CaseStudy:How do annual members and casual riders use Cyclistic bikes differently?

Ashutosh

2022-11-03

#This analysis is done on data provided by Motivate International Inc. #under the license https://www.divvybikes.com/data-license-agreement , its a public data , there is PII in these datasets. Below script cosolidates cyclistic data into a single data frame followed by analysis to answer questions. #Question : How do annual members and casual riders use Cyclistic bikes differently?

#package installation #tidyverse : data import and wrangling #lubridate : data functions #ggplot : Visualization #

```
install.packages("tinytex",repos="https://github.com/rstudio/tinytex")
```

```
## Installing package into 'C:/Users/ASHUTOSH KM/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository https://github.com/rstudio/tinytex/src/contrib:
##   cannot open URL 'https://github.com/rstudio/tinytex/src/contrib/PACKAGES'
```

```
## Warning: package 'tinytex' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository https://github.com/rstudio/tinytex/bin/windows/contri
##   cannot open URL 'https://github.com/rstudio/tinytex/bin/windows/contrib/4.2/PACKAGES'
```

```
tinytex::install_tinytex(force = TRUE)
library(tidyverse)  #helps wrangle data
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

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

```
library(lubridate)  #helps wrangle date attributes
```

```
## Warning: package 'lubridate' was built under R version 4.2.2
```

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

```
library(ggplot2)  #helps visualize data
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.2.2
```

```
getwd() #to find current working directory
```

```
## [1] "C:/Users/ASHUTOSH KM/Documents"
```

```
setwd("/Users/ASHUTOSH KM/Documents")
#for setting current working directory where data sets are placed
```

```
#-------1- "Data Collection" : Import data (.csv file) in MSSQL Server --------

# Upload Divvy datasets (csv files) using below code :

q2_2019 <- read_csv("Divvy_Trips_2019_Q2.csv")
```

```
## Rows: 1108163 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...
## dbl  (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...
## num  (1): 01 - Rental Details Duration In Seconds Uncapped
## dttm (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local 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.
```

```
q3_2019 <- read_csv("Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
q4_2019 <- read_csv("Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
q1_2020 <- read_csv("Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426887 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.
```

```
#-----------------------------------------------------------------------------
```

```
#------2-"Combine Data in single table"-------------------------------------
#a) - First rename columns so that its consistant in all tables.
#Keep column names of Q1_2022 in all tables.

(q4_2019 <- rename(q4_2019
                   ,ride_id = trip_id
                   ,rideable_type = bikeid
                   ,started_at = start_time
                   ,ended_at = end_time
                   ,start_station_name = from_station_name
                   ,start_station_id = from_station_id
                   ,end_station_name = to_station_name
                   ,end_station_id = to_station_id
                   ,member_casual = usertype))
```

```
## # A tibble: 704,054 x 12
##    ride_id started_at          ended_at            rideable_t~1 tripd~2 start~3
```

```
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20         2215     940      20
##  2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34         6328     258      19
##  3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43         3003     850      84
##  4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43         3275    2350     313
##  5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42         5294    1867     210
##  6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51         1891     373     156
##  7 25223646 2019-10-01 00:04:52 2019-10-01 00:22:45         1061    1072      84
##  8 25223647 2019-10-01 00:04:57 2019-10-01 00:29:16         1274    1458     156
##  9 25223648 2019-10-01 00:05:20 2019-10-01 00:29:18         6011    1437     156
## 10 25223649 2019-10-01 00:05:20 2019-10-01 02:23:46         2957    8306     336
## # ... with 704,044 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, and abbreviated variable names
## #   1: rideable_type, 2: tripduration, 3: start_station_id
```

```
(q3_2019 <- rename(q3_2019
                  ,ride_id = trip_id
                  ,rideable_type = bikeid
                  ,started_at = start_time
                  ,ended_at = end_time
                  ,start_station_name = from_station_name
                  ,start_station_id = from_station_id
                  ,end_station_name = to_station_name
                  ,end_station_id = to_station_id
                  ,member_casual = usertype))
```

