Cyclistic (case study)

Manny

2024-07-23

R Markdown

Vamos a analizar el caso de estudio de una empresa dedicada a rentar bicicletas compartidas, Cyclistic. Es una empresa ficticia que usará datos proporcionados por una empresa real, que sí se dedica a eso.

Escenario

Mi rol es el de un analista de datos júnior que trabaja en el equipo de marketing. Tu equipo quiere entender que diferencias en el uso de las bicicletas Cyclistic entre los ciclistas ocasionales y los miembros anuales. A través de estos conocimientos, tu equipo diseñará una nueva estrategia de marketing para convertir a los ciclistar ocasionales en miembros anuales.

La pregunta que nos toca resolver: *¿En qué se diferencian los socios anuales y los ciclistas ocasionales con respecto al uso de las bicicletas de Cyclistic?*

Preparemos nuestro entorno y nuestros datos

Una vez que decidimos trabajar con R, debido a la gran cantidad de datos que tenemos. Lo primero que hacemos es preparar el entorno que vamos a usar, cargando las librerias adecuadas.

```
library(tidyverse)  #ayuda a limpiar data
```

```
## -- 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(lubridate)  #ayuda a modificar los atributos de la data
library(ggplot2)  #para visualizar
```

Cargamos nuestros datasets

```
setwd("/Users/manuelvenegas/Desktop/cyclistic_case/csv")
# Upload Divvy datasets (csv files) here
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.
```

Comparamos las columnas

Es importante comparar las columnas para asegurarnos que nuestros datasets conservan un mismo formato a través del tiempo

```
colnames(q2_2019)
```

```
##  [1] "01 - Rental Details Rental ID"
##  [2] "01 - Rental Details Local Start Time"
##  [3] "01 - Rental Details Local End Time"
```

```
##  [4] "01 - Rental Details Bike ID"
##  [5] "01 - Rental Details Duration In Seconds Uncapped"
##  [6] "03 - Rental Start Station ID"
##  [7] "03 - Rental Start Station Name"
##  [8] "02 - Rental End Station ID"
##  [9] "02 - Rental End Station Name"
## [10] "User Type"
## [11] "Member Gender"
## [12] "05 - Member Details Member Birthday Year"
```

```
colnames(q3_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"
```

```
colnames(q4_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"
```

```
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"
```

Renombrar

Tras la comparación vemos que las columnas cambiaron de nombre a partir del 2020, hay que empatar los nombres para tener un formato homogeneo:

```
(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_type tripduration
##       <dbl> <dttm>              <dttm>                      <dbl>        <dbl>
##  1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20          2215          940
##  2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34          6328          258
##  3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43          3003          850
##  4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43          3275         2350
##  5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42          5294         1867
##  6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51          1891          373
```

```
##  7 25223646 2019-10-01 00:04:52 2019-10-01 00:22:45          1061         1072
##  8 25223647 2019-10-01 00:04:57 2019-10-01 00:29:16          1274         1458
##  9 25223648 2019-10-01 00:05:20 2019-10-01 00:29:18          6011         1437
## 10 25223649 2019-10-01 00:05:20 2019-10-01 02:23:46          2957         8306
## # i 704,044 more rows
## # i 7 more variables: start_station_id <dbl>, start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>
```

```
(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_type tripduration
##       <dbl> <dttm>              <dttm>                      <dbl>        <dbl>
##  1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41          3591         1214
##  2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44          5353         1048
##  3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42          6180         1554
##  4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10          5540         1503
##  5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26          6014         1213
##  6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31          4941          310
##  7 23479394 2019-07-01 00:02:24 2019-07-01 00:23:12          3770         1248
##  8 23479395 2019-07-01 00:02:26 2019-07-01 00:28:16          5442         1550
##  9 23479396 2019-07-01 00:02:34 2019-07-01 00:28:57          2957         1583
## 10 23479397 2019-07-01 00:02:45 2019-07-01 00:29:14          6091         1589
## # i 1,640,708 more rows
## # i 7 more variables: start_station_id <dbl>, start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>
```

```
(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_type
##       <dbl> <dttm>              <dttm>                      <dbl>
##  1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48          6251
##  2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30          6226
##  3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19          5649
```

```
##  4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58          4151
##  5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13          3270
##  6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56          3123
##  7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41          6418
##  8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11          4513
##  9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44          3280
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39          5534
## # i 1,108,153 more rows
## # i 8 more variables: `01 - Rental Details Duration In Seconds Uncapped` <dbl>,
## #   start_station_id <dbl>, 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>
```

