Cyclistic Users

Natalia Romanini

2023-11-06

Usuarios Cyclistic

Tarea:

- Determinar las diferencias entre los usuarios que son miembros de la aplicacién, a través de una subscripción, de aquellos que son usuarios ocasionales

Data disponible

Se solicita se usen los 12 últimos meses de registro de los usuarios, obtenidos desde el siguiente enlace: https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html) Al descargar los archivos, son 12 carpetas en .zip y cada una contiene un mes de registros, se descomprimen las carpetas y se renombra cada uno de los archivos .csv a fin de comenzar a trabjar con ellos desde R Studio por la magnitud y peso de la data

Preparamos nuestro espacio de trabajo

Creamos una carpeta que contiene 12 archivos .csv y para poder importarlos, ordenarlos y limpiarlos y luego trabajar con ellos y realizar visualizaciones, instalamos y llamamos las siguientes librerias.

```
install.packages("tidyverse")
```

```
## Installing package into 'C:/Users/natal/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\natal\AppData\Local\Temp\RtmpYr5Tdd\downloaded_packages
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
```

```
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
install.packages("gglopt2")
```

```
## Installing package into 'C:/Users/natal/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## Warning: package 'gglopt2' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
library(ggplot2)
install.packages("geosphere")
```

```
## Installing package into 'C:/Users/natal/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'geosphere' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'geosphere'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\natal\AppData\Local\R\win-library\4.3\00LOCK\geosphere\libs\x64\geosphere.dll
## to
## C:\Users\natal\AppData\Local\R\win-library\4.3\geosphere\libs\x64\geosphere.dll:
## Permission denied
```

```
## Warning: restored 'geosphere'
```

```
##
## The downloaded binary packages are in
##  C:\Users\natal\AppData\Local\Temp\RtmpYr5Tdd\downloaded_packages
```

```
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.3.2
```

```
library(wesanderson)


month1 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month1_oct2022.csv')
month2 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month2_nov2022.csv')
month3 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month3_dic2022.csv')
month4 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month4_ene2023.csv')
month5 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month5_feb2023.csv')
month6 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month6_mar2023.csv')
month7 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month7_abr2023.csv')
month8 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month8_may2023.csv')
month9 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month9_jun2023.csv')
month10 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month10_jul2023.csv')
month11 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month11_ago2023.csv')
month12 <- read.csv('C:/Users/natal/OneDrive/Documents/bike_data/month12_sep2023.csv')
```

(las rutas mostradas aqui son de PC y de archivos en un disco local)

Revisando la data, sus columnas y tipos de datos.

- Se revisan las columnas de 3 archivos.

```
colnames(month1)
```

```
##  [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"
```

```
colnames(month6)
```

```
##  [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"
```

```
colnames(month12)
```

```
##  [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"
```

