

Reporte: diferencias entre miembros anuales y ocasionales de la empresa Cyclistic.

Tadeo Roberto Servin Ordoñez

2022-10-30

Introducción

En 2016, Cyclistic inició operaciones en la ciudad de Chicago. Actualmente, tiene una flota de 5,824 bicicletas georreferenciadas y bloqueadas en una red de 692 estaciones. El proceso de la renta de bicicletas se resume a: el usuario desbloquea en una estación una unidad y la devuelve en otra quedando nuevamente bloqueada y disponible para otro viaje.

Al día de hoy, existen planes de precios: pases de un solo viaje y pases de un día completo para los ciclistas ocasionales y membresías anuales para los miembros de Cyclistic.

Su departamento de finanzas asegura que los miembros anuales son mucho más rentables que los ciclistas ocasionales. La gerente de marketing, Lily Moreno, tiene la hipótesis de que aumentar el número de membresías anuales será importante financieramente. En lugar de propiciar el aumento clientes nuevos, la gerente ve la oportunidad de convertir a los ciclistas ocasionales en miembros anuales por el uso ya probado de las bicicletas rentadas.

Para ello, se requiere entender las diferencias entre los usuarios ciclistas ocasionales y los miembros que pagan una anualidad. Resultado de ese análisis, el equipo de marketing diseñará una nueva estrategia para convencer a los ciclistas de ocasión de convertirse en miembros anuales.

Paso 1: Preguntar

La meta de la gerente de marketing es: “Diseñar estrategias de marketing orientadas a convertir a los ciclistas ocasionales en miembros anuales.” Por lo que, el equipo de analistas de marketing necesita entender mejor en qué difieren los miembros anuales y los ciclistas ocasionales, por qué los ciclistas ocasionales comprarían una membresía y cómo los medios digitales podrían afectar sus tácticas de marketing.

En este reporte, se abordará la primera de las preguntas: “¿En qué se diferencian los socios anuales y los ciclistas ocasionales con respecto al uso de las bicicletas de Cyclistic?”. Por lo que, este análisis servirá poder ver patrones de comportamiento que apoyen la meta de la gerente de mercadotecnia.

¿Quiénes son las personas interesadas en este proyecto?

- Cyclistic: La empresa de renta de bicicletas compartidas tiene 5,800 unidades y 600 estaciones. Los usos más frecuentes de renta es para usos recreativos, pero alrededor del 30% la utiliza para ir al trabajo.
- Lily Moreno: La directora de marketing es responsable del desarrollo de campañas, en redes sociales y otros canales, para promover el programa de renta de bicicletas.
- Equipo de análisis computacional de datos de marketing de Cyclistic: se encarga de recopilar, analizar e informar datos que coadyuvan a la estrategia de marketing de la empresa. El autor de este informe, miembro de este equipo, trabaja desde hace seis meses como analista de datos júnior.

- Equipo ejecutivo de Cyclistic: Los ejecutivos de la empresa quienes decidirán si aprueba el programa de marketing recomendado, basado en el análisis de datos presente.

Paso 2: Preparar

Los datos para este análisis, son proporcionados por la propia empresa (para efectos de este ejercicio, la licencia de uso de estos datos se encuentra aquí.). Los archivos con la información son mensuales y del tipo .csv, se pide que se analice los datos de los últimos 12 meses, de octubre del 2021 a septiembre del 2022.

Los datos se organizan de esta manera:

Campo	Español	Tipo
ride_id	id de la unidad	caracter
rideable_type	tipo de bicicleta	caracter
started_at	comienza en (desbloqueo)	fecha-hora
ended_at	termina en (bloqueo)	fecha-hora
start_station_name	estación inicial	caracter
start_station_id	id estación inicial	caracter
end_station_name	estación final	caracter
end_station_id	id estación final	caracter
start_lat	latitud inicial	numérico
start_lng	longitud inicial	numérico
end_lat	latitud final	numérico
end_lng	longitud final	numérico
member_casual	miembro u ocasional	caracter

En una revisión rápida, todas las columnas de todos los archivos tienen el mismo nombre. Es de notar que hay muchos registros sin observaciones en los campos referentes a las estaciones. Además, no existen campos que refieran a id o nombre de usuario, por lo que se usará el número de viajes, la duración de los mismos y el tipo de usuario para el análisis.

Paso 3: Procesar

Se procede a procesar y limpiar los datos:

```
# Cargando librerías
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr    1.0.10
## v tidyverse 1.2.1     v stringr  1.4.1
## v readr   2.1.3      vforcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

library(lubridate)

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

```

library(ggplot2)
library(dplyr)
library(lessR)

##                                     feedback: gerbing@pdx.edu
## -----
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables.
##   Enter: browseVignettes("lessR")
##
## View changes in this and recent versions of lessR.
##   Enter: news(package="lessR")
##
## **New Feature**: Interactive analysis of your data
##   Enter: interact()
##
##
## Attaching package: 'lessR'
##
## The following objects are masked from 'package:dplyr':
##
##     recode, rename

# Importando archivos csv
m1<- read_csv("202110-divvy-tripdata.csv")

## Rows: 631226 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.

m2<- read_csv("202111-divvy-tripdata.csv")

## Rows: 359978 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.

m3<- read_csv("202112-divvy-tripdata.csv")