```
## # A tibble: 1,640,718 x 12
##     ride_id started_at          ended_at            rideable_t~1 tripd~2 start~3
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41         3591    1214     117
##  2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44         5353    1048     381
##  3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42         6180    1554     313
##  4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10         5540    1503     313
##  5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26         6014    1213     168
##  6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31         4941     310     300
##  7 23479394 2019-07-01 00:02:24 2019-07-01 00:23:12         3770    1248     168
##  8 23479395 2019-07-01 00:02:26 2019-07-01 00:28:16         5442    1550     313
##  9 23479396 2019-07-01 00:02:34 2019-07-01 00:28:57         2957    1583      43
## 10 23479397 2019-07-01 00:02:45 2019-07-01 00:29:14         6091    1589      43
## # ... with 1,640,708 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, and abbreviated variable names
## #   1: rideable_type, 2: tripduration, 3: start_station_id
```

```
(q2_2019 <- rename(q2_2019
                  ,ride_id = "01 - Rental Details Rental ID"
                  ,rideable_type = "01 - Rental Details Bike ID"
                  ,started_at = "01 - Rental Details Local Start Time"
                  ,ended_at = "01 - Rental Details Local End Time"
                  ,start_station_name = "03 - Rental Start Station Name"
                  ,start_station_id = "03 - Rental Start Station ID"
                  ,end_station_name = "02 - Rental End Station Name"
```

```
                   ,end_station_id = "02 - Rental End Station ID"
                   ,member_casual = "User Type"))
```

```
## # A tibble: 1,108,163 x 12
##     ride_id started_at          ended_at            rideable_t~1 01 - ~2 start~3
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48         6251     446      81
##  2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30         6226    1048     317
##  3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19         5649     252     283
##  4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58         4151     357      26
##  5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13         3270    1007     202
##  6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56         3123     257     420
##  7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41         6418     548     503
##  8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11         4513     383     260
##  9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44         3280    2137     211
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39         5534    2120     211
## # ... with 1,108,153 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   `Member Gender` <chr>, `05 - Member Details Member Birthday Year` <dbl>,
## #   and abbreviated variable names 1: rideable_type,
## #   2: `01 - Rental Details Duration In Seconds Uncapped`, 3: start_station_id
```

```
#b). Convert ride_id and rideable_type to character data type

q4_2019 <-  mutate(q4_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))
q3_2019 <-  mutate(q3_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))
q2_2019 <-  mutate(q2_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))

#c). combine all data frames into one big data frame.
all_trips <- bind_rows(q2_2019, q3_2019, q4_2019, q1_2020)

#d). remove columns not required in all tables
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, "01 - Rental Details Duration In

#------------------------------------------------------------------------
```

```
#------------- 3- Data Cleaning and add new columns ----------------------

# created data frame needs to be inspected
colnames(all_trips)  #List of column names
```

```
## [1] "ride_id"            "started_at"         "ended_at"
## [4] "rideable_type"      "start_station_id"   "start_station_name"
## [7] "end_station_id"     "end_station_name"   "member_casual"
```

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

```
## [1] 3879822
```

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

```
## [1] 3879822       9
```

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

```
## # A tibble: 6 x 9
##   ride_id  started_at          ended_at            rideable_type start~1 start~2
##   <chr>    <dttm>              <dttm>              <chr>           <dbl> <chr>
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251               81 Daley ~
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226              317 Wood S~
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649              283 LaSall~
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151               26 McClur~
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270              202 Halste~
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123              420 Ellis ~
## # ... with 3 more variables: end_station_id <dbl>, end_station_name <chr>,
## #   member_casual <chr>, and abbreviated variable names 1: start_station_id,
## #   2: start_station_name
```

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

```
## tibble [3,879,822 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...
##  $ started_at        : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...
##  $ ended_at          : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...
##  $ rideable_type     : chr [1:3879822] "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jacks
##  $ end_station_id    : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal
##  $ member_casual     : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
```

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

```
##    ride_id            started_at
##  Length:3879822     Min.   :2019-04-01 00:02:22.00
##  Class :character   1st Qu.:2019-06-23 07:49:09.25
##  Mode  :character   Median :2019-08-14 17:43:38.00
##                     Mean   :2019-08-26 00:49:59.38
##                     3rd Qu.:2019-10-12 12:10:21.00
##                     Max.   :2020-03-31 23:51:34.00
##
##     ended_at                      rideable_type      start_station_id
##  Min.   :2019-04-01 00:09:48.00   Length:3879822     Min.   :  1.0
##  1st Qu.:2019-06-23 08:20:27.75   Class :character   1st Qu.: 77.0
##  Median :2019-08-14 18:02:04.00   Mode  :character   Median :174.0
##  Mean   :2019-08-26 01:14:37.06                      Mean   :202.9
##  3rd Qu.:2019-10-12 12:36:16.75                      3rd Qu.:291.0
##  Max.   :2020-05-19 20:10:34.00                      Max.   :675.0
##
##  start_station_name end_station_id  end_station_name   member_casual
```