- Coinciden en la cantidad de columnas y nombres, ahora revisemos el tipo de dato

```
glimpse(month3)
```

```
## Rows: 181,806
## Columns: 13
## $ ride_id            <chr> "65DBD2F447EC51C2", "0C201AA7EA0EA1AD", "E0B148CCB3…
## $ rideable_type      <chr> "electric_bike", "classic_bike", "electric_bike", "…
## $ started_at         <chr> "2022-12-05 10:47:18", "2022-12-18 06:42:33", "2022…
## $ ended_at           <chr> "2022-12-05 10:56:34", "2022-12-18 07:08:44", "2022…
## $ start_station_name <chr> "Clifton Ave & Armitage Ave", "Broadway & Belmont A…
## $ start_station_id   <chr> "TA1307000163", "13277", "TA1306000015", "KA1503000…
## $ end_station_name   <chr> "Sedgwick St & Webster Ave", "Sedgwick St & Webster…
## $ end_station_id     <chr> "13191", "13191", "13016", "13134", "13288", "KA150…
## $ start_lat          <dbl> 41.91824, 41.94011, 41.88592, 41.83846, 41.89595, 4…
## $ start_lng          <dbl> -87.65711, -87.64545, -87.65113, -87.63541, -87.667…
## $ end_lat            <dbl> 41.92217, 41.92217, 41.89435, 41.88137, 41.92008, 4…
## $ end_lng            <dbl> -87.63889, -87.63889, -87.62280, -87.67493, -87.677…
## $ member_casual      <chr> "member", "casual", "member", "member", "casual", "…
```

```
glimpse(month8)
```

```
## Rows: 604,827
## Columns: 13
## $ ride_id            <chr> "0D9FA920C3062031", "92485E5FB5888ACD", "FB144B3FC8…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <chr> "2023-05-07 19:53:48", "2023-05-06 18:54:08", "2023…
## $ ended_at           <chr> "2023-05-07 19:58:32", "2023-05-06 19:03:35", "2023…
## $ start_station_name <chr> "Southport Ave & Belmont Ave", "Southport Ave & Bel…
## $ start_station_id   <chr> "13229", "13229", "13162", "13196", "TA1308000047",…
## $ end_station_name   <chr> "", "", "", "Damen Ave & Cortland St", "Southport A…
## $ end_station_id     <chr> "", "", "", "13133", "13229", "TA1306000029", "1343…
## $ start_lat          <dbl> 41.93941, 41.93948, 41.85379, 41.89456, 41.95708, 4…
## $ start_lng          <dbl> -87.66383, -87.66385, -87.64672, -87.65345, -87.664…
## $ end_lat            <dbl> 41.93000, 41.94000, 41.86000, 41.91598, 41.93948, 4…
## $ end_lng            <dbl> -87.65000, -87.69000, -87.65000, -87.67733, -87.663…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
```

```
glimpse(month11)
```

```
## Rows: 771,693
## Columns: 13
## $ ride_id            <chr> "903C30C2D810A53B", "F2FB18A98E110A2B", "D0DEC7C94E…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <chr> "2023-08-19 15:41:53", "2023-08-18 15:30:18", "2023…
## $ ended_at           <chr> "2023-08-19 15:53:36", "2023-08-18 15:45:25", "2023…
## $ start_station_name <chr> "LaSalle St & Illinois St", "Clark St & Randolph St…
## $ start_station_id   <chr> "13430", "TA1305000030", "TA1305000030", "KA1504000…
## $ end_station_name   <chr> "Clark St & Elm St", "", "", "", "", "", "", "", ""…
## $ end_station_id     <chr> "TA1307000039", "", "", "", "", "", "", "", "", "",…
## $ start_lat          <dbl> 41.89072, 41.88451, 41.88498, 41.90310, 41.88555, 4…
## $ start_lng          <dbl> -87.63148, -87.63155, -87.63079, -87.63467, -87.632…
## $ end_lat            <dbl> 41.90297, 41.93000, 41.91000, 41.90000, 41.89000, 4…
## $ end_lng            <dbl> -87.63128, -87.64000, -87.63000, -87.62000, -87.680…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
```

Al ver que todos los archivos comparten el mismo tipo de información, podemos unirlos en uno solo.

```
all_trips<-bind_rows(month1, month2, month3, month4, month5, month6, month7, month8, month9, mon
th10, month11, month12)
```

Modifiquemos algunos nombres de columnas y sus características y creemos una columna que cuente el tiempo de cada viaje y otra que nos señale el dia de esos viajes

```
all_trips <-rename(all_trips, trips_id = ride_id, start_time = started_at,
                   end_time = ended_at, usertype = member_casual)
```