## Rows: 247540 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.
m4<- read_csv("202201-divvy-tripdata.csv")

## Rows: 103770 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.
m5<- read_csv("202202-divvy-tripdata.csv")

## Rows: 115609 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.
m6<- read_csv("202203-divvy-tripdata.csv")

## Rows: 284042 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.
m7<- read_csv("202204-divvy-tripdata.csv")

## Rows: 371249 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.

```

```

m8<- read_csv("202205-divvy-tripdata.csv")

## Rows: 634858 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.

m9<- read_csv("202206-divvy-tripdata.csv")

## Rows: 769204 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.

m10<- read_csv("202207-divvy-tripdata.csv")

## Rows: 823488 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.

m11<- read_csv("202208-divvy-tripdata.csv")

## Rows: 785932 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.

m12<- read_csv("202209-divvy-tripdata.csv")

## Rows: 701339 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.
# Revisando nombres de columnas
colnames(m1)

## [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(m2)

## [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(m3)

## [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(m4)

## [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(m5)

## [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(m6)

## [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(m7)

## [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(m8)

## [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(m9)

## [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(m10)

## [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(m11)

## [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(m12)

## [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"

# Vista previa y estructura
glimpse(m1)

## #> #> Rows: 631,226
## #> Columns: 13
## #> $ ride_id          <chr> "620BC6107255BF4C", "4471C70731AB2E45", "26CA69D43D~
## #> $ rideable_type    <chr> "electric_bike", "electric_bike", "electric_bike", ~
## #> $ started_at       <dttm> 2021-10-22 12:46:42, 2021-10-21 09:12:37, 2021-10-~
## #> $ ended_at         <dttm> 2021-10-22 12:49:50, 2021-10-21 09:14:14, 2021-10-~
## #> $ start_station_name <chr> "Kingsbury St & Kinzie St", NA, NA, NA, NA, NA, NA, ~
## #> $ start_station_id   <chr> "KA1503000043", NA, NA, NA, NA, NA, NA, NA, ~
## #> $ end_station_name   <chr> NA, ~
## #> $ end_station_id     <chr> NA, ~
## #> $ start_lat          <dbl> 41.88919, 41.93000, 41.92000, 41.89000, 4~
## #> $ start_lng          <dbl> -87.63850, -87.70000, -87.70000, -87.69000, -87.710~
## #> $ end_lat            <dbl> 41.89000, 41.93000, 41.94000, 41.92000, 41.89000, 4~
## #> $ end_lng             <dbl> -87.63000, -87.71000, -87.72000, -87.69000, -87.690~
## #> $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```

glimpse(m2)

## Rows: 359,978
## Columns: 13
## $ ride_id      <chr> "7C00A93E10556E47", "90854840DFD508BA", "0A7D10CDD1~
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at    <dttm> 2021-11-27 13:27:38, 2021-11-27 13:38:25, 2021-11-~
## $ ended_at      <dttm> 2021-11-27 13:46:38, 2021-11-27 13:56:10, 2021-11-~
## $ start_station_name <chr> NA, NA, NA, NA, NA, "Michigan Ave & Oak St", NA, NA~
## $ start_station_id   <chr> NA, NA, NA, NA, NA, "13042", NA, NA, NA, NA, NA, NA~
## $ end_station_name  <chr> NA, ~
## $ end_station_id    <chr> NA, ~
## $ start_lat        <dbl> 41.93000, 41.96000, 41.96000, 41.94000, 41.90000, 4~
## $ start_lng         <dbl> -87.72000, -87.70000, -87.70000, -87.79000, -87.630~
## $ end_lat          <dbl> 41.96, 41.92, 41.96, 41.93, 41.88, 41.90, 41.80, 41~
## $ end_lng           <dbl> -87.73, -87.70, -87.70, -87.79, -87.62, -87.63, -87~
## $ member_casual    <chr> "casual", "casual", "casual", "casual", "casual", "~

glimpse(m3)

## Rows: 247,540
## Columns: 13
## $ ride_id      <chr> "46F8167220E4431F", "73A77762838B32FD", "4CF4245205~
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at    <dttm> 2021-12-07 15:06:07, 2021-12-11 03:43:29, 2021-12-~
## $ ended_at      <dttm> 2021-12-07 15:13:42, 2021-12-11 04:10:23, 2021-12-~
## $ start_station_name <chr> "Laflin St & Cullerton St", "LaSalle Dr & Huron St"~
## $ start_station_id   <chr> "13307", "KP1705001026", "KA1504000117", "KA1504000~
## $ end_station_name  <chr> "Morgan St & Polk St", "Clarendon Ave & Leland Ave"~
## $ end_station_id    <chr> "TA1307000130", "TA1307000119", "13137", "KP1705001~
## $ start_lat        <dbl> 41.85483, 41.89441, 41.89936, 41.89939, 41.89558, 4~
## $ start_lng         <dbl> -87.66366, -87.63233, -87.64852, -87.64854, -87.682~
## $ end_lat          <dbl> 41.87197, 41.96797, 41.93758, 41.89488, 41.93125, 4~
## $ end_lng           <dbl> -87.65097, -87.65000, -87.64410, -87.63233, -87.644~
## $ member_casual    <chr> "member", "casual", "member", "member", "member", "~

glimpse(m4)

