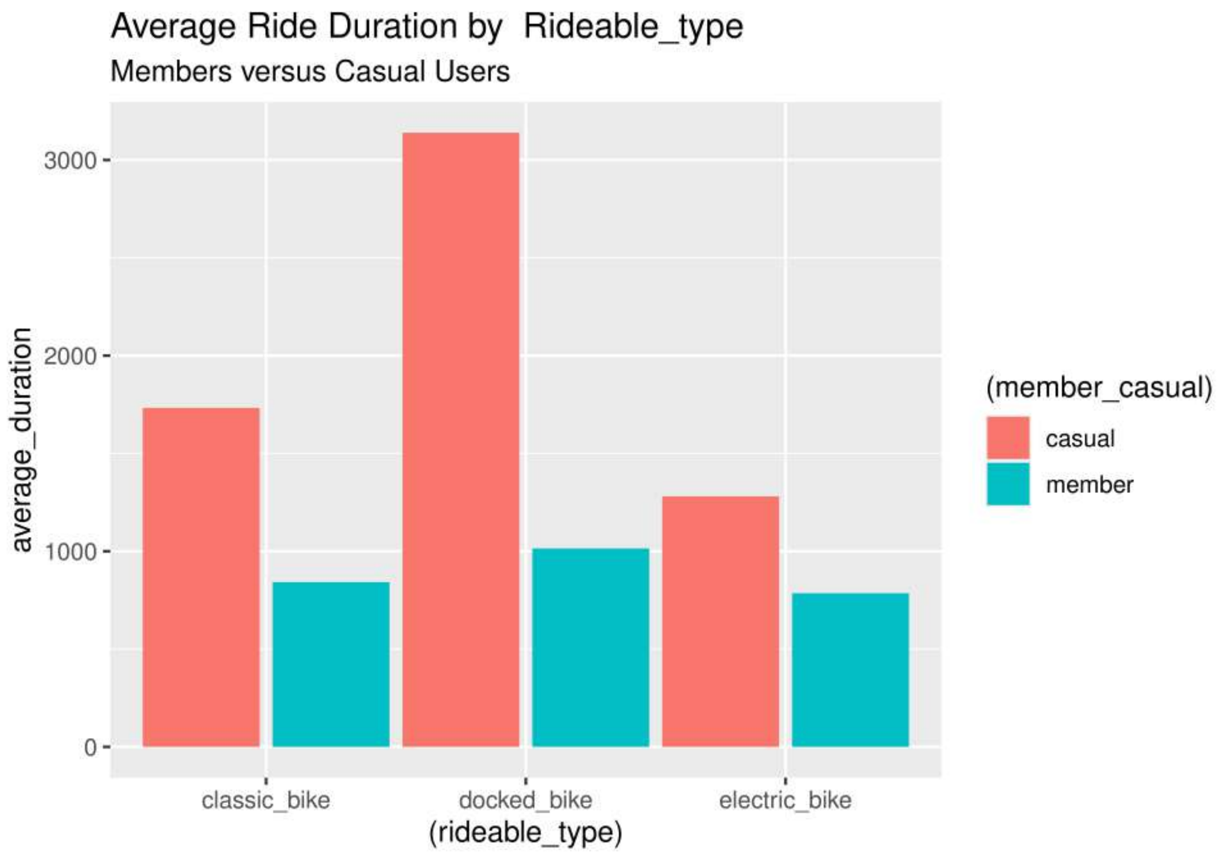
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
trip_add
```

```
## # A tibble: 6 x 4
## # Groups:   member_casual [2]
##   member_casual rideable_type number_of_rides average_duration
##   <chr>         <chr>                   <int> <drtn>
## 1 casual        classic_bike           264398 1732.6565 secs
## 2 casual        docked_bike           1179423 3139.2790 secs
## 3 casual        electric_bike          241328 1280.3002 secs
## 4 member        classic_bike           576650  842.1474 secs
## 5 member        docked_bike           1432357 1014.2990 secs
## 6 member        electric_bike          338969  785.3464 secs
```

```
ggplot(data = trip_add) + aes( x = (rideable_type), y = average_duration, fill = (member_casual))+geom_
       subtitle = "Members versus Casual Users")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



The most popular mode of transportation among riders is docked bicycles.

```
trip_month <- tripdata_cleaned %>% mutate(month = month(started_at, label = T)) %>% group_by(member_casu
```