Y revisemos nuestro dataframe

```
colnames(all_trips)
```

```
##  [1] "trips_id"           "rideable_type"      "start_time"
##  [4] "end_time"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "usertype"
```

```
nrow(all_trips)
```

```
## [1] 5674399
```

```
dim(all_trips)
```

```
## [1] 5674399      13
```

```
head(all_trips)
```

```
##           trips_id rideable_type          start_time            end_time
## 1 A50255C1E17942AB  classic_bike 2022-10-14 17:13:30 2022-10-14 17:19:39
## 2 DB692A70BD2DD4E3 electric_bike 2022-10-01 16:29:26 2022-10-01 16:49:06
## 3 3C02727AAF60F873 electric_bike 2022-10-19 18:55:40 2022-10-19 19:03:30
## 4 47E653FDC2D99236 electric_bike 2022-10-31 07:52:36 2022-10-31 07:58:49
## 5 8B5407BE535159BF  classic_bike 2022-10-13 18:41:03 2022-10-13 19:26:18
## 6 A177C92E9F021B99 electric_bike 2022-10-13 15:53:27 2022-10-13 15:59:17
##          start_station_name start_station_id
## 1  Noble St & Milwaukee Ave            13290
## 2 Damen Ave & Charleston St            13288
## 3  Hoyne Ave & Balmoral Ave              655
## 4        Rush St & Cedar St     KA1504000133
## 5         900 W Harrison St            13028
## 6         900 W Harrison St            13028
##                      end_station_name end_station_id start_lat start_lng
## 1            Larrabee St & Division St   KA1504000079  41.90068 -87.66260
## 2             Damen Ave & Cullerton St          13089  41.92004 -87.67794
## 3             Western Ave & Leland Ave   TA1307000140  41.97988 -87.68190
## 4 Orleans St & Chestnut St (NEXT Apts)            620  41.90227 -87.62769
## 5                    Adler Planetarium          13431  41.87475 -87.64981
## 6             Loomis St & Lexington St          13332  41.87472 -87.64983
##    end_lat   end_lng usertype
## 1 41.90349 -87.64335   member
## 2 41.85497 -87.67570   casual
## 3 41.96640 -87.68870   member
## 4 41.89820 -87.63754   member
## 5 41.86610 -87.60727   casual
## 6 41.87219 -87.66150   casual
```

```
str(all_trips)
```

```
## 'data.frame':    5674399 obs. of  13 variables:
##  $ trips_id          : chr  "A50255C1E17942AB" "DB692A70BD2DD4E3" "3C02727AAF60F873" "47E653F
DC2D99236" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ start_time        : chr  "2022-10-14 17:13:30" "2022-10-01 16:29:26" "2022-10-19 18:55:40"
"2022-10-31 07:52:36" ...
##  $ end_time          : chr  "2022-10-14 17:19:39" "2022-10-01 16:49:06" "2022-10-19 19:03:30"
"2022-10-31 07:58:49" ...
##  $ start_station_name: chr  "Noble St & Milwaukee Ave" "Damen Ave & Charleston St" "Hoyne Ave
& Balmoral Ave" "Rush St & Cedar St" ...
##  $ start_station_id  : chr  "13290" "13288" "655" "KA1504000133" ...
##  $ end_station_name  : chr  "Larrabee St & Division St" "Damen Ave & Cullerton St" "Western A
ve & Leland Ave" "Orleans St & Chestnut St (NEXT Apts)" ...
##  $ end_station_id    : chr  "KA1504000079" "13089" "TA1307000140" "620" ...
##  $ start_lat         : num  41.9 41.9 42 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.9 42 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.7 -87.6 -87.6 ...
##  $ usertype          : chr  "member" "casual" "member" "member" ...
```

```
summary(all_trips)
```

```
##    trips_id         rideable_type       start_time          end_time
##  Length:5674399     Length:5674399     Length:5674399     Length:5674399
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:5674399     Length:5674399     Length:5674399     Length:5674399
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.63   Min.   :-87.94   Min.   : 0.00   Min.   :-88.16
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.07   Max.   :-87.46   Max.   :42.18   Max.   :  0.00
##                                   NA's   :6642    NA's   :6642
##    usertype
##  Length:5674399
##  Class :character
##  Mode  :character
##
##
##
##
```

```
n_distinct(all_trips$usertype)
```

```
## [1] 2
```

Todo parece correcto.

Busquemos **valores faltantes** para conocer si afectaran nuestro analisis y deben retirar o son despreciables para el objetivo

```
trips_na <- all_trips %>% filter(if_any(everything(), is.na))
```

Podemos agrupar por tipo de usuario para revisar

```
trips_na %>% group_by(usertype) %>%
  count(usertype)
```

```
## # A tibble: 2 × 2
## # Groups:   usertype [2]
##   usertype     n
##   <chr>    <int>
## 1 casual    5671
## 2 member     971
```

