Case Study Report

Moise Nvekounou

2024-07-24

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(conflicted)
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
conflict_prefer("lag", "dplyr")
```

```
## [conflicted] Will prefer dplyr::lag over any other package.
```

Introduction

Cyclistic is a bike-share company in Chicago that offers to its customers a program of pricing plan based on two category riders: casual riders and annual members. The stakeholder' goal is to increase the number of the annual members that will allow the company to succeed. To reach this business objective, it is important to understand how different customer types are using Cyclistic bikes in order to know what marketing strategy to use to convert the first category into second one. This is the problem I will address in this report by following the six data analysis phases: ask, prepare, process, analyze, share and act.

1- Ask phase: Statement of business task

- How different customer category use Cyclistic bikes ?
- How to encourage casual riders to become Cyclistic annual members ?

2- Prepare phase: Description of all data sources used

Location of data source

All data used for this analysis is located: here : https://divvy-tripdata.s3.amazonaws.com/index.html

All datasets uploaded but only those listed above are used for the analysis:

divvy_tripdata_202406.csv (renamed) divvy_tripdata_202405.csv (renamed) Divvy_Trips_2019_Q1.csv Divvy_Trips_2020_Q1.csv

Uploading csv files and compare column names for different files

```
divvy_tripdata_202406 <- read_csv("divvy_tripdata_202406.csv")
```

```
## Rows: 710721 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.
```

```
colnames(divvy_tripdata_202406)
```

```
##  [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"
```

```
divvy_tripdata_202405 <- read_csv("divvy_tripdata_202405.csv")
```

```
## Rows: 609493 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.
```

```
colnames(divvy_tripdata_202405)
```

```
##  [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"
```

The structure of each of both files are the same.

Checking the data errors into divvy_tripdata_202406

```
rows_with_na <- divvy_tripdata_202406[!complete.cases(divvy_tripdata_202406),]
print(rows_with_na)
```

```
## # A tibble: 216,395 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 CDE6023BE6B11D2F electric_bike 2024-06-11 17:20:06 2024-06-11 17:21:39
##  2 462B48CD292B6A18 electric_bike 2024-06-11 17:19:21 2024-06-11 17:19:36
##  3 9CFB6A858D23ABF7 electric_bike 2024-06-11 17:25:27 2024-06-11 17:30:13
##  4 6365EFEB64231153 electric_bike 2024-06-11 11:53:50 2024-06-11 12:08:13
##  5 BA0323C33134CBA8 electric_bike 2024-06-11 00:11:08 2024-06-11 00:11:22
##  6 DE26F0D728517B77 electric_bike 2024-06-11 00:12:38 2024-06-11 00:12:57
##  7 C806B4280D388CC0 electric_bike 2024-06-11 00:14:00 2024-06-11 00:20:35
##  8 45CBD564C461DBE9 electric_bike 2024-06-11 18:22:23 2024-06-11 19:26:41
##  9 D943D9C4841D9A60 electric_bike 2024-06-11 12:51:10 2024-06-11 12:51:28
## 10 2C9D2B7821354BFF electric_bike 2024-06-11 12:49:24 2024-06-11 12:49:45
## # i 216,385 more rows
## # i 9 more variables: start_station_name <chr>, 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>
```

Using drop_na function to remove all NA from divvy_tripdata_202406

```
clean_dt_202406 <- divvy_tripdata_202406 %>% drop_na()
str(clean_dt_202406)
```

```
## tibble [494,326 x 13] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:494326] "7FED56E160AFB564" "84260B28A7C9BBA1" "95367640BB007C8D" "4DF08
##  $ rideable_type     : chr [1:494326] "classic_bike" "classic_bike" "classic_bike" "electric_bike" .
##  $ started_at        : POSIXct[1:494326], format: "2024-06-17 15:10:56" "2024-06-17 15:10:35" ...
##  $ ended_at          : POSIXct[1:494326], format: "2024-06-17 15:12:30" "2024-06-17 15:12:12" ...
##  $ start_station_name: chr [1:494326] "California Ave & Division St" "California Ave & Division St"
##  $ start_station_id  : chr [1:494326] "13256" "13256" "13256" "13256" ...
##  $ end_station_name  : chr [1:494326] "California Ave & Division St" "California Ave & Division St"
##  $ end_station_id    : chr [1:494326] "13256" "13256" "13084" "637" ...
##  $ start_lat         : num [1:494326] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:494326] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:494326] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:494326] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:494326] "casual" "casual" "casual" "casual" ...
```

Creating columns trip_duration and day_of_week