## Rows: 103,770
## Columns: 13
## $ ride_id      <chr> "C2F7DD78E82EC875", "A6CF8980A652D272", "BD0F91DFF7~
## $ rideable_type <chr> "electric_bike", "electric_bike", "classic_bike", "~
## $ started_at    <dttm> 2022-01-13 11:59:47, 2022-01-10 08:41:56, 2022-01-~
## $ ended_at      <dttm> 2022-01-13 12:02:44, 2022-01-10 08:46:17, 2022-01-~
## $ start_station_name <chr> "Glenwood Ave & Touhy Ave", "Glenwood Ave & Touhy A~
## $ start_station_id   <chr> "525", "525", "TA1306000016", "KA1504000151", "TA13~
## $ end_station_name  <chr> "Clark St & Touhy Ave", "Clark St & Touhy Ave", "Gr~
## $ end_station_id    <chr> "RP-007", "RP-007", "TA1307000001", "TA1309000021", ~
## $ start_lat        <dbl> 42.01280, 42.01276, 41.92560, 41.98359, 41.87785, 4~
## $ start_lng         <dbl> -87.66591, -87.66597, -87.65371, -87.66915, -87.624~
## $ end_lat          <dbl> 42.01256, 42.01256, 41.92533, 41.96151, 41.88462, 4~
## $ end_lng           <dbl> -87.67437, -87.67437, -87.66580, -87.67139, -87.627~
## $ member_casual    <chr> "casual", "casual", "member", "casual", "member", "~

```

```

glimpse(m5)

## Rows: 115,609
## Columns: 13
## $ ride_id      <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B~  

## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl~  

## $ started_at    <dttm> 2022-02-19 18:08:41, 2022-02-20 17:41:30, 2022-02-~  

## $ ended_at      <dttm> 2022-02-19 18:23:56, 2022-02-20 17:45:56, 2022-02-~  

## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood ~  

## $ start_station_id   <chr> "TA1305000029", "TA1309000061", "TA1305000029", "13~  

## $ end_station_name  <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo~  

## $ end_station_id    <chr> "13179", "TA1307000113", "13011", "13323", "TA13070~  

## $ start_lat        <dbl> 41.88462, 41.92914, 41.88462, 41.94815, 41.88462, 4~  

## $ start_lng         <dbl> -87.62783, -87.64908, -87.62783, -87.66394, -87.627~  

## $ end_lat          <dbl> 41.91569, 41.92877, 41.87926, 41.95283, 41.88584, 4~  

## $ end_lng           <dbl> -87.63460, -87.66391, -87.63990, -87.64999, -87.635~  

## $ member_casual    <chr> "member", "member", "member", "member", "member", "~

glimpse(m6)

## Rows: 284,042
## Columns: 13
## $ ride_id      <chr> "47EC0A7F82E65D52", "8494861979B0F477", "EFE527AF80~  

## $ rideable_type <chr> "classic_bike", "electric_bike", "classic_bike", "c~  

## $ started_at    <dttm> 2022-03-21 13:45:01, 2022-03-16 09:37:16, 2022-03-~  

## $ ended_at      <dttm> 2022-03-21 13:51:18, 2022-03-16 09:43:34, 2022-03-~  

## $ start_station_name <chr> "Wabash Ave & Wacker Pl", "Michigan Ave & Oak St", ~  

## $ start_station_id   <chr> "TA1307000131", "13042", "13109", "TA1307000131", "~  

## $ end_station_name  <chr> "Kingsbury St & Kinzie St", "Orleans St & Chestnut ~  

## $ end_station_id    <chr> "KA1503000043", "620", "15578", "TA1305000025", "13~  

## $ start_lat        <dbl> 41.88688, 41.90100, 41.97835, 41.88688, 41.91172, 4~  

## $ start_lng         <dbl> -87.62603, -87.62375, -87.65975, -87.62603, -87.626~  

## $ end_lat          <dbl> 41.88918, 41.89820, 41.98404, 41.87771, 41.87794, 4~  

## $ end_lng           <dbl> -87.63851, -87.63754, -87.66027, -87.63532, -87.662~  

## $ member_casual    <chr> "member", "member", "member", "member", "member", "~

glimpse(m7)

## Rows: 371,249
## Columns: 13
## $ ride_id      <chr> "3564070EEFD12711", "0B820C7FCF22F489", "89EEEE3229~  

## $ rideable_type <chr> "electric_bike", "classic_bike", "classic_bike", "c~  

## $ started_at    <dttm> 2022-04-06 17:42:48, 2022-04-24 19:23:07, 2022-04-~  

## $ ended_at      <dttm> 2022-04-06 17:54:36, 2022-04-24 19:43:17, 2022-04-~  

## $ start_station_name <chr> "Paulina St & Howard St", "Wentworth Ave & Cermak R~  

## $ start_station_id   <chr> "515", "13075", "TA1307000121", "13075", "TA130700~  

## $ end_station_name  <chr> "University Library (NU)", "Green St & Madison St", ~  

## $ end_station_id    <chr> "605", "TA1307000120", "TA1307000120", "KA170600500~  

## $ start_lat        <dbl> 42.01913, 41.85308, 41.87184, 41.85308, 41.87181, 4~  

## $ start_lng         <dbl> -87.67353, -87.63193, -87.64664, -87.63193, -87.646~  

## $ end_lat          <dbl> 42.05294, 41.88189, 41.88189, 41.86749, 41.88224, 4~  

## $ end_lng           <dbl> -87.67345, -87.64879, -87.64879, -87.63219, -87.641~  

## $ member_casual    <chr> "member", "member", "casual", "member", "member", "~
```

```

glimpse(m8)