Nos damos cuenta que hay 6642 registros que no tienen informacion de estacion de llegada y por tanto tampoco de su punto geografico y que esto principalmente afecta a usuarios “casual” pero la cantidad no es tanta en comparacion con la muestra total por lo que la podemos mantener, pero revisaremos si algo es anormal

```
trips_na <- trips_na %>% mutate(trip_duration = difftime(as.POSIXct(end_time), as.POSIXct(start_
time), units = "hours"))
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `trip_duration = difftime(as.POSIXct(end_time),
##   as.POSIXct(start_time), units = "hours")`.
## Caused by warning in `strptime()`:
## ! unknown timezone '%Y-%m-%d %H:%M:%S'
```

```
mean(trips_na$trip_duration)
```

```
## Time difference of 46.53996 hours
```

El tiempo promedio de viaje es superior a 46 horas, lo que no parece razonable, por lo que ** eliminaremos esos registros ** y agregamos una columna de duracion de viaje.

Últimas transformaciones de nuestro data frame.

Eliminemos las filas que no tiene informacion sobre el punto final del viaje y creemos una columna que determine la duracion de los viajes en horas

```
all_trips_clean <- all_trips[complete.cases(all_trips), ]
all_trips_clean <- all_trips_clean %>% mutate(trip_duration = difftime(as.POSIXct(end_time), as.
POSIXct(start_time), units = "hours"))
```

Para terminar las transformaciones del datafram, agregaremos una columna que nos señale la distancia que se recorrió en cada viaje y el dia de la semana en que se hizo

```
all_trips_clean <- all_trips_clean %>% mutate(distance=distHaversine(matrix(c(start_lng,start_la
t), ncol = 2), matrix(c(end_lng,end_lat), ncol=2)))

all_trips_clean <- all_trips_clean %>% mutate(weekday = wday(start_time, label = TRUE))
```

```
## Warning: There were 3 warnings in `mutate()`.
## The first warning was:
## ℹ In argument: `weekday = wday(start_time, label = TRUE)`.
## Caused by warning in `as.POSIXlt.POSIXct()`:
## ! unknown timezone '%Y-%m-%d %H:%M:%S'
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

Ahora podemos empezar a explorar los datos.

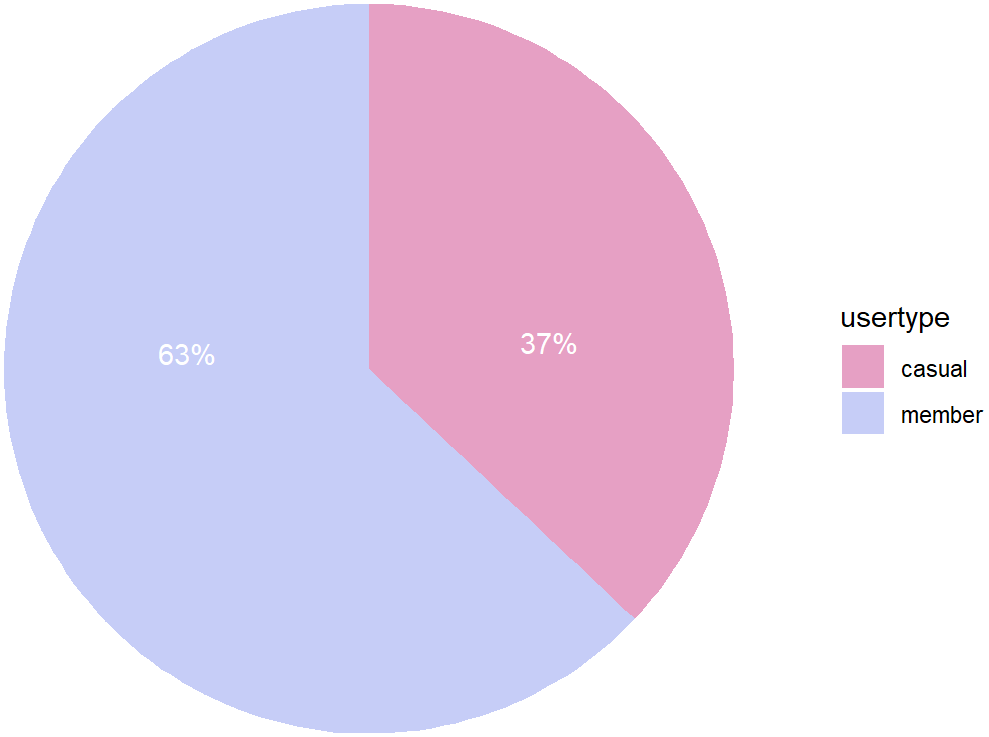
cuántos de los viajes los hicieron miembros y cuántos fueron hechos por usuarios casuales.(Nótese que no hay informacion para poder determinar si una persona determinada hizo uno o más viajes) Para eso, creamos una tabla.

```
usertypecantidad <- all_trips_clean %>% group_by(usertype) %>%
  count(usertype)
```

Y ahora a esos números le podemos dar mas sentido e informacion para graficar y que nos muestre un porcentaje:

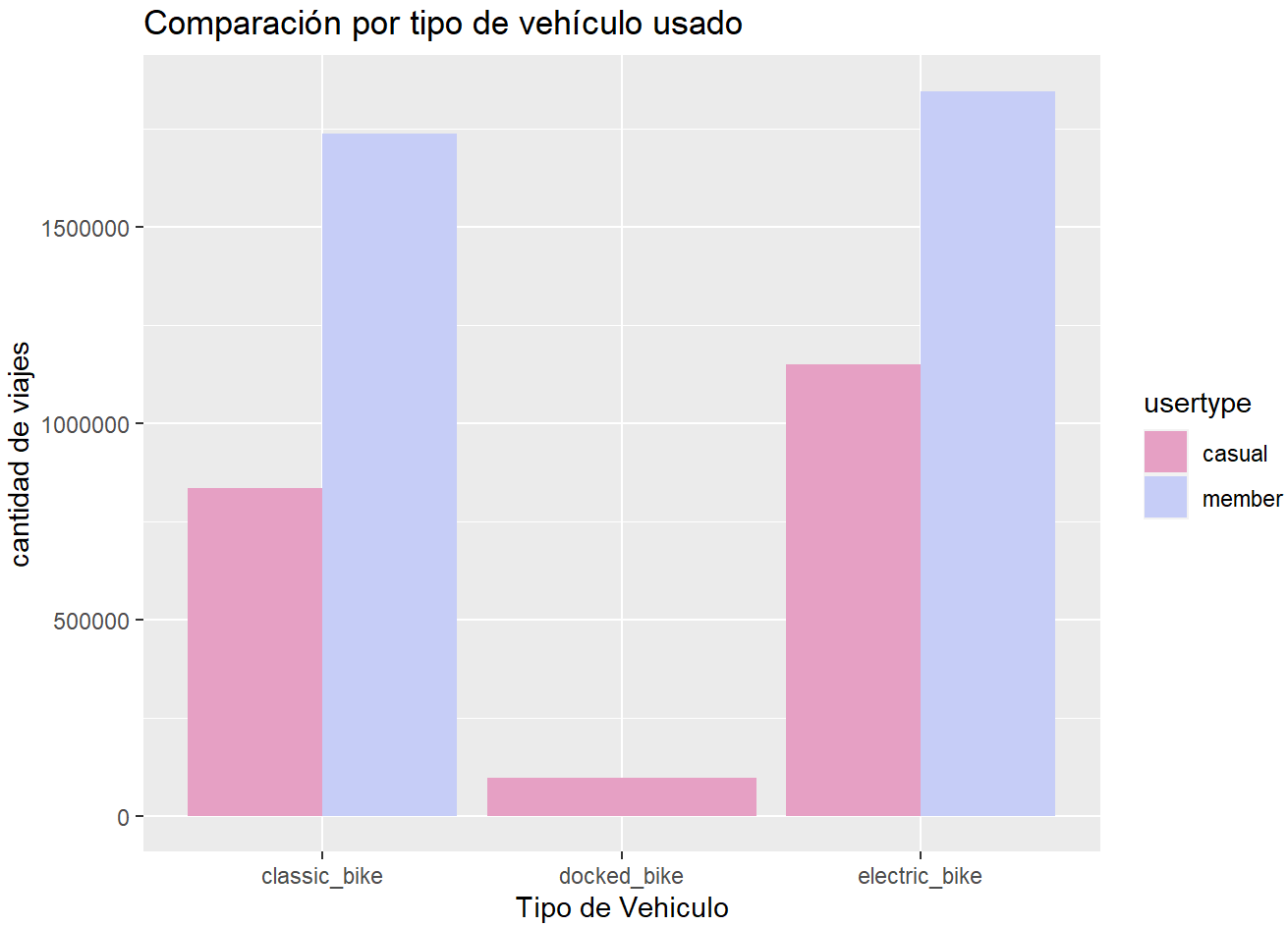
```
usertypecantidad <- usertypecantidad %>% mutate(pct = paste(round(n/sum(usertypecantidad$n)*10
0), "%", sep =""))
usertypecantidad <-usertypecantidad %>% mutate(cantidad = round(n/sum(usertypecantidad$n)*100))
usertypecantidad <-usertypecantidad %>% mutate(pct_y = 100 - cantidad )

ggplot(usertypecantidad, aes(x = 1, y = cantidad, fill = usertype)) +
  geom_col(position = "stack", orientation = "x") +
  geom_text(aes(x=1, y = pct_y, label = pct), col="white", position = position_stack(vjust = 0.6
4))+
  coord_polar(theta = "y", direction = -1) +
  theme_void() + scale_fill_manual(values=wes_palette(n=2, name="GrandBudapest2")) + ggtitle("Po
rcentaje de viajes realizado por cada tipo de usuario")
```

Porcentaje de viajes realizado por cada tipo de usuario



Podemos comparar los tipos de bicicleta que usa cada tipo de usuario.

```
ridetype <- all_trips_clean %>% group_by(rideable_type) %>%
  count(usertype)
ggplot(ridetype, aes(fill=usertype, y=n, x=rideable_type)) +
  geom_bar(position='dodge', stat="identity")+scale_fill_manual(values=wes_palette(n=2, name="Gr
andBudapest2")) + labs(x= "Tipo de Vehiculo", y ="cantidad de viajes") + ggtitle("Comparación po
r tipo de vehículo usado")
```



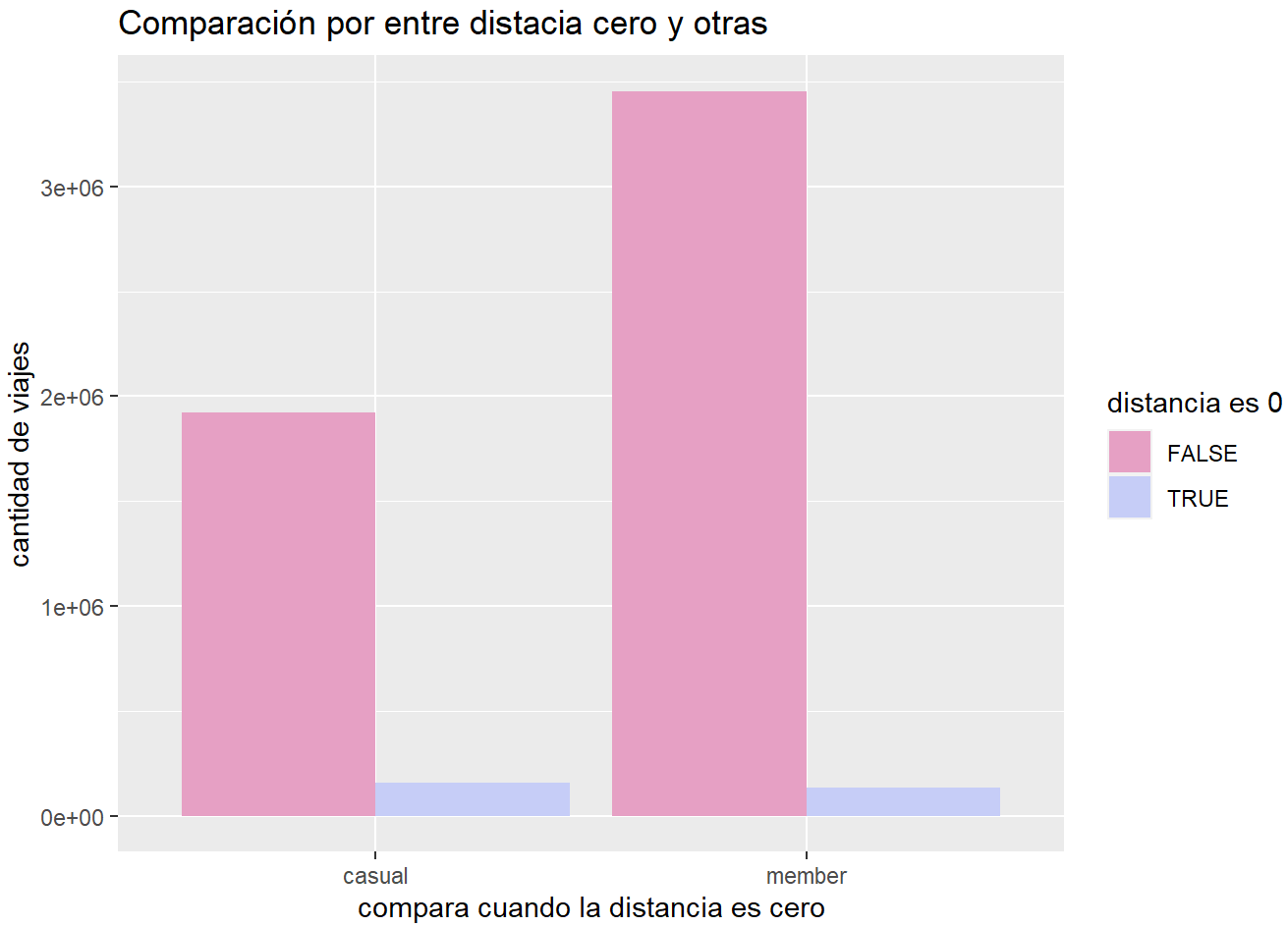
Aprendimos que solo los usuarios no miembros utilizan docked bikes, en los otros dos tipos la proporci[on de uso es similar

Sobre la distancia que recorre cada tipo de usuario