```
##  Length:3879822     Min.   :  1.0   Length:3879822     Length:3879822
##  Class :character   1st Qu.: 77.0   Class :character   Class :character
##  Mode  :character   Median :174.0   Mode  :character   Mode  :character
##                     Mean   :203.8
##                     3rd Qu.:291.0
##                     Max.   :675.0
##                     NA's   :1
```

```
#Data needs to be fixed
# 1. member_casual column has4 types of values "member","Subscriber","Customer","casual".
                        #here only 2 values are required that is "Member" and "casuals". need to change
                        #"Subcriber" to "member" and "customer" to "casual"

#2. some additional columns are required day, month, year
#3. 2020Q1 is not having trip duration column. so adding ride_length to the entire dataframe.
#4. for some rides trip duration is negative . those records will be deleted .


# Reassigning desired values (current 2020 labels are used)
all_trips <-  all_trips %>%
  mutate(member_casual = recode(member_casual
                            ,"Subscriber" = "member"
                            ,"Customer" = "casual"))
```

```
# Add columns that list the date, month, day, and year of each ride
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")

# Addition of ride length column
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)

# deleting negative values
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]

# ---------------------------------------------------------------------
```

```
# 4- DESCRIPTIVE ANALYSIS

# mean of ride length
mean(all_trips_v2$ride_length)
```

```
## Time difference of 1479.139 secs
```

```
# max of ride length
max(all_trips_v2$ride_length)
```

```
## Time difference of 9387024 secs
```

```r
# min of ride length
min(all_trips_v2$ride_length)
```

```
## Time difference of 1 secs
```

```r
#median
median(all_trips_v2$ride_length)
```

```
## Time difference of 712 secs
```

```r
# check summary
summary(all_trips_v2$ride_length)
```

```
##   Length    Class     Mode
##  3876042 difftime  numeric
```

```r
# lets compare members and casual riders data
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual           3552.7502 secs
## 2                     member            850.0662 secs
```

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1546 secs
## 2                     member                 589 secs
```

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual             9387024 secs
## 2                     member             9056634 secs
```

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                   2 secs
## 2                     member                   1 secs
```

```r
#check average ride time by each day for members vs casual users
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                  Friday           3773.8351 secs
## 2                      member                  Friday            824.5305 secs
## 3                      casual                  Monday           3372.2869 secs
```

```
## 4                      member                Monday           842.5726 secs
## 5                      casual              Saturday          3331.9138 secs
## 6                      member              Saturday           968.9337 secs
## 7                      casual                Sunday          3581.4054 secs
## 8                      member                Sunday           919.9746 secs
## 9                      casual              Thursday          3682.9847 secs
## 10                     member              Thursday           823.9278 secs
## 11                     casual               Tuesday          3596.3599 secs
## 12                     member               Tuesday           826.1427 secs
## 13                     casual             Wednesday          3718.6619 secs
## 14                     member             Wednesday           823.9996 secs
```

```
# order by days of week
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "
```

```
# check again average ride time by each day for members vs casual users
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday          3581.4054 secs
## 2                      member                   Sunday           919.9746 secs
## 3                      casual                   Monday          3372.2869 secs
## 4                      member                   Monday           842.5726 secs
## 5                      casual                  Tuesday          3596.3599 secs
## 6                      member                  Tuesday           826.1427 secs
## 7                      casual                Wednesday          3718.6619 secs
## 8                      member                Wednesday           823.9996 secs
## 9                      casual                 Thursday          3682.9847 secs
## 10                     member                 Thursday           823.9278 secs
## 11                     casual                   Friday          3773.8351 secs
## 12                     member                   Friday           824.5305 secs
## 13                     casual                 Saturday          3331.9138 secs
## 14                     member                 Saturday           968.9337 secs
```

```
# analyze ridership data by type and weekday
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()#calculates the number of rides and average duration
  ,average_duration = mean(ride_length)) %>% # calculates the average duration
  arrange(member_casual, weekday)# sorts
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int> <drtn>
##  1 casual        Sun              181293 3581.4054 secs
##  2 casual        Mon              103296 3372.2869 secs
##  3 casual        Tue               90510 3596.3599 secs
```

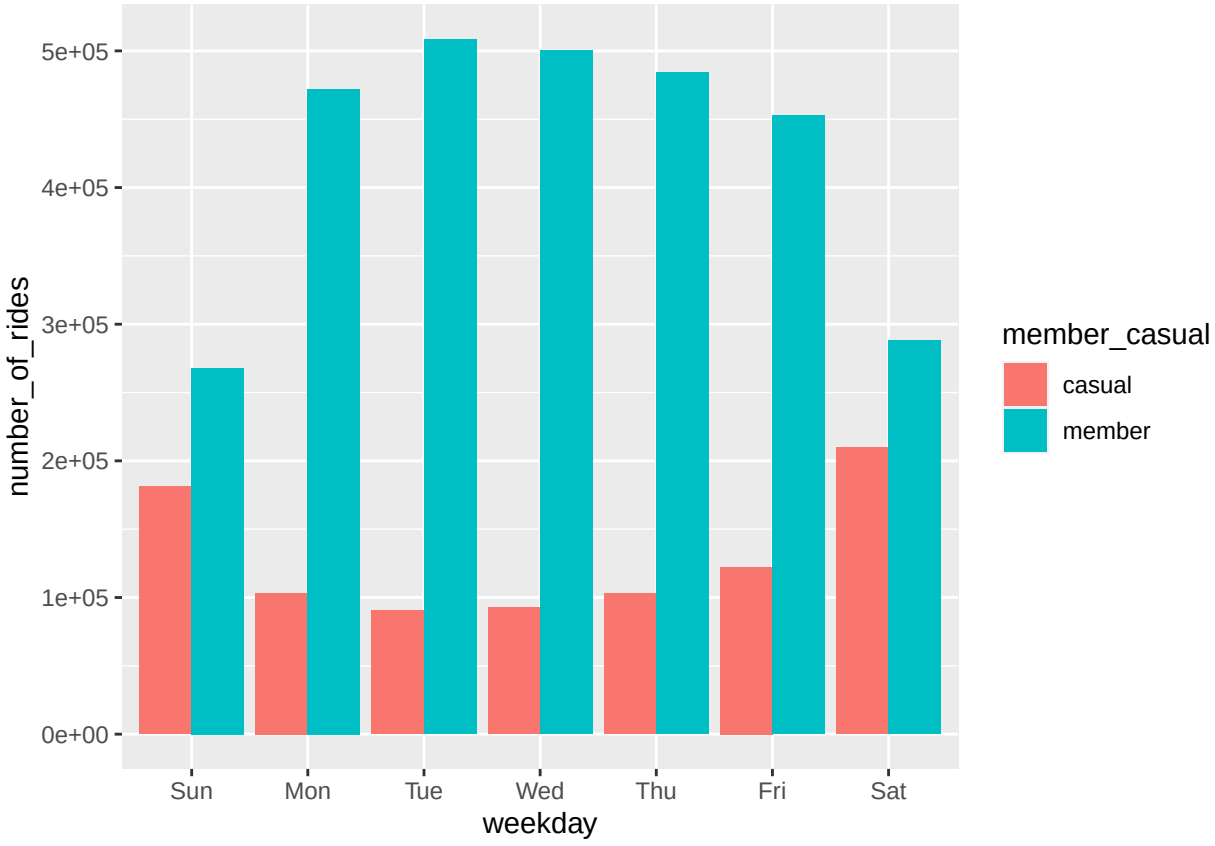
```
##  4 casual        Wed               92457 3718.6619 secs
##  5 casual        Thu              102679 3682.9847 secs
##  6 casual        Fri              122404 3773.8351 secs
##  7 casual        Sat              209543 3331.9138 secs
##  8 member        Sun              267965  919.9746 secs
##  9 member        Mon              472196  842.5726 secs
## 10 member        Tue              508445  826.1427 secs
## 11 member        Wed              500329  823.9996 secs
## 12 member        Thu              484177  823.9278 secs
## 13 member        Fri              452790  824.5305 secs
## 14 member        Sat              287958  968.9337 secs
```

```r
# Visualization

# number of rides by rider type

all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
  ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

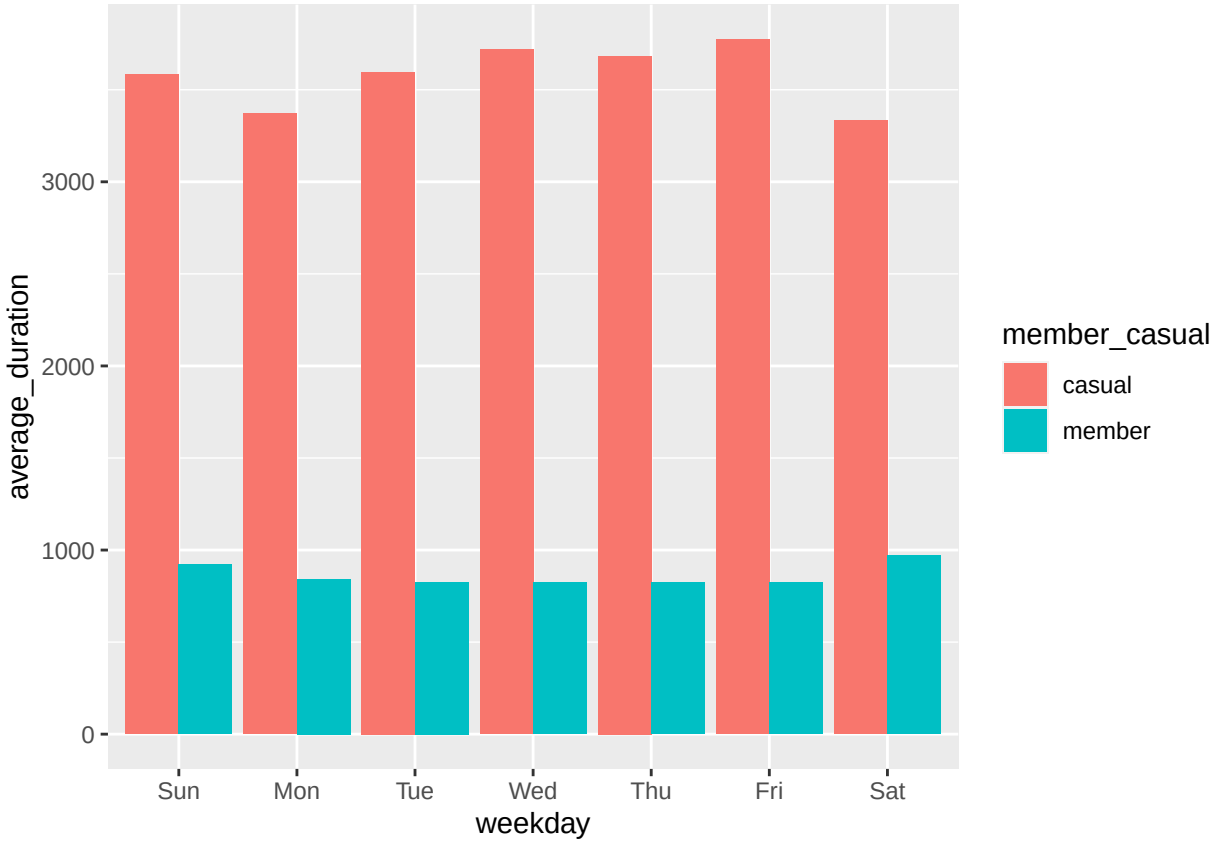
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```




```
# by average duration

all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type difftime.
## Defaulting to continuous.
```




```
#-------------------------------------------------------------------------

#5- Summary of Analysis
#a). Complete analysis includes running various queries to perform calculations to calculate mean,
                      #median, max and min of rides .
#b). The trend i can see is there are less casual riders but they drive for long but not frequently
                      #whereas member riders are more in number and they ride frequently
                      #but for less duration.
#c). final table is prepared for sharing visualization

#-------------------------------------------------------------------------

#6- Share vizualization and Key Findings:
#a).complete visualization is done in R

#b). Key findings :  Annual Members who are 3 times in number compared to Casual riders use
                      #frequently for short duration whereas Casual use for long duration
                      #but not frequently.

#-------------------------------------------------------------------------


#7- ACT :
#Top three recommendations based on analysis

#Convert Casual riders to Annual rider : Come up with a membership plan to attract casual members,
```

```
                        #which is profitable for Cyclistic as well.

#Target new customers : Members are those who ride less duration , less distance but frequently or on
                        #daily basis . so, we can connect win new audience who ride less duration but
                        #frequently like office goers and try to make them members.

# We can target casual members with a campaign and attractive membership plans in the month of July ,
                        #Aug, Sept that is 3rd Quarter
```