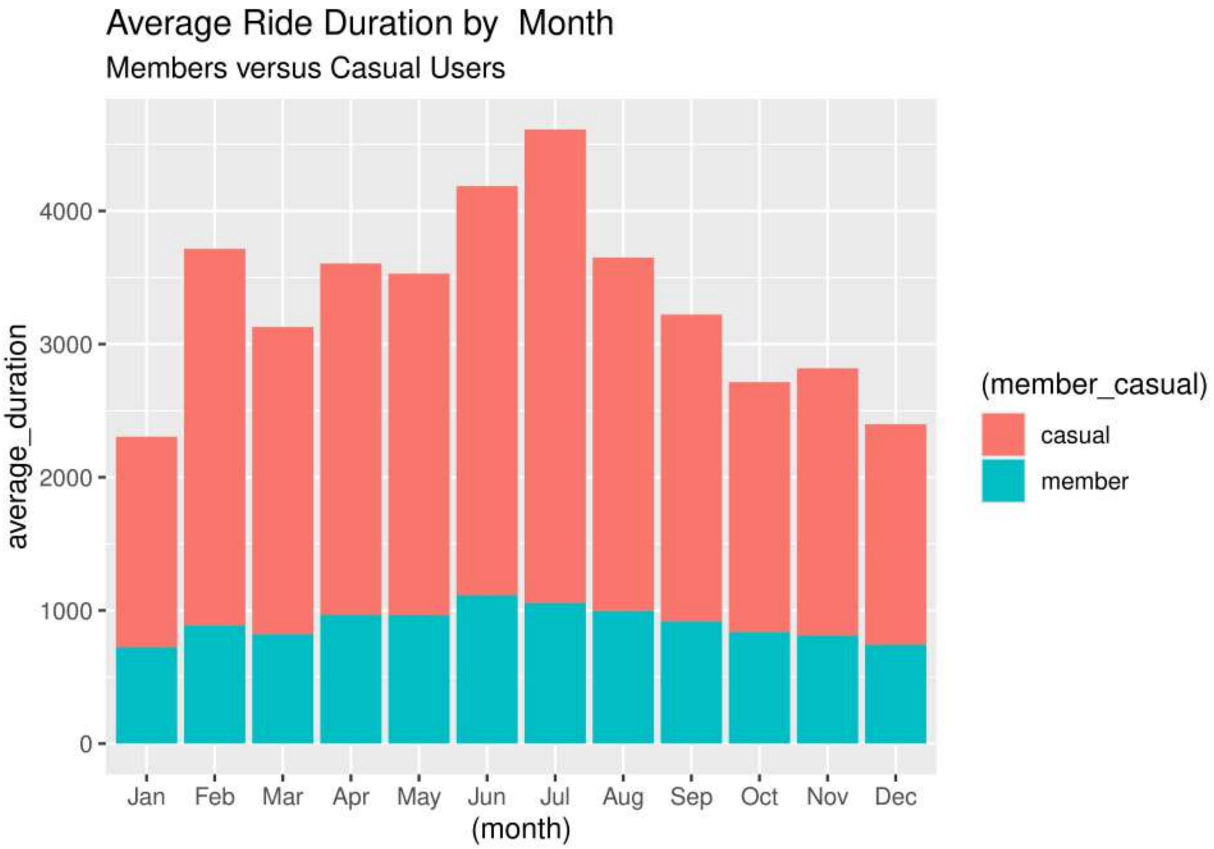
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
trip_month
```

```
## # A tibble: 24 x 4
## # Groups:   member_casual [2]
##    member_casual month number_of_rides average_duration
##    <chr>         <ord>           <int> <drtn>
##  1 casual        Jan             14690 1582.200 secs
##  2 casual        Feb              8613 2828.227 secs
##  3 casual        Mar             75641 2308.829 secs
##  4 casual        Apr            143984 2640.993 secs
##  5 casual        May            303516 2566.193 secs
##  6 casual        Jun            154329 3073.990 secs
##  7 casual        Jul            268103 3557.368 secs
##  8 casual        Aug            281945 2655.036 secs
##  9 casual        Sep            214672 2305.292 secs
## 10 casual        Oct            122317 1877.962 secs
## # ... with 14 more rows
```

```
ggplot(data = trip_month) + aes( x = (month), y = average_duration, fill = (member_casual))+geom_bar(st
       subtitle = "Members versus Casual Users")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



The longest riders are most active in July.

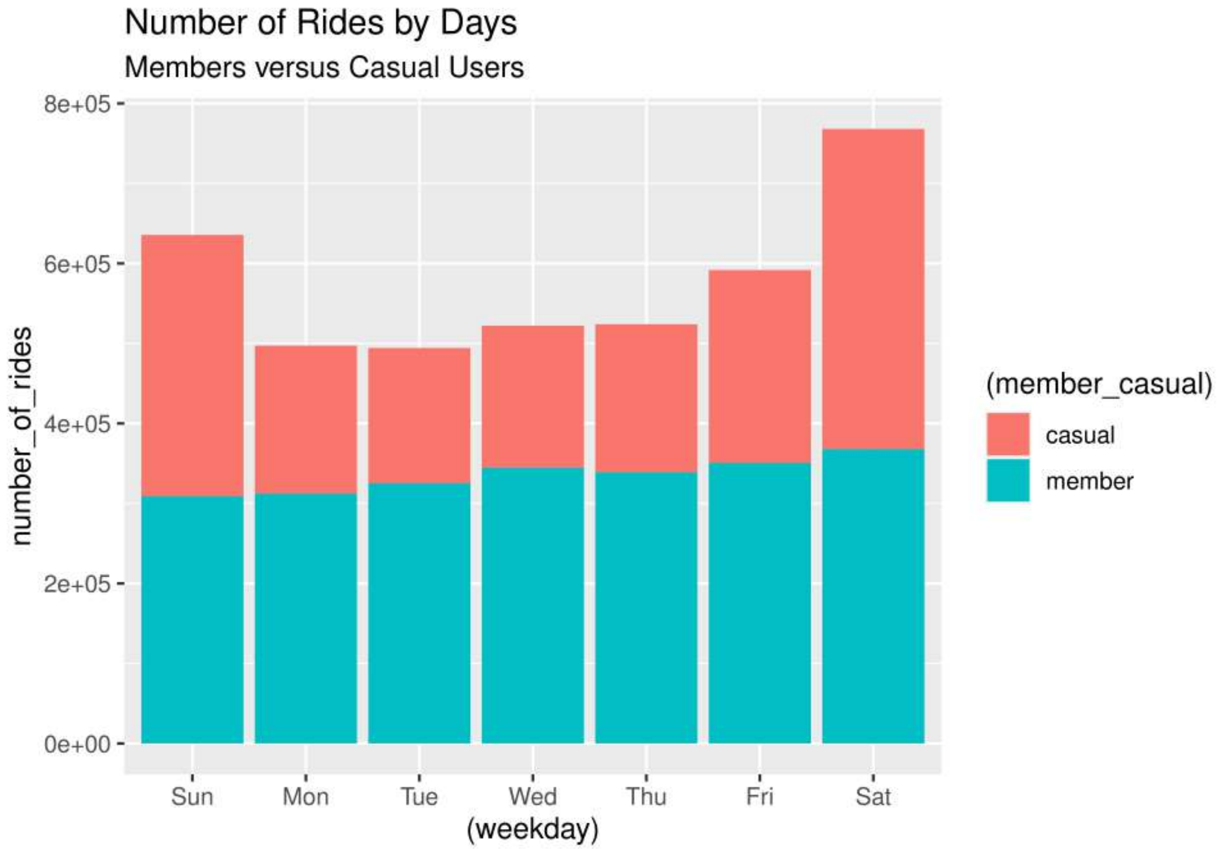
```
trip_ride <- tripdata_cleaned %>%  mutate(weekday = wday(started_at, label = T)) %>% group_by(member_ca
```

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

```
trip_ride
```

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides
##    <chr>         <ord>             <int>
##  1 casual        Sun              326741
##  2 casual        Mon              185093
##  3 casual        Tue              169129
##  4 casual        Wed              177615
##  5 casual        Thu              185151
##  6 casual        Fri              241185
##  7 casual        Sat              400235
##  8 member         Sun                308858
##  9 member         Mon                312152
## 10 member         Tue                325188
## 11 member         Wed                344604
## 12 member         Thu                338827
## 13 member         Fri                350555
## 14 member         Sat                367792
```

```
ggplot(data = trip_ride) + aes( x = (weekday), y = number_of_rides, fill = (member_casual))+geom_bar(st
       subtitle = "Members versus Casual Users")
```



I propose starting a marketing campaign in June to encourage casual riders to our Docked Bikes service on weekends, particularly Saturdays.

```
trip_station <- tripdata_cleaned %>%  group_by(member_casual,start_station_name ) %>% summarise(number_
```

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

```
trip_station
```

```
## # A tibble: 1,403 x 4
## # Groups:   member_casual [2]
##    member_casual start_station_name        number_of_rides average_duration
##    <chr>         <chr>                               <int> <drtn>
##  1 casual        Streeter Dr & Grand Ave             36471 3037.2404 secs
##  2 casual        Lake Shore Dr & Monroe St           28000 3167.0789 secs
##  3 casual        Millennium Park                     25240 4368.8014 secs
##  4 member        Clark St & Elm St                   23885  823.8508 secs
##  5 member        Wells St & Concord Ln               18484  799.1689 secs
##  6 casual        Theater on the Lake                 18362 2402.8746 secs
##  7 casual        Michigan Ave & Oak St               18351 3180.9244 secs
##  8 member        Broadway & Barry Ave                18319  902.7046 secs
##  9 member        Dearborn St & Erie St               18152  809.5024 secs
## 10 member        St. Clair St & Erie St              18116  941.9241 secs
## # ... with 1,393 more rows
```

We should also pay attention to these stations

- Streeter Sr & Grand Ave
- Lake Shore Dr & Monroe St

- Millenium Park.