```
distanciausuario <- all_trips_clean %>% group_by(usertype) %>%
  summarise(Mean_distance = mean(distance, na.rm=TRUE), Min_distance= min(distance, na.rm=TRUE),
Max_distance = max(distance, na.rm=TRUE))

length(which(all_trips_clean$distance == 0))
```

```
## [1] 290942
```

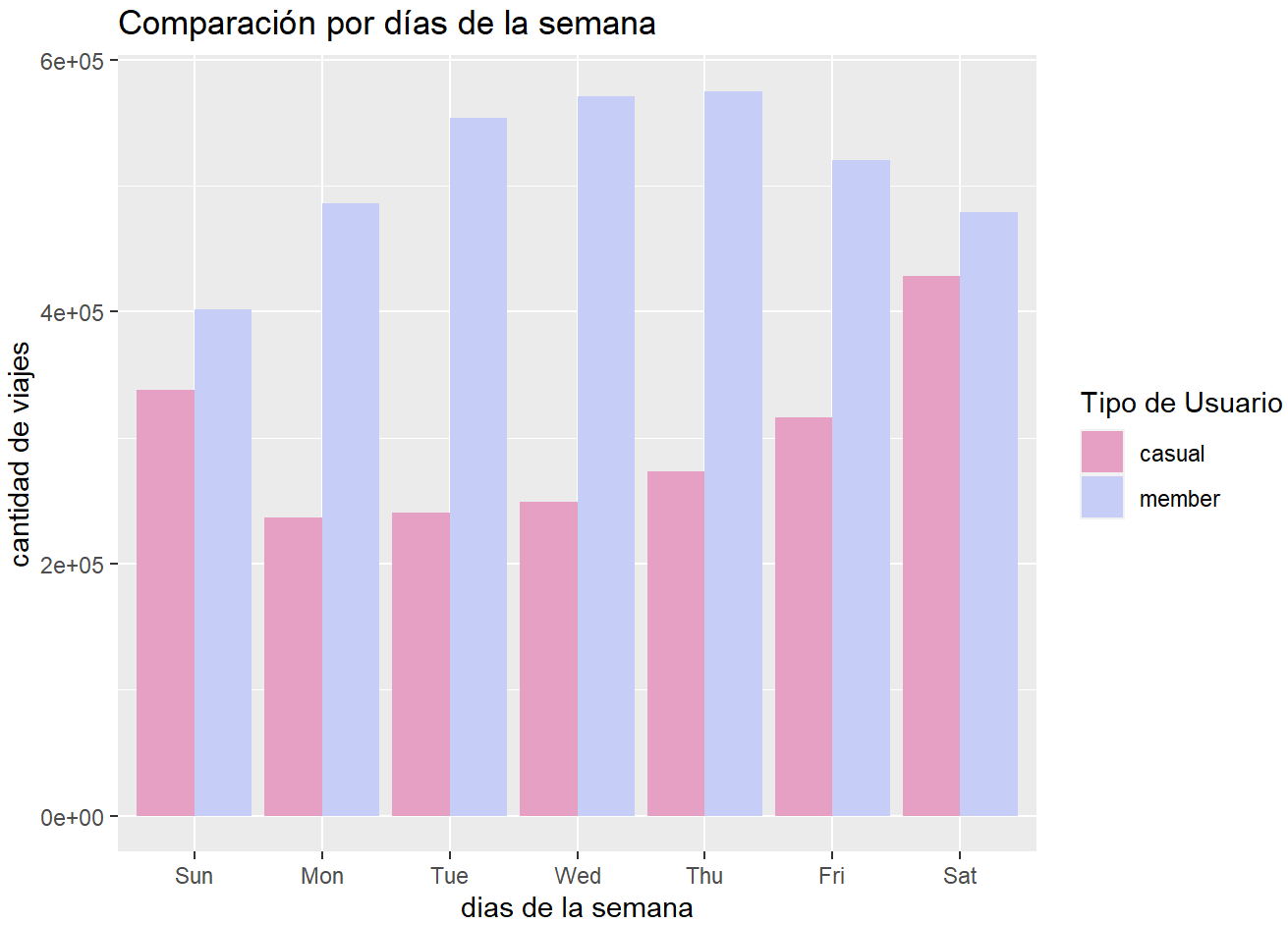
```
distancia0 <-all_trips_clean %>% group_by(usertype) %>%
  count(distance == 0)
ggplot(distancia0, aes(x=usertype, y=n, fill = `distance == 0`)) +
  geom_bar(position='dodge', stat="identity")+scale_fill_manual(values=wes_palette(n=2, name="Gr
andBudapest2")) + labs(x= "compara cuando la distancia es cero", y ="cantidad de viajes", fill =
"distancia es 0") + ggtitle("Comparación por entre distacia cero y otras")
```



No se aprecian diferencias respecto a la distancia que recorren los tipos de usuarios, pero existe un numero alto de viajes con distancia igual a 0, especialmente en usuarios casuales, si se compara con el total de viajes que cada tipo de usuario realiza

Días en que se utiliza el servicio. agreguemos una columna con el dia

```
weekdays <- all_trips_clean %>% group_by(usertype, weekday) %>%
  count(usertype)
ggplot(weekdays, aes(x = weekday, y = n, fill = usertype)) +
  geom_col(position = "dodge")+scale_fill_manual(values=wes_palette(n=2, name="GrandBudapest2"))
+ labs(x= "dias de la semana", y ="cantidad de viajes", fill = "Tipo de Usuario") + ggtitle("Com
paración por días de la semana")
```



Se aprecia que los usuarios con membresia utilizan mas el servicio en dias de semana y los no miembros fines de semana.

Conclusiones

Los datos utilizados no permiten responder completamente la pregunta, falta información sobre sexo, edad y cantidad de viajes que realiza cada usuario a fin de poder realizar un perfil detallado, pero si podemos concluir:

- Los usuarios con membresia representan el 67% del la cantidad de viajes
- Usuarios suscriptores utilizan mas el servicios los dias de semana.
- Solo los usuarios casuales utilizan el servicio de docked bikes.
- No se aprecian diferencias significativas en distancia media y distancia maxima entre los tipos de usuario
- la incidencia de falta de datos en la estacion de llegada es mayor en usuarios casuales.

Recomendaciones

- Subir los precios para no miembros los fines de semana, a ver si se convierten en miembros
- Se necesitan datos de viajes por usuario para revisar la conveniencia económica efectiva de los usuarios sean miembros.
- Deberia revisarse la app para ver por que falla al guardar la informacion de punto de término del viaje