Inspeccionamos los datasets

Ya que cambiamos las columnas, inspeccionamos otra vez los datasets

```
str(q1_2020)
```

```
## spc_tbl_ [426,887 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A38
##  $ 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" "Wilto
##  $ 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>
```

```
str(q4_2019)
```

```
## spc_tbl_ [704,054 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:704054] 25223640 25223641 25223642 25223643 25223644 ...
##  $ started_at        : POSIXct[1:704054], format: "2019-10-01 00:01:39" "2019-10-01 00:02:16" ...
```

```
##  $ ended_at          : POSIXct[1:704054], format: "2019-10-01 00:17:20" "2019-10-01 00:06:34" ...
##  $ rideable_type     : num [1:704054] 2215 6328 3003 3275 5294 ...
##  $ tripduration      : num [1:704054] 940 258 850 2350 1867 ...
##  $ start_station_id  : num [1:704054] 20 19 84 313 210 156 84 156 156 336 ...
##  $ start_station_name: chr [1:704054] "Sheffield Ave & Kingsbury St" "Throop (Loomis) St & Taylor St
##  $ end_station_id    : num [1:704054] 309 241 199 290 382 226 142 463 463 336 ...
##  $ end_station_name  : chr [1:704054] "Leavitt St & Armitage Ave" "Morgan St & Polk St" "Wabash Ave &
##  $ member_casual     : chr [1:704054] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender            : chr [1:704054] "Male" "Male" "Female" "Male" ...
##  $ birthyear         : num [1:704054] 1987 1998 1991 1990 1987 ...
##  - 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(q3_2019)
```

```
## spc_tbl_ [1,640,718 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:1640718] 23479388 23479389 23479390 23479391 23479392 ...
##  $ started_at        : POSIXct[1:1640718], format: "2019-07-01 00:00:27" "2019-07-01 00:01:16" ...
##  $ ended_at          : POSIXct[1:1640718], format: "2019-07-01 00:20:41" "2019-07-01 00:18:44" ...
##  $ rideable_type     : num [1:1640718] 3591 5353 6180 5540 6014 ...
##  $ tripduration      : num [1:1640718] 1214 1048 1554 1503 1213 ...
##  $ start_station_id  : num [1:1640718] 117 381 313 313 168 300 168 313 43 43 ...
##  $ start_station_name: chr [1:1640718] "Wilton Ave & Belmont Ave" "Western Ave & Monroe St" "Lakeview
##  $ end_station_id    : num [1:1640718] 497 203 144 144 62 232 62 144 195 195 ...
##  $ end_station_name  : chr [1:1640718] "Kimball Ave & Belmont Ave" "Western Ave & 21st St" "Larrabee
##  $ member_casual     : chr [1:1640718] "Subscriber" "Customer" "Customer" "Customer" ...
##  $ gender            : chr [1:1640718] "Male" NA NA NA ...
##  $ birthyear         : num [1:1640718] 1992 NA NA NA NA ...
##  - 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(q2_2019)
```

```
## spc_tbl_ [1,108,163 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id                                          : num [1:1108163] 22178529 22178530 22178531 2217
##  $ started_at                                       : POSIXct[1:1108163], format: "2019-04-01 00:02:2
##  $ ended_at                                         : POSIXct[1:1108163], format: "2019-04-01 00:09:4
##  $ rideable_type                                    : num [1:1108163] 6251 6226 5649 4151 3270 ...
##  $ 01 - Rental Details Duration In Seconds Uncapped: num [1:1108163] 446 1048 252 357 1007 ...
##  $ start_station_id                                 : num [1:1108163] 81 317 283 26 202 420 503 260 2
##  $ start_station_name                               : chr [1:1108163] "Daley Center Plaza" "Wood St &
##  $ end_station_id                                   : num [1:1108163] 56 59 174 133 129 426 500 499 2
##  $ end_station_name                                 : chr [1:1108163] "Desplaines St & Kinzie St" "Wal
##  $ member_casual                                    : chr [1:1108163] "Subscriber" "Subscriber" "Subs
##  $ Member Gender                                    : chr [1:1108163] "Male" "Female" "Male" "Male" .
##  $ 05 - Member Details Member Birthday Year         : num [1:1108163] 1975 1984 1990 1993 1992 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `01 - Rental Details Rental ID` = col_double(),
##   ..   `01 - Rental Details Local Start Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Local End Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Bike ID` = col_double(),
##   ..   `01 - Rental Details Duration In Seconds Uncapped` = col_number(),
##   ..   `03 - Rental Start Station ID` = col_double(),
##   ..   `03 - Rental Start Station Name` = col_character(),
##   ..   `02 - Rental End Station ID` = col_double(),
##   ..   `02 - Rental End Station Name` = col_character(),
##   ..   `User Type` = col_character(),
##   ..   `Member Gender` = col_character(),
##   ..   `05 - Member Details Member Birthday Year` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Convertimos datos incongruentes

Hay que convertir ride_id y rideable_type a character para que puedan empatar con los datos del 2020