## Rows: 634,858
## Columns: 13
## $ ride_id      <chr> "EC2DE40644C6B0F4", "1C31AD03897EE385", "1542FBEC83~
## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl~
## $ started_at    <dttm> 2022-05-23 23:06:58, 2022-05-11 08:53:28, 2022-05~~
## $ ended_at      <dttm> 2022-05-23 23:40:19, 2022-05-11 09:31:22, 2022-05~~
## $ start_station_name <chr> "Wabash Ave & Grand Ave", "DuSable Lake Shore Dr & ~
## $ start_station_id   <chr> "TA1307000117", "13300", "TA1305000032", "TA1305000~
## $ end_station_name   <chr> "Halsted St & Roscoe St", "Field Blvd & South Water~
## $ end_station_id     <chr> "TA1309000025", "15534", "13221", "TA1305000030", "~
## $ start_lat        <dbl> 41.89147, 41.88096, 41.88224, 41.88224, 4~
## $ start_lng         <dbl> -87.62676, -87.61674, -87.64107, -87.64107, -87.641~
## $ end_lat          <dbl> 41.94367, 41.88635, 41.90765, 41.88458, 41.88578, 4~
## $ end_lng           <dbl> -87.64895, -87.61752, -87.67255, -87.63189, -87.651~
## $ member_casual    <chr> "member", "member", "member", "member", "member", "~

glimpse(m9)

## Rows: 769,204
## Columns: 13
## $ ride_id      <chr> "600CFD130D0FD2A4", "F5E6B5C1682C6464", "B6EB6D27BA~
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at    <dttm> 2022-06-30 17:27:53, 2022-06-30 18:39:52, 2022-06~~
## $ ended_at      <dttm> 2022-06-30 17:35:15, 2022-06-30 18:47:28, 2022-06~~
## $ start_station_name <chr> NA, ~
## $ start_station_id   <chr> NA, ~
## $ end_station_name   <chr> NA, ~
## $ end_station_id     <chr> NA, ~
## $ start_lat        <dbl> 41.89, 41.91, 41.91, 41.80, 41.91, 42.03, 41.91, 41~
## $ start_lng         <dbl> -87.62, -87.62, -87.65, -87.66, -87.63, -87.71, -87~
## $ end_lat          <dbl> 41.91, 41.93, 41.89, 41.80, 41.93, 42.06, 41.92, 41~
## $ end_lng           <dbl> -87.62, -87.63, -87.61, -87.65, -87.64, -87.73, -87~
## $ member_casual    <chr> "casual", "casual", "casual", "casual", "casual", "~

glimpse(m10)

## Rows: 823,488
## Columns: 13
## $ ride_id      <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258~
## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl~
## $ started_at    <dttm> 2022-07-05 08:12:47, 2022-07-26 12:53:38, 2022-07~~
## $ ended_at      <dttm> 2022-07-05 08:24:32, 2022-07-26 12:55:31, 2022-07~~
## $ start_station_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain ~
## $ start_station_id   <chr> "13224", "15541", "15541", "15541", "TA1307000117", ~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"~
## $ end_station_id     <chr> "KA1503000043", "623", "623", "TA1307000164", "TA13~
## $ start_lat        <dbl> 41.90707, 41.86962, 41.86962, 41.86962, 41.89147, 4~
## $ start_lng         <dbl> -87.66725, -87.62398, -87.62398, -87.62398, -87.626~
## $ end_lat          <dbl> 41.88918, 41.87277, 41.87277, 41.79526, 41.93625, 4~
## $ end_lng           <dbl> -87.63851, -87.62398, -87.62398, -87.59647, -87.652~
## $ member_casual    <chr> "member", "casual", "casual", "casual", "member", "~

```

```

glimpse(m11)

## Rows: 785,932
## Columns: 13
## $ ride_id      <chr> "550CF7EFEAEC0C618", "DAD198F405F9C5F5", "E6F2BC47B6~
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at    <dttm> 2022-08-07 21:34:15, 2022-08-08 14:39:21, 2022-08-~
## $ ended_at      <dttm> 2022-08-07 21:41:46, 2022-08-08 14:53:23, 2022-08-~
## $ start_station_name <chr> NA, ~
## $ start_station_id   <chr> NA, ~
## $ end_station_name  <chr> NA, ~
## $ end_station_id    <chr> NA, ~
## $ start_lat         <dbl> 41.93, 41.89, 41.97, 41.94, 41.85, 41.79, 41.89, 41-
## $ start_lng          <dbl> -87.69, -87.64, -87.69, -87.65, -87.65, -87.72, -87-
## $ end_lat            <dbl> 41.94, 41.92, 41.97, 41.97, 41.84, 41.82, 41.89, 41-
## $ end_lng             <dbl> -87.72, -87.64, -87.66, -87.69, -87.66, -87.69, -87-
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "~

glimpse(m12)