```
mutate_dt_202406 <- clean_dt_202406 %>%
  select(ride_id, rideable_type, started_at, ended_at, start_station_name, member_casual) %>%
  mutate(trip_duration = round(difftime(ended_at, started_at), 0), day_of_week = weekdays(started_at))
print(mutate_dt_202406)
```

```
## # A tibble: 494,326 x 8
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 7FED56E160AFB564 classic_bike  2024-06-17 15:10:56 2024-06-17 15:12:30
##  2 84260B28A7C9BBA1 classic_bike  2024-06-17 15:10:35 2024-06-17 15:12:12
##  3 95367640BB007C8D classic_bike  2024-06-08 16:11:10 2024-06-08 16:21:25
##  4 4DF083CCDC1B950F electric_bike 2024-06-07 21:33:36 2024-06-07 21:45:23
##  5 BFAD51AB1A4887B2 classic_bike  2024-06-24 17:51:13 2024-06-24 17:56:09
##  6 4495EFAA7B0F28DF electric_bike 2024-06-29 08:42:23 2024-06-29 08:59:28
##  7 D050A9E811C95628 classic_bike  2024-06-22 11:34:39 2024-06-22 12:04:17
##  8 A8F5AAC42C028AE3 classic_bike  2024-06-22 01:53:07 2024-06-22 01:55:57
##  9 2FFBDD1386C14044 electric_bike 2024-06-19 19:14:04 2024-06-19 19:34:41
## 10 EE4AD66ED3084103 classic_bike  2024-06-24 18:33:16 2024-06-24 18:47:53
## # i 494,316 more rows
## # i 4 more variables: start_station_name <chr>, member_casual <chr>,
## #   trip_duration <drtn>, day_of_week <chr>
```

Now looking at the other files Divvy_Trips_2019_Q1.csv and Divvy_Trips_2020_Q1.csv

Uploading csv files and compare column names for files Divvy_Trips_2019_Q1.csv and Divvy_Trips_2020_Q1.csv

```
q1_2019 <- read_csv("Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 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.
```

```
colnames(q1_2019)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
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.
```

```
colnames(q1_2020)
```

```
##  [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"
```

The column names in this case are a little different. We need to make them consistent before working with data

Renaming columns for q1_2019 to make it consistent with q1_2020

```
q1_2019 <- rename(q1_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)
```

Checking data frame

```
str(q1_2019) #Listing q1_2019 columns and data types
```

```
## spc_tbl_ [365,069 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id          : num [1:365069] 21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at       : POSIXct[1:365069], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at         : POSIXct[1:365069], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type    : num [1:365069] 2167 4386 1524 252 1170 ...
##  $ tripduration     : num [1:365069] 390 441 829 1783 364 ...
##  $ start_station_id : num [1:365069] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:365069] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
```

```
##  $ end_station_id    : num [1:365069] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:365069] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "
##  $ member_casual     : chr [1:365069] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender            : chr [1:365069] "Male" "Female" "Female" "Male" ...
##  $ birthyear         : num [1:365069] 1989 1990 1994 1993 1994 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q1_2020) #Listing q1_2020 columns and data types
```

```
## spc_tbl_ [426,887 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A3
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilt
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member" ...
##  - 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_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
```

```
##  - attr(*, "problems")=<externalptr>
```

Converting ride_id and rideable_type to character

```
q1_2019 <- mutate(q1_2019, ride_id = as.character(ride_id)
                  ,rideable_type = as.character(rideable_type))
```

Merging q1_2019 and q1_2020 into one date frame

```
all_trips <- bind_rows(q1_2019, q1_2020)
```

Removing lat, long, birthyear, and gender fields as this data was dropped beginning in 2020

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, "tripduration"))
```

3- Process phase: Documentation of cleaning or manipulation of data

Inspecting the new table that has been created by listing column names, Displaying number rows, Listing columns and data types and Getting statistical summary of data

```
colnames(all_trips)
```

```
## [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)
```

```
## [1] 791956
```

```
str(all_trips)
```

```
## tibble [791,956 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:791956] "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at        : POSIXct[1:791956], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:791956], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : chr [1:791956] "2167" "4386" "1524" "252" ...
##  $ start_station_id  : num [1:791956] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:791956] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:791956] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:791956] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ member_casual     : chr [1:791956] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
```

```
summary(all_trips)
```

```
##    ride_id              started_at
##  Length:791956      Min.   :2019-01-01 00:04:37.00
##  Class :character   1st Qu.:2019-02-28 17:04:04.75
##  Mode  :character   Median :2020-01-07 12:48:50.50
##                     Mean   :2019-09-01 11:58:08.35
##                     3rd Qu.:2020-02-19 19:31:54.75
##                     Max.   :2020-03-31 23:51:34.00
##
##     ended_at                      rideable_type      start_station_id
##  Min.   :2019-01-01 00:11:07.00   Length:791956      Min.   :  2.0
##  1st Qu.:2019-02-28 17:15:58.75   Class :character   1st Qu.: 77.0
##  Median :2020-01-07 13:02:50.00   Mode  :character   Median :174.0
##  Mean   :2019-09-01 12:17:52.17                      Mean   :204.4
##  3rd Qu.:2020-02-19 19:51:54.50                      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:791956      Min.   :  2.0   Length:791956      Length:791956
##  Class :character   1st Qu.: 77.0   Class :character   Class :character
##  Mode  :character   Median :174.0   Mode  :character   Mode  :character
##                     Mean   :204.4
##                     3rd Qu.:291.0
##                     Max.   :675.0
##                     NA's   :1
```

Seeing how many observations fall under each usertype

```
table(all_trips$member_casual)
```

```
##
##     casual   Customer     member Subscriber
##      48480      23163     378407     341906
```

Reassigning to the desired values and checking the result

```
all_trips <- all_trips %>%
  mutate(member_casual = recode(member_casual
                                ,"Subscriber" = "member"
                                ,"Customer" = "casual"))
table(all_trips$member_casual)
```

```
##
## casual member
##  71643 720313
```

Formatting date by adding columns in order to list date, month, day and year

```
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")
```

Adding "ride_length" and inspecting the structure of the columns

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
str(all_trips)
```

```
## tibble [791,956 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:791956] "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at        : POSIXct[1:791956], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:791956], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : chr [1:791956] "2167" "4386" "1524" "252" ...
##  $ start_station_id  : num [1:791956] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:791956] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:791956] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:791956] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "U
##  $ member_casual     : chr [1:791956] "member" "member" "member" "member" ...
##  $ date              : Date[1:791956], format: "2019-01-01" "2019-01-01" ...
##  $ month             : chr [1:791956] "01" "01" "01" "01" ...
##  $ day               : chr [1:791956] "01" "01" "01" "01" ...
##  $ year              : chr [1:791956] "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr [1:791956] "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ ride_length       : 'difftime' num [1:791956] 390 441 829 1783 ...
##   ..- attr(*, "units")= chr "secs"
```

Converting "ride_length" to numeric

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)
```

```
## [1] TRUE
```

Creating all_trips_v2 new version of data frame

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]
```

4- Analyze phase: Summary of your analysis

Calculating average(total ride length / rides), median(midpoint number), max(longest ride) and min()

```
all_trips_v2 %>%
  summarise(mean_ride_trip = round(mean(ride_length), 2)
            , median_ride_trip = median(ride_length)
            , max_ride_trip = max(ride_length)
            , min_ride_trip = min(ride_length))
```

```
## # A tibble: 1 x 4
##   mean_ride_trip median_ride_trip max_ride_trip min_ride_trip
##            <dbl>            <dbl>         <dbl>         <dbl>
## 1          1189.              539      10632022             1
```

Comparing members and casual users

```
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                5372.7839
## 2                     member                 795.2523
```

```
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                      1393
## 2                     member                       508
```

```
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                10632022
## 2                     member                 6096428
```

```
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
## 2                     member                         1
```

Calculating average ride time by each day for members vs casual users

```r
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                6090.7373
## 2                      member                   Friday                 796.7338
## 3                      casual                   Monday                4752.0504
## 4                      member                   Monday                 822.3112
## 5                      casual                 Saturday                4950.7708
## 6                      member                 Saturday                 974.0730
## 7                      casual                   Sunday                5061.3044
## 8                      member                   Sunday                 972.9383
## 9                      casual                 Thursday                8451.6669
## 10                     member                 Thursday                 707.2093
## 11                     casual                  Tuesday                4561.8039
## 12                     member                  Tuesday                 769.4416
## 13                     casual                Wednesday                4480.3724
## 14                     member                Wednesday                 711.9838
```

Fixing ordering issue and running the average ride time by each day for members vs casual users

```r
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday",
                                                                       "Tuesday", "Wednesday", "Thursda
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                5061.3044
## 2                      member                   Sunday                 972.9383
## 3                      casual                   Monday                4752.0504
## 4                      member                   Monday                 822.3112
## 5                      casual                  Tuesday                4561.8039
## 6                      member                  Tuesday                 769.4416
## 7                      casual                Wednesday                4480.3724
## 8                      member                Wednesday                 711.9838
## 9                      casual                 Thursday                8451.6669
## 10                     member                 Thursday                 707.2093
## 11                     casual                   Friday                6090.7373
## 12                     member                   Friday                 796.7338
## 13                     casual                 Saturday                4950.7708
## 14                     member                 Saturday                 974.0730
```

Analyzing ridership data by type and weekday

```r
all_trips_v2 %>%
  mutate(weekday = weekdays(started_at)) %>%
group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
```

```
            , average_duration = round(mean(ride_length), 2)) %>%
arrange(member_casual, weekday)
```

```
## `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>         <chr>               <int>            <dbl>
##  1 casual        Friday               8013            6091.
##  2 casual        Monday               5591            4752.
##  3 casual        Saturday            13473            4951.
##  4 casual        Sunday              18652            5061.
##  5 casual        Thursday             7147            8452.
##  6 casual        Tuesday              7311            4562.
##  7 casual        Wednesday            7690            4480.
##  8 member        Friday             115168             797.
##  9 member        Monday             110430             822.
## 10 member        Saturday            59413             974.
## 11 member        Sunday              60197             973.
## 12 member        Thursday           125228             707.
## 13 member        Tuesday            127974             769.
## 14 member        Wednesday          121902             712.
```

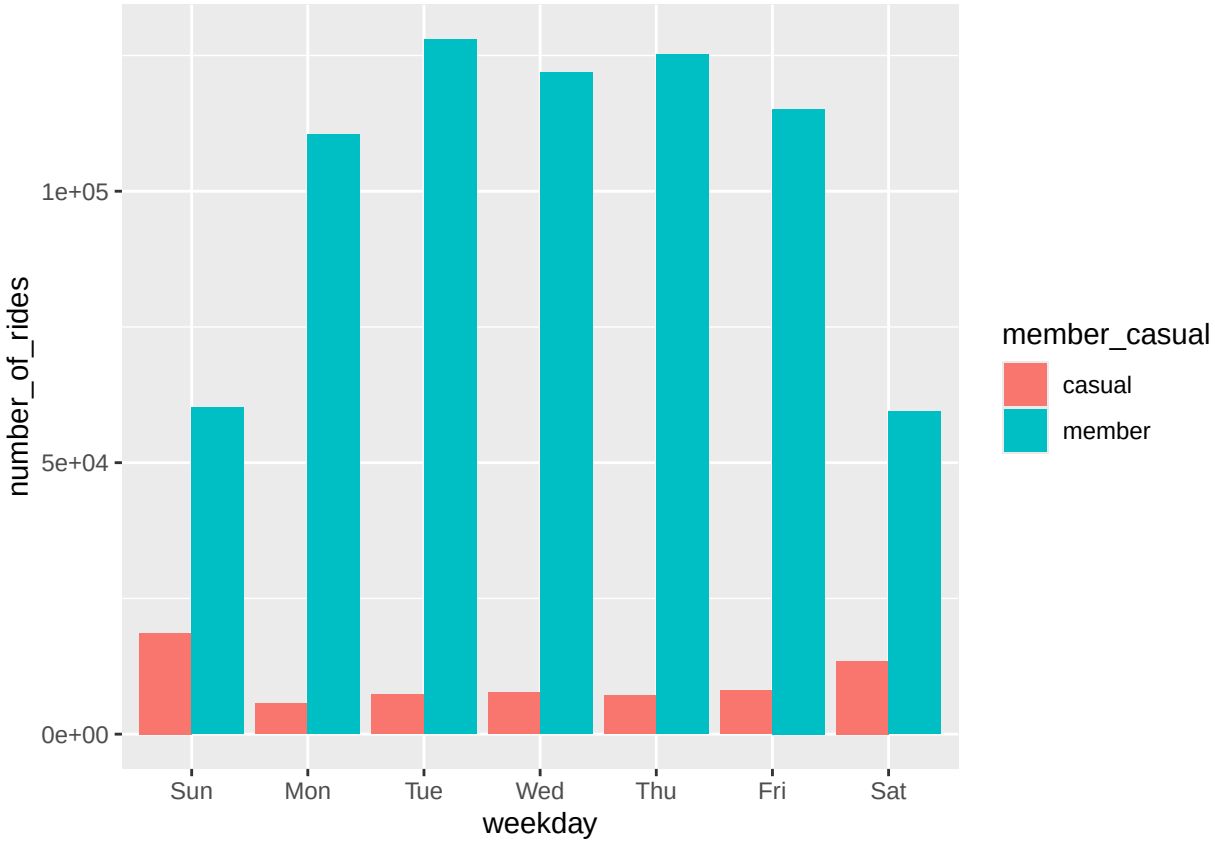
5- Share phase: Visualizations and key findings

Creating visualization for the 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), .group = "keep") %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

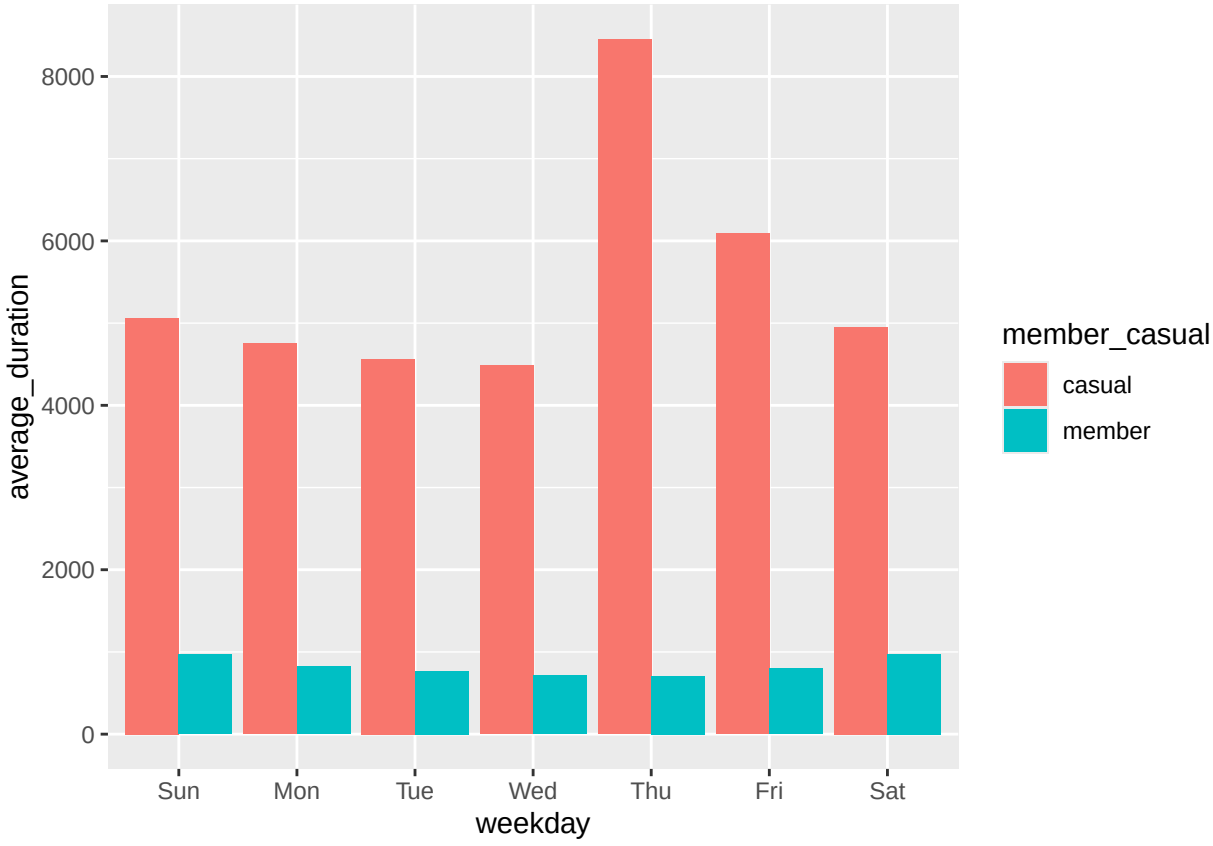
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



Creating visualization for 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), .group = "keep") %>%
  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.
```



Key finding

Our analyze reveals:

- Considering the ridership data by type and weekday that members do much more trips than casual users.
- Casual have a longer average duration compared to member

6- Act phase: Recommendations based on analysis

Considering the result of the analyze, I would invite the stakeholders to:

- Review the benefits and offerings of the member in order to encourage casual members to become annual members.
- Design new strategies to improve satisfaction and revisit the value proposition for annual members.
- Investigate more to determine if there are other factors that contribute to casual members to have a longer duration.