```
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))
```

Creamos un dataframe grande con los 4 cuartetos

```
all_trips <- bind_rows(q2_2019, q3_2019, q4_2019, q1_2020)
```

Hay que quitar lat, long, birthyear,y gender ya que estas columnas se dejaron de incluir a partir del 2020

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, "01 - Rental Details Duration In
```

Inspeccionamos la nueva tabla que creamos

```
colnames(all_trips)  #Lista de columnas
```

```
## [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)  #Cuantas filas hay en el data frame?
```

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

```
dim(all_trips)  #Dimensiones del data frame
```

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

```
head(all_trips)  #Ver las primeras 6 filas del data frame
```

```
## # A tibble: 6 x 9
##   ride_id started_at          ended_at            rideable_type start_station_id
##   <chr>   <dttm>              <dttm>              <chr>                      <dbl>
## 1 221785~ 2019-04-01 00:02:22 2019-04-01 00:09:48 6251                        81
## 2 221785~ 2019-04-01 00:03:02 2019-04-01 00:20:30 6226                       317
## 3 221785~ 2019-04-01 00:11:07 2019-04-01 00:15:19 5649                       283
## 4 221785~ 2019-04-01 00:13:01 2019-04-01 00:18:58 4151                        26
## 5 221785~ 2019-04-01 00:19:26 2019-04-01 00:36:13 3270                       202
## 6 221785~ 2019-04-01 00:19:39 2019-04-01 00:23:56 3123                       420
## # i 4 more variables: start_station_name <chr>, end_station_id <dbl>,
## #   end_station_name <chr>, member_casual <chr>
```

```
tail(all_trips) #Ver las ultimas 6 filas del data frame
```

```
## # A tibble: 6 x 9
##   ride_id started_at          ended_at            rideable_type start_station_id
##   <chr>   <dttm>              <dttm>              <chr>                      <dbl>
## 1 6F4D22~ 2020-03-10 10:40:27 2020-03-10 10:40:29 docked_bike                675
## 2 ADDAA3~ 2020-03-10 10:40:06 2020-03-10 10:40:07 docked_bike                675
## 3 82B10F~ 2020-03-07 15:25:55 2020-03-07 16:14:03 docked_bike                161
## 4 AA0D5A~ 2020-03-01 13:12:38 2020-03-01 13:38:29 docked_bike                141
## 5 329636~ 2020-03-07 18:02:45 2020-03-07 18:13:18 docked_bike                672
## 6 064EC7~ 2020-03-08 13:03:57 2020-03-08 13:32:27 docked_bike                110
## # i 4 more variables: start_station_name <chr>, end_station_id <dbl>,
## #   end_station_name <chr>, member_casual <chr>
```

Agregamos columnas con la fecha, mes, dia, y año de cada viaje

```
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")
```

Agregamos un calculo "ride_length" a all_trips (en segundos)

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

Convertimos "ride_length" a valor numerico para poder hacer cálculos con los datos

```
is.factor(all_trips$ride_length)
```

```
## [1] FALSE
```

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

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

Creamos una nueva versión, quitando valores negativos del ride_length

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

Vemos los atributos de ride_length

```
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     412     712    1479    1289 9387024
```

Comparamos usuarios miembros y casual

```
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                6230.7734
## 2                   Customer                3413.1005
## 3                     member                 760.6287
## 4                 Subscriber                 863.1057
```

```
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                      1389
## 2                   Customer                      1554
## 3                     member                       515
## 4                 Subscriber                       601
```

```
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
## 2                   Customer                   8582302
## 3                     member                   5627611
```

```
## 4                  Subscriber                   9056634
```

```
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                   Customer                       61
## 3                     member                        1
## 4                 Subscriber                       61
```

Vemos el promedio del tiempo del ride de los miembros vs casual

```
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                7907.8883
## 2                    Customer                   Friday                3611.0229
## 3                      member                   Friday                 757.3241
## 4                  Subscriber                   Friday                 833.9182
## 5                      casual                   Monday                5818.3439
## 6                    Customer                   Monday                3281.4412
## 7                      member                   Monday                 778.6286
## 8                  Subscriber                   Monday                 852.2237
## 9                      casual                 Saturday                6017.1560
## 10                   Customer                 Saturday                3232.5111
## 11                     member                 Saturday                 929.9892
## 12                 Subscriber                 Saturday                 973.4804
## 13                     casual                   Sunday                5710.5665
## 14                   Customer                   Sunday                3390.9405
## 15                     member                   Sunday                 949.3401
## 16                 Subscriber                   Sunday                 915.4225
## 17                     casual                 Thursday                8744.6574
## 18                   Customer                 Thursday                3465.6636
## 19                     member                 Thursday                 693.2325
## 20                 Subscriber                 Thursday                 842.8539
## 21                     casual                  Tuesday                5832.3594
## 22                   Customer                  Tuesday                3477.1007
## 23                     member                  Tuesday                 692.0323
## 24                 Subscriber                  Tuesday                 847.4468
## 25                     casual                Wednesday                5132.6226
## 26                   Customer                Wednesday                3634.3811
## 27                     member                Wednesday                 699.5471
## 28                 Subscriber                Wednesday                 842.2466
```

Vemos que los días de la semana están en desorden, hay que acomodarlos

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "
```

Analizamos la data por tipo de membresía y día de la semana

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #crea el campo de día de la semana usando wday(
  group_by(member_casual, weekday) %>%  #agrupa por usertype y weekday
  summarise(number_of_rides = n()                           #calcula el número de viajes y el promedio
```

```
            ,average_duration = mean(ride_length)) %>%      # promedio de duración
  arrange(member_casual, weekday)                               # sorts
```

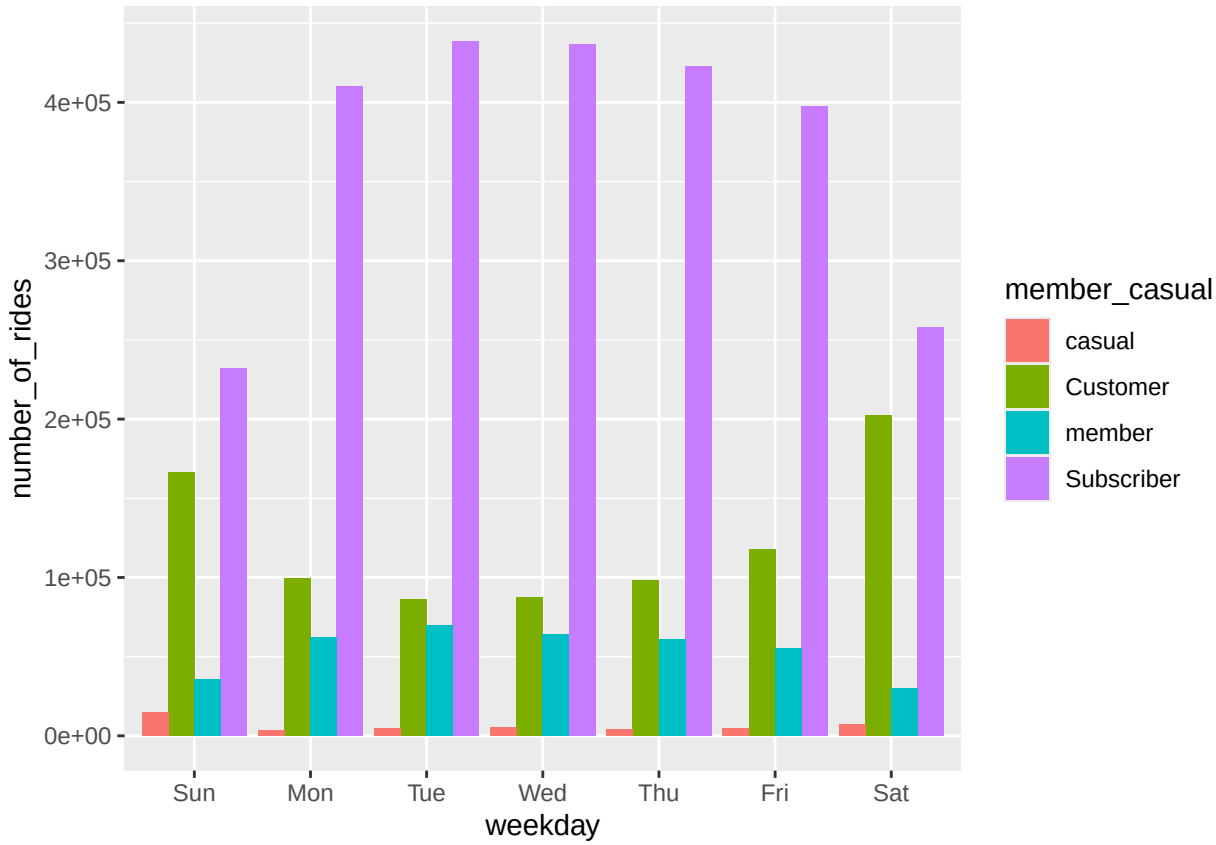
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 28 x 4
## # Groups:   member_casual [4]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 Customer      Sun              166407            3391.
##  2 Customer      Mon               99597            3281.
##  3 Customer      Tue               85927            3477.
##  4 Customer      Wed               87256            3634.
##  5 Customer      Thu               98452            3466.
##  6 Customer      Fri              117766            3611.
##  7 Customer      Sat              202063            3233.
##  8 Subscriber    Sun              232001             915.
##  9 Subscriber    Mon              410273             852.
## 10 Subscriber    Tue              438748             847.
## # i 18 more rows
```

Visualizamos el número de viajes por tipo de miembro

```
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")
```

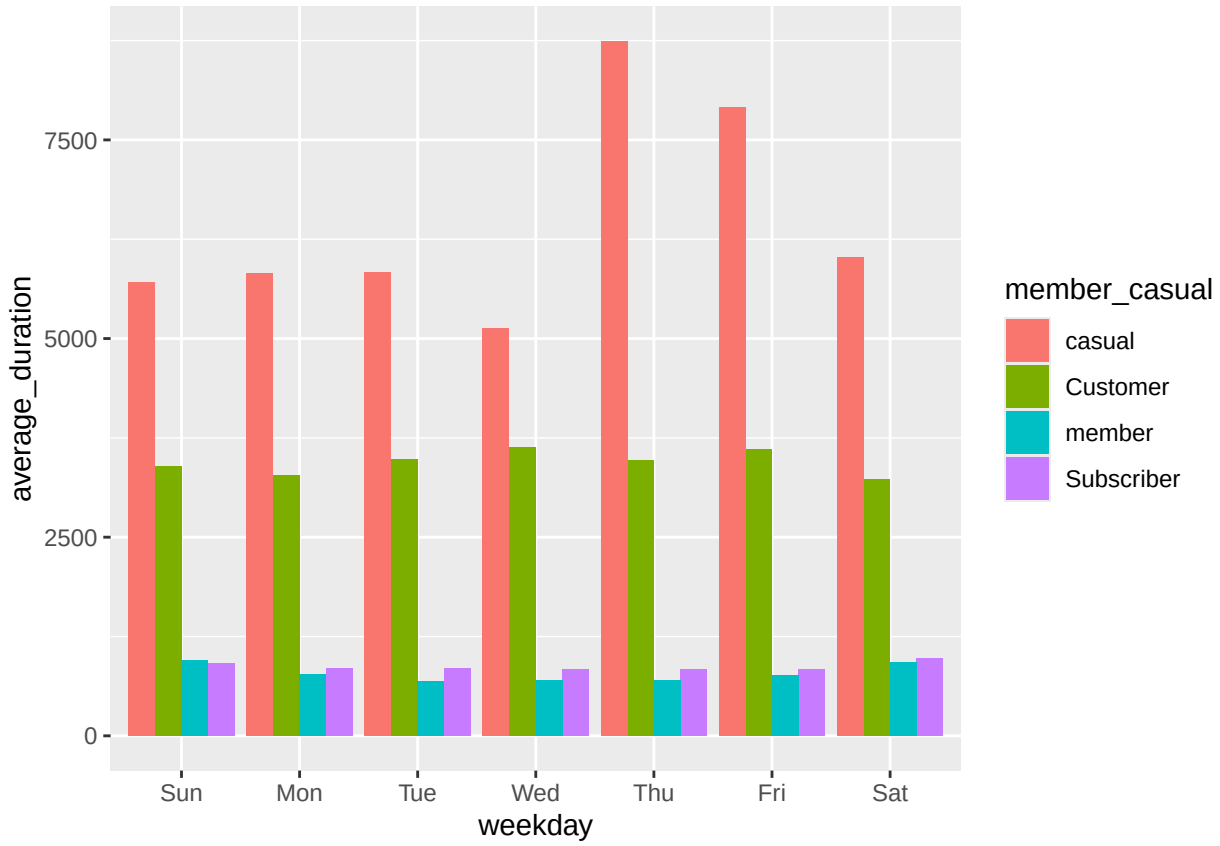
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



Visualizamos el promedio de duración del viaje

```
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.
```



Y así tenemos visualizaciones valiosas que nos ayudarán con nuestro análisis y podremos responder la pregunta inicial, aquí no incluimos el análisis ya que esté es un medio solo para documentar el proceso de limpieza y preparación de los datos