## Rows: 701,339
## Columns: 13
## $ ride_id      <chr> "5156990AC19CA285", "E12D4A16BF51C274", "A02B53CD7D~
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at    <dttm> 2022-09-01 08:36:22, 2022-09-01 17:11:29, 2022-09-~
## $ ended_at      <dttm> 2022-09-01 08:39:05, 2022-09-01 17:14:45, 2022-09-~
## $ start_station_name <chr> NA, ~
## $ start_station_id   <chr> NA, ~
## $ end_station_name  <chr> "California Ave & Milwaukee Ave", NA, NA, NA, NA, ~
## $ end_station_id    <chr> "13084", NA, ~
## $ start_lat         <dbl> 41.93000, 41.87000, 41.87000, 41.93000, 41.92000, 4-
## $ start_lng          <dbl> -87.69000, -87.62000, -87.62000, -87.69000, -87.730-
## $ end_lat            <dbl> 41.92269, 41.87000, 41.87000, 41.94000, 41.92000, 4-
## $ end_lng             <dbl> -87.69715, -87.62000, -87.62000, -87.67000, -87.730-
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "~

# Seleccionando columnas útiles y también para disminuir el impacto en la memoria RAM
m1 <- m1 %>% select(rideable_type, started_at, ended_at, member_casual)
m2 <- m2 %>% select(rideable_type, started_at, ended_at, member_casual)
m3 <- m3 %>% select(rideable_type, started_at, ended_at, member_casual)
m4 <- m4 %>% select(rideable_type, started_at, ended_at, member_casual)
m5 <- m5 %>% select(rideable_type, started_at, ended_at, member_casual)
m6 <- m6 %>% select(rideable_type, started_at, ended_at, member_casual)
m7 <- m7 %>% select(rideable_type, started_at, ended_at, member_casual)
m8 <- m8 %>% select(rideable_type, started_at, ended_at, member_casual)
m9 <- m9 %>% select(rideable_type, started_at, ended_at, member_casual)
m10 <- m10 %>% select(rideable_type, started_at, ended_at, member_casual)
m11 <- m11 %>% select(rideable_type, started_at, ended_at, member_casual)
m12 <- m12 %>% select(rideable_type, started_at, ended_at, member_casual)

# Activar cuando sea necesario:
# Guardando cambios en el data frame en una nueva tabla
# write_csv(total_trips_v2, "total_trips_v1.csv")
# Cargando datos desde archivo csv
# total_trips_v2 <- read.csv("total_trips_v1.csv")

```

Paso 4: Analizar

Antes de analizar los datos, se deben consolidar todos ellos en un data frame y revisar que tenga el formato correcto.

```
# Uniendo todos los data frame
total_trips_v2 <- bind_rows(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12)

# Revisando columna, estructura y resumen
colnames(total_trips_v2)

## [1] "rideable_type" "started_at"      "ended_at"       "member_casual"

glimpse(total_trips_v2)

## # Rows: 5,828,235
## # Columns: 4
## # $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", "elec~
## # $ started_at     <dttm> 2021-10-22 12:46:42, 2021-10-21 09:12:37, 2021-10-16 16~
## # $ ended_at       <dttm> 2021-10-22 12:49:50, 2021-10-21 09:14:14, 2021-10-16 16~
## # $ member_casual <chr> "member", "member", "member", "member", "member", "membe~
summary(total_trips_v2)

## # rideable_type      started_at
## # Length:5828235    Min.   :2021-10-01 00:00:09.00
## # Class :character  1st Qu.:2022-02-28 19:21:08.50
## # Mode  :character  Median :2022-06-08 06:41:28.00
## #                  Mean   :2022-05-06 21:39:18.18
## #                  3rd Qu.:2022-08-02 11:26:01.00
## #                  Max.   :2022-09-30 23:59:56.00
## # 
## # ended_at           member_casual
## # Min.   :2021-10-01 00:03:11.0  Length:5828235
## # 1st Qu.:2022-02-28 19:34:02.5  Class :character
## # Median :2022-06-08 06:55:07.0  Mode  :character
## # Mean   :2022-05-06 21:58:54.2
## # 3rd Qu.:2022-08-02 11:46:26.0
## # Max.   :2022-10-05 19:53:11.0
```

Realizando algunos cálculos preliminares:

```
# Mostrar, agrupar y ordenar de forma descendente los tipos de usuario en "member_casual".
total_trips_v2 %>%
  group_by(member_casual) %>%
  summarize(n = n()) %>%
  arrange(desc(n), .by_group=TRUE)

## # A tibble: 2 x 2
##   member_casual      n
##   <chr>            <int>
## 1 member             3426949
## 2 casual             2401286

# Mostrar, agrupar y ordenar de forma descendente los tipos de bicicleta en "rideable_type".
total_trips_v2 %>%
  group_by(rideable_type) %>%
  summarize(n = n()) %>%
  arrange(desc(n), .by_group=TRUE)
```

```

## # A tibble: 3 x 2
##   rideable_type     n
##   <chr>           <int>
## 1 electric_bike 2895244
## 2 classic_bike  2740516
## 3 docked_bike    192475

# Crear nuevas columnas a partir de "started_at" y "ended_at"
# Esto nos ayudará más adelante a calcular estadísticas por estos períodos, en el criterio
# de que el mismo día de desbloqueo de la bicicleta, es el mismo que el del bloqueo en otra estación

total_trips_v2$date <- as.Date(total_trips_v2$started_at)

# Usar esto si se quiere que el mes aparezca abreviado
total_trips_v2$month <- format(as.Date(total_trips_v2$date), "%b") # %m, número; %b, abreviado

total_trips_v2$day <- format(as.Date(total_trips_v2$date), "%d")
total_trips_v2$year <- format(as.Date(total_trips_v2$date), "%Y")
total_trips_v2$day_of_week <- format(as.Date(total_trips_v2$date), "%A")

# Calculando el tiempo de uso entre desbloqueo y bloqueo de la bicicleta: "ride_length"
total_trips_v2$ride_length <- difftime(total_trips_v2$ended_at, total_trips_v2$started_at
                                         , units = "mins")

# Calculando media, mediana, máximo y mínimo, desviación estándar y cuartiles de "ride_length"
mean(total_trips_v2$ride_length)

## Time difference of 19.60198 mins
median(total_trips_v2$ride_length)

## Time difference of 10.48333 mins
max(total_trips_v2$ride_length)

## Time difference of 40705.02 mins
min(total_trips_v2$ride_length)

## Time difference of -10353.35 mins
sd(total_trips_v2$ride_length)

## [1] 169.8883
quantile(total_trips_v2$ride_length)

## Time differences in mins
##      0%        25%        50%        75%        100%
## -10353.350000  5.933333  10.483333 18.850000  40705.016667

```

Problemas detectados:

- Hay mucho outliers por la derecha, hace que los valores de los cuartiles estén muy por debajo de los valores necesarios para el análisis.
- Hay diferencias de tiempo negativas, hay que filtrar el valor de 1 minuto hacia adelante.

```

# Filtrando tiempo de viaje conforme a restricciones establecidas, mínimo 1 min, máximo 60 min.
total_trips_v2 <- total_trips_v2 %>%

```

```

filter(ride_length >= 1) %>%
filter(ride_length <= 60)

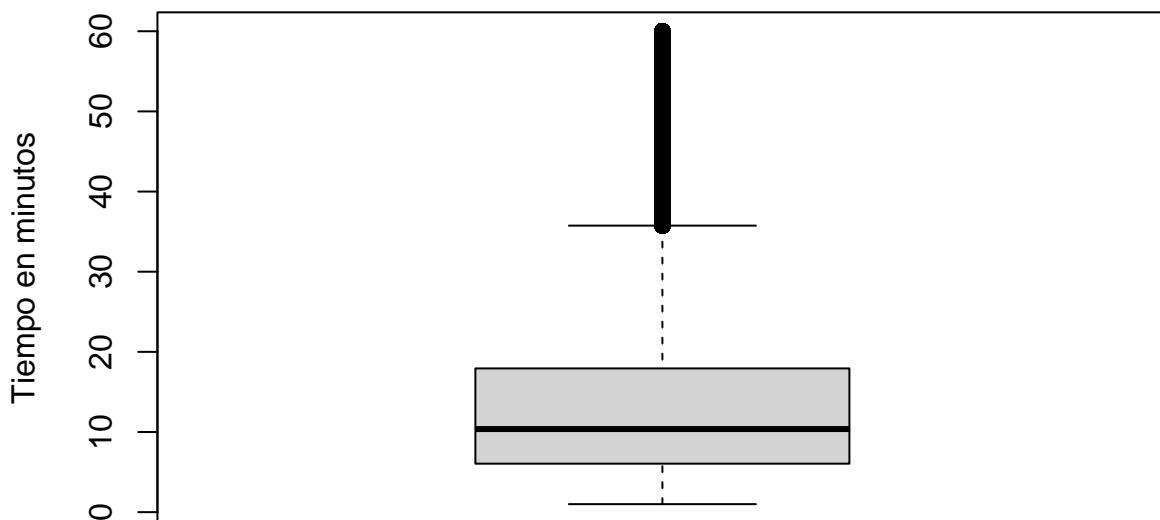
# Revisando cuartiles y boxplot
quantile(total_trips_v2$ride_length)

## Time differences in mins
##      0%      25%      50%      75%     100%
## 1.00000 6.05000 10.36667 17.93333 60.00000

# Boxplot 1
boxplot(total_trips_v2$ride_length
       , main = "Tiempo de viaje, con outliers"
       , ylab = "Tiempo en minutos")

```

Tiempo de viaje, con outliers



```

# Calculando la última observación antes de (Q3 + 1.5RI)
# Q1 = 8.18, Q3 = 20.22, IR = Q3 - Q1
Q1 <- 6.05
Q3 <- 17.93
IR <- Q3 - Q1
out <- Q3 + (1.5 * IR)
out

## [1] 35.75

# Filtrando outliers por encima de (Q3 + 1.5RI) = 35.75.
total_trips_v2 <- total_trips_v2 %>%
  filter(ride_length <= 35.75 )

```

```

# Revisando estadísticas
mean(total_trips_v2$ride_length)

## Time difference of 11.91215 mins
sd(total_trips_v2$ride_length)

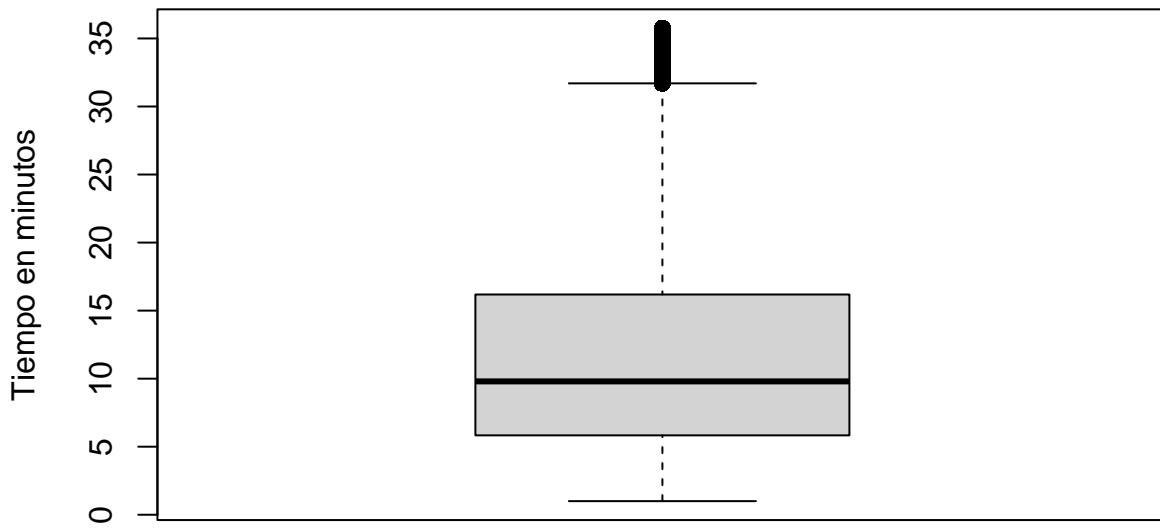
## [1] 7.849941
quantile(total_trips_v2$ride_length)

## Time differences in mins
##      0%      25%      50%      75%     100%
## 1.000000  5.833333  9.800000 16.183333 35.750000

# Boxplot 2
boxplot(total_trips_v2$ride_length
        , main = "Tiempo de viaje"
        , ylab = "Tiempo en minutos")

```

Tiempo de viaje



```

# Número de viajes y porcentaje por tipo de usuario y tipo de bicicleta ordenado
# descendente por número de viajes y grupos
total_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarize(n = n(), pct = n() * 100 / nrow(total_trips_v2)) %>%
  arrange(desc(n), .by_group=TRUE)

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

```

```

## # A tibble: 5 x 4
## # Groups:   member_casual [2]
##   member_casual rideable_type     n   pct
##   <chr>          <chr>      <int> <dbl>
## 1 casual         electric_bike 1118707 21.4
## 2 casual         classic_bike  769267 14.7
## 3 casual         docked_bike   116775  2.23
## 4 member        classic_bike  1690592 32.3
## 5 member        electric_bike 1533238 29.3

# Número de viajes y porcentaje por tipo de usuario y tipo de bicicleta ordenado
# descendente por número de viajes
total_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarize(n = n(), pct = n() * 100 / nrow(total_trips_v2)) %>%
  arrange(desc(n))

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

## # A tibble: 5 x 4
## # Groups:   member_casual [2]
##   member_casual rideable_type     n   pct
##   <chr>          <chr>      <int> <dbl>
## 1 member         classic_bike 1690592 32.3
## 2 member         electric_bike 1533238 29.3
## 3 casual         electric_bike 1118707 21.4
## 4 casual         classic_bike  769267 14.7
## 5 casual         docked_bike   116775  2.23

# Algunas medidas de tendencia central, comparando entre tipo de usuarios
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, FUN = mean)

##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                      casual           13.64153 mins
## 2                      member            10.83673 mins

aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, FUN = median)

##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                      casual           11.716667 mins
## 2                      member            8.716667 mins

aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, FUN = max)

##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                      casual            35.75 mins
## 2                      member            35.75 mins

aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, FUN = min)

##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                      casual             1 mins
## 2                      member             1 mins

# Promedio agrupado de "ride_length" agrupado por member_casual" y "ride_length"
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual
          + total_trips_v2$day_of_week, FUN = mean)

```

```

##      total_trips_v2$member_casual total_trips_v2$day_of_week
## 1                      casual                domingo
## 2                     member                domingo
## 3                      casual                jueves
## 4                     member                jueves
## 5                      casual                 lunes
## 6                     member                 lunes
## 7                      casual                martes
## 8                     member                martes
## 9                      casual            miércoles
## 10                    member            miércoles
## 11                      casual                sábado
## 12                     member                sábado
## 13                      casual               viernes
## 14                     member               viernes

##      total_trips_v2$ride_length
## 1          14.66198 mins
## 2          11.50916 mins
## 3          12.86256 mins
## 4          10.60281 mins
## 5          13.48651 mins
## 6          10.53172 mins
## 7          12.78764 mins
## 8          10.50219 mins
## 9          12.78481 mins
## 10         10.57512 mins
## 11         14.65987 mins
## 12         11.72014 mins
## 13         13.36838 mins
## 14         10.70879 mins

# Número de viajes por día de la semana
total_trips_v2 %>%
  group_by(day_of_week) %>%
  summarize(n = n()) %>%
  arrange(day_of_week)

## # A tibble: 7 x 2
##   day_of_week     n
##   <chr>       <int>
## 1 domingo     686242
## 2 jueves      768080
## 3 lunes        677601
## 4 martes      751561
## 5 miércoles   755912
## 6 sábado       824846
## 7 viernes     764337

# Reordenando la tabla en la columna "day_of_week" en secuencia de domingo a sábado.
total_trips_v2$day_of_week <- ordered(total_trips_v2$day_of_week
                                         , levels=c("domingo", "lunes", "martes"
                                         , "miércoles", "jueves", "viernes", "sábado"))

# Revisando cambio anterior con aggregate
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual

```

```

+ total_trips_v2$day_of_week, FUN = mean)

##   total_trips_v2$member_casual total_trips_v2$day_of_week
## 1               casual           domingo
## 2             member           domingo
## 3               casual            lunes
## 4             member            lunes
## 5               casual           martes
## 6             member           martes
## 7               casual        miércoles
## 8             member        miércoles
## 9               casual           jueves
## 10            member           jueves
## 11               casual           viernes
## 12             member           viernes
## 13               casual          sábado
## 14             member          sábado

##   total_trips_v2$ride_length
## 1           14.66198 mins
## 2           11.50916 mins
## 3           13.48651 mins
## 4           10.53172 mins
## 5           12.78764 mins
## 6           10.50219 mins
## 7           12.78481 mins
## 8           10.57512 mins
## 9           12.86256 mins
## 10          10.60281 mins
## 11          13.36838 mins
## 12          10.70879 mins
## 13          14.65987 mins
## 14          11.72014 mins

# Tabla con medidas de tendencia central y número de viajes (n) agrupada por
# "member_casual" y "day_of_week", ordenada por "day_of_week"
total_trips_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average=mean(ride_length), median=median(ride_length), min=min(ride_length),
            max=max(ride_length), number_of_rides = n()) %>%
  arrange (day_of_week)

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

## # A tibble: 14 x 7
## # Groups:   member_casual [2]
##   member_casual day_of_week average     median      min      max number~1
##   <chr>         <ord>       <drtn>      <drtn>     <drtn> <drtn>    <int>
## 1 casual        domingo    14.66198 mins 12.950000 mins 1 mins 35.75 ~ 323068
## 2 member        domingo    11.50916 mins  9.316667 mins 1 mins 35.75 ~ 363174
## 3 casual        lunes      13.48651 mins 11.450000 mins 1 mins 35.75 ~ 230930
## 4 member        lunes      10.53172 mins  8.366667 mins 1 mins 35.75 ~ 446671
## 5 casual        martes     12.78764 mins 10.783333 mins 1 mins 35.75 ~ 237778
## 6 member        martes     10.50219 mins  8.433333 mins 1 mins 35.75 ~ 513783
## 7 casual        miércoles 12.78481 mins 10.800000 mins 1 mins 35.75 ~ 245440

```

```

## 8 member      miércoles 10.57512 mins 8.533333 mins 1 mins 35.75 ~ 510472
## 9 casual      jueves    12.86256 mins 10.916667 mins 1 mins 35.75 ~ 265673
## 10 member     jueves    10.60281 mins 8.533333 mins 1 mins 35.75 ~ 502407
## 11 casual     viernes   13.36838 mins 11.433333 mins 1 mins 35.75 ~ 300698
## 12 member     viernes   10.70879 mins 8.616667 mins 1 mins 35.75 ~ 463639
## 13 casual     sábado    14.65987 mins 12.983333 mins 1 mins 35.75 ~ 401162
## 14 member     sábado    11.72014 mins 9.633333 mins 1 mins 35.75 ~ 423684
## # ... with abbreviated variable name 1: number_of_rides

# Tabla con medidas de tendencia central y número de viajes (n) agrupada por
# "member_casual" y "rideable_type", ordenada por "rideable_type"
total_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarize(average=mean(ride_length), median=median(ride_length), min=min(ride_length)
            , max=max(ride_length), number_of_rides = n()) %>%
  arrange (rideable_type)

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

## # A tibble: 5 x 7
## # Groups:   member_casual [2]
##   member_casual rideable_type average      median      min      max  number~1
##   <chr>          <chr>        <drtn>      <drtn>      <drtn> <drtn> <int>
## 1 casual         classic_bike 14.31105 mins 12.533333 mins 1 mins 35.75~ 769267
## 2 member         classic_bike 11.30452 mins 9.133333 mins 1 mins 35.75~ 1690592
## 3 casual         docked_bike 19.06834 mins 18.850000 mins 1 mins 35.75~ 116775
## 4 casual         electric_bike 12.61467 mins 10.633333 mins 1 mins 35.75~ 1118707
## 5 member         electric_bike 10.32092 mins 8.300000 mins 1 mins 35.75~ 1533238
## # ... with abbreviated variable name 1: number_of_rides

# Tabla con medidas de tendencia central y número de viajes (n) agrupada por
# "rideable_type" y "member_casual", ordenada por "member_casual"
total_trips_v2 %>%
  group_by(rideable_type, member_casual) %>%
  summarize(average=mean(ride_length), median=median(ride_length), min=min(ride_length)
            , max=max(ride_length), number_of_rides = n()) %>%
  arrange (member_casual)

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

## # A tibble: 5 x 7
## # Groups:   rideable_type [3]
##   rideable_type member_casual average      median      min      max  number~1
##   <chr>          <chr>        <drtn>      <drtn>      <drtn> <drtn> <int>
## 1 classic_bike  casual      14.31105 mins 12.533333 mins 1 mins 35.75~ 769267
## 2 docked_bike   casual      19.06834 mins 18.850000 mins 1 mins 35.75~ 116775
## 3 electric_bike casual      12.61467 mins 10.633333 mins 1 mins 35.75~ 1118707
## 4 classic_bike  member      11.30452 mins 9.133333 mins 1 mins 35.75~ 1690592
## 5 electric_bike member      10.32092 mins 8.300000 mins 1 mins 35.75~ 1533238
## # ... with abbreviated variable name 1: number_of_rides

```

Paