Cyclistic Analisis

Horacio Laphitz

2024-03-26

Historia de la Empresa

- En 2016, Cyclistic irrumpió en la escena con una oferta de bicicletas compartidas que fue un rotundo éxito. Desde entonces, el programa ha crecido exponencialmente hasta contar con una impresionante flota de 5,824 bicicletas, todas georreferenciadas y bloqueadas en una extensa red de 692 estaciones repartidas por toda la ciudad de Chicago.
- Las bicicletas de Cyclistic ofrecen una flexibilidad sin precedentes: puedes desbloquear una bicicleta en cualquier estación y devolverla en cualquier otra estación del sistema, ¡en cualquier momento!
- Este análisis se inspira en el estudio de caso de Divvy titulado " 'Sofisticado, claro y pulido': Divvy y visualización de datos", escrito por el renombrado Kevin Hartman. Puedes encontrar el estudio completo aca <[https://artscience.blog/home/divvy-dataviz-case-study\\](https://artscience.blog/home/divvy-dataviz-case-study){.uri}> .

El objetivo de mi script es consolidar los datos descargados de Divvy en un único marco de datos. A partir de ahí, realizo un análisis sencillo pero revelador para responder a la pregunta clave: "¿Cómo difieren los usos que los miembros y los ciclistas ocasionales hacen de las bicicletas Divvy?"

Importacion de las Liberias

Se procede con la importación de las librerías necesarias, *tidyverse* -> para análisis y manipulación de los datos , *readr* -> para lectura de los archivos separados por comas("csv's") , *lubridate* -> para el manejo de las fechas , *ggplot2* -> para los gráficos.

```
library("tidyverse")
library("readr")
library("lubridate")
library("ggplot2")
install.packages("devtools")
```

```
## package 'devtools' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\horac\AppData\Local\Temp\RtmpMTZ3E4\downloaded_packages
```

```
devtools::install_github("r-lib/conflicted")
```

Paso 1: Importacion de los Datos

```
q2_2019 <- read_csv("Divvy_Trips_2019_Q2.csv")
q3_2019 <- read_csv("Divvy_Trips_2019_Q3.csv")
q4_2019 <- read_csv("Divvy_Trips_2019_Q4.csv")
q1_2020 <- read_csv("Divvy_Trips_2020_Q1.csv")
```

Paso 2: ORDENAR DATOS Y COMBINAR EN UN ÚNICO ARCHIVO

Comparar los nombres de las columnas de cada uno de los archivos.

Si bien los nombres no tienen que estar en el mismo orden, SÍ deben coincidir perfectamente antes de que se pueda usar un comando para unirlos en un solo archivo.

```
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(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(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"
```

```r
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"
```

Cambie el nombre de las columnas para que sean coherentes con el primer trimestre de 2020 (ya que este será el supuesto diseño de tabla futuro para Divvy)

```r
(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>
```

```r
(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>
```

Inspeccionar los marcos de datos y buscar incongruencias.

```
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" "Lakevie
##  $ 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>
```

Convirtiendo ride_id y rideable_type en caracteres para que se puedan apilar correctamente

```
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))
```

Apilar los marcos de datos de cada trimestre en un marco de datos grande

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

Elimine los campos de latitud, longitud, año de nacimiento y género, ya que estos datos se eliminaron a partir de 2020.

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

PASO 3: LIMPIAR Y AGREGAR DATOS PARA PREPARAR EL ANÁLISIS

Inspeccionar la nueva tabla que se ha creado

Lista de nombres de columnas

```
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"
```

Filas del marco de datos

```
nrow(all_trips)
```

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

¿Dimensiones del marco de datos?

```
dim(all_trips)
```

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

Ver las primeras 6 filas del marco de datos. También cola (all_trips)

```
head(all_trips)
```

```
## # 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>
```

Ver lista de columnas y tipos de datos (numéricos, de caracteres, etc.)

```r
str(all_trips)
```

```
## 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" ...
```

Resumen estadístico de los datos. Principalmente para números

```r
summary(all_trips)
```

```
##    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
```

Hay algunos problemas que se deben solucionar:

1- En la columna "member_casual", hay dos nombres para miembros ("Member" y "Subscriber") y dos nombres para pasajeros ocasionales ("Customer" y "Casual"). Se necesitara consolidar eso de cuatro a dos etiquetas.

2- Los datos solo se pueden agregar a nivel de viaje, lo cual es demasiado granular.Se quiere agregar algunas columnas de datos adicionales como: día, mes, año, que brinden oportunidades adicionales para agregar los datos.

3- Quiero agregar un campo calculado para la duración del viaje, ya que los datos del primer trimestre de 2020 no tenían la columna "duración del viaje". Agregaremos "ride_length" a todo el marco de datos para mantener la coherencia.

4- Hay algunos viajes en los que la duración del viaje se muestra negativa, incluidos varios cientos de viajes en los que Divvy sacó de circulación las bicicletas por motivos de control de calidad. Querremos eliminar estos viajes.

En la columna "miembro_casual", reemplace "Suscriptor" por "miembro" y "Cliente" por "casual"

Antes de 2020, Divvy usaba etiquetas diferentes para estos dos tipos de ciclistas... quiero que mi marco de datos sea coherente con su nomenclatura actual.

N.B.: "Nivel" es una propiedad especial de una columna que se conserva incluso si un subconjunto no contiene ningún valor de un nivel específico

Comenzando por ver cuántas observaciones corresponden a cada tipo de usuario

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

```
##
##     casual   Customer     member Subscriber
##      48480     857474     378407    2595461
```

Reasigne los valores deseados (elijo las etiquetas actuales de 2020)

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

Verifique para asegurarme de que se haya reasignado el número adecuado de observaciones

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

```
##
##  casual  member
##  905954 2973868
```

Agregue columnas que enumeren la fecha, mes, día y año de cada viaje.

Esto permitirá agregar datos de viaje para cada mes, día o año... antes de completar estas operaciones solo podíamos agregar a nivel de viaje

```
all_trips$date <- as.Date(all_trips$started_at)
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")
```

Agregue un cálculo de "duración del viaje" a todos los viajes (en segundos)

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

Inspeccionar la estructura de las columnas.

```
str(all_trips)
```

```
## tibble [3,879,822 x 15] (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] "member" "member" "member" "member" ...
##  $ date              : Date[1:3879822], format: "2019-04-01" "2019-04-01" ...
##  $ month             : chr [1:3879822] "04" "04" "04" "04" ...
##  $ day               : chr [1:3879822] "01" "01" "01" "01" ...
##  $ year              : chr [1:3879822] "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr [1:3879822] "lunes" "lunes" "lunes" "lunes" ...
##  $ ride_length       : 'difftime' num [1:3879822] 446 1048 252 357 ...
##   ..- attr(*, "units")= chr "secs"
```

Convertí "ride_length" de Factor a numérico para que se pueda ejecutar cálculos sobre 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
```

Eliminar datos "Malos"

El marco de datos incluye algunos cientos de entradas cuando Divvy sacó las bicicletas de los muelles y Divvy verificó su calidad o la longitud de paseo fue negativa

Se Creara una nueva versión del marco de datos (v2) ya que se están eliminando datos

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

PASO 4: REALIZAR UN ANÁLISIS DESCRIPTIVO

Análisis descriptivo de ride_length (todas las cifras en segundos)

Promedio Directo (longitud total del recorrido / recorridos)

```
mean(all_trips_v2$ride_length)
```

```
## [1] 1479.139
```

Número de punto medio en el conjunto ascendente de longitudes de recorrido

```
median(all_trips_v2$ride_length)
```

```
## [1] 712
```

Longitud máximo de recorrido

```
max(all_trips_v2$ride_length)
```

```
## [1] 9387024
```

Longitud mínimo de recorrido

```
min(all_trips_v2$ride_length)
```

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

Se puede condensar las cuatro líneas anteriores en una línea usando summary() en el atributo específico

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

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

Comparacion miembros y usuarios ocasionales

```
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
## 2                     member                 850.0662
```

```
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
## 2                     member                       589
```

```
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                     member                   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                     member                         1
```

Visualizacion de tiempo promedio de viaje cada día para miembros frente a usuarios ocasionales

```
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                 domingo                3581.4054
## 2                      member                 domingo                 919.9746
## 3                      casual                  jueves                3682.9847
## 4                      member                  jueves                 823.9278
## 5                      casual                   lunes                3372.2869
```

```
## 6                    member                  lunes                842.5726
## 7                    casual                 martes               3596.3599
## 8                    member                 martes                826.1427
## 9                    casual              miércoles               3718.6619
## 10                   member              miércoles                823.9996
## 11                   casual                 sábado               3331.9138
## 12                   member                 sábado                968.9337
## 13                   casual                viernes               3773.8351
## 14                   member                viernes                824.5305
```

Observe que los días de la semana están desordenados.Se va a arreglar eso

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Domingo", "Lunes", "Martes", "M
```

Ahora,se calcula el tiempo promedio de viajes de cada día para miembros y usuarios ocasionales

```
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
## 2                     member                 850.0662
```

Se analizan datos de número de pasajeros por tipo y día de la semana

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #crea un campo de día laborable usando wday()
  group_by(member_casual, weekday) %>%  #grupos por tipo de usuario y día de la semana
  summarise(number_of_rides = n()                            #calcula el número de viajes y la duración
  ,average_duration = mean(ride_length)) %>%        #calcula la duración media
  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>         <ord>              <int>            <dbl>
##  1 casual        "dom\\."          181293            3581.
##  2 casual        "lun\\."          103296            3372.
##  3 casual        "mar\\."           90510            3596.
##  4 casual        "mié\\."           92457            3719.
##  5 casual        "jue\\."          102679            3683.
##  6 casual        "vie\\."          122404            3774.
##  7 casual        "sáb\\."          209543            3332.
```

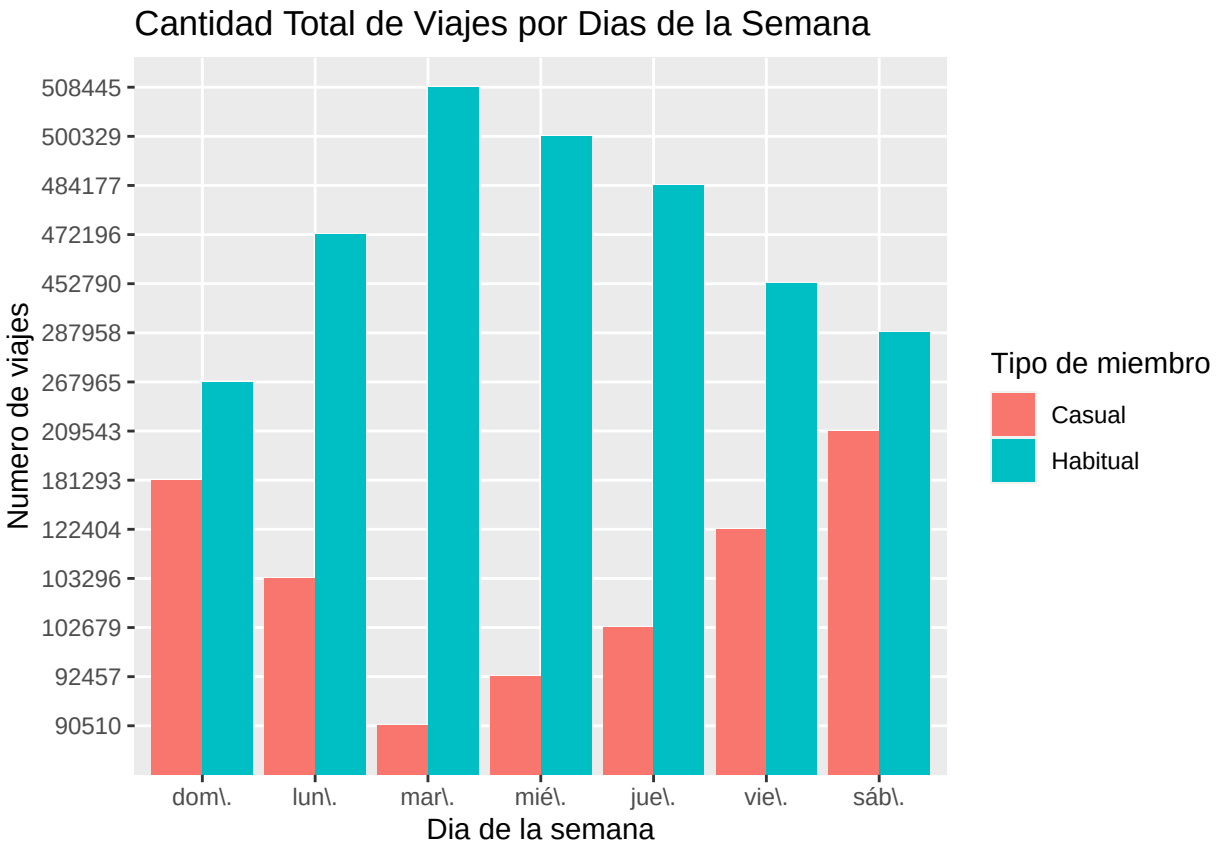
```
##  8 member        "dom\\."          267965             920.
##  9 member        "lun\\."          472196             843.
## 10 member        "mar\\."          508445             826.
## 11 member        "mié\\."          500329             824.
## 12 member        "jue\\."          484177             824.
## 13 member        "vie\\."          452790             825.
## 14 member        "sáb\\."          287958             969.
```

Visualizacion del número de viajes por tipo de ciclista

Se crea una visualización de la cantidad de los viajes realizados, por tipo de ciclista

```
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)) %>%
mutate(number_of_rides = format(number_of_rides, scientific = FALSE)) %>%
arrange(member_casual, weekday) %>%
ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
geom_col(position = "dodge") +
labs(x = "Dia de la semana", y = "Numero de viajes", title = "Cantidad Total de Viajes por Dias de la S
scale_fill_discrete(name = "Tipo de miembro", labels = c("Casual", "Habitual"))
```

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

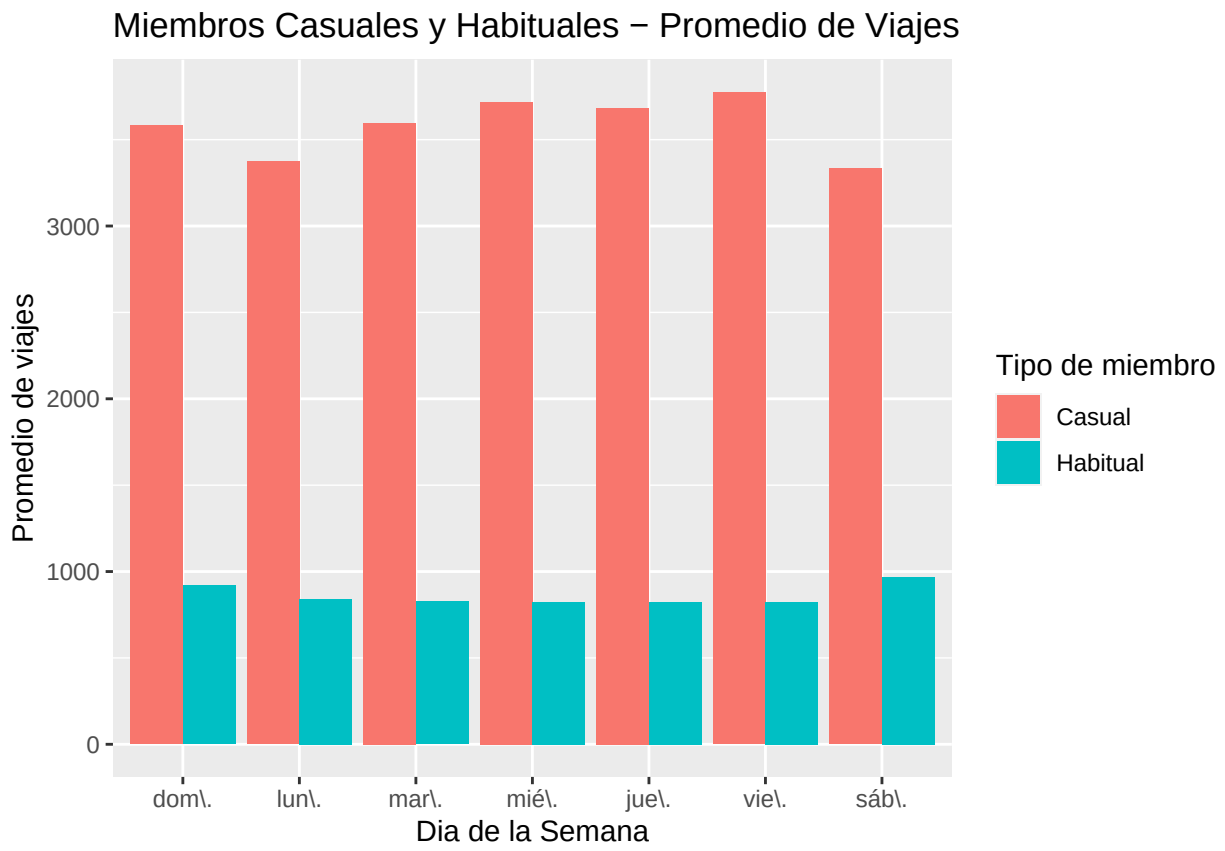


- Este gráfico de barras compara el número total de viajes realizados por los miembros casuales y habituales en diferentes días de la semana. Se destaca un aumento notable en los viajes realizados por los miembros habituales durante los días de semana. *Esto podría indicar que los miembros habituales de Cyclistic utilizan más frecuentemente el servicio para desplazamientos regulares, como ir al trabajo o a la escuela, mientras que los miembros casuales podrían estar utilizando el servicio de manera más esporádica o para actividades recreativas durante los fines de semana.* Estos patrones de uso pueden ayudar a Cyclistic a adaptar sus servicios y promociones para satisfacer mejor las necesidades de sus usuarios. Por ejemplo, podrían ofrecer tarifas especiales para los miembros habituales durante los días de semana o promociones para los miembros casuales durante los fines de semana.

Visualizacion del promedio de viajes realizado por tipo de ciclista

```
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") +
  labs(x = "Dia de la Semana", y = "Promedio de viajes", title = "Miembros Casuales y Habituales - Prome
scale_fill_discrete(name = "Tipo de miembro", labels = c("Casual", "Habitual"))
```

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



- El gráfico muestra la comparación de la cantidad promedio de viajes realizados por los miembros casuales y habituales de Cyclistic durante cada día de la semana. *Se observa que los miembros casuales realizan consistentemente más viajes que los habituales, independientemente del día de la semana. Esto podría indicar que los miembros casuales utilizan las bicicletas de Cyclistic con más frecuencia para actividades recreativas o esporádicas, mientras que los miembros habituales podrían tener un patrón de uso más regular y predecible. Esta información puede ser útil para Cyclistic al planificar estrategias de marketing y operaciones.* Por ejemplo, podrían ofrecer promociones especiales para los días de la semana cuando la actividad de los miembros casuales es alta para atraer a más usuarios ocasionales.

PASO 5: EXPORTAR EL ARCHIVO DE RESUMEN PARA UN ANÁLISIS ADICIONAL

Crear un archivo csv que se puede visualizar en Excel, Tableau o cualquier software de presentación, en el caso de ser necesario , en esta ocasion se omite este paso

```
#counts <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual , FUN = mean)
#write.csv(counts, file = 'cloud/project/duracion_media_viajes.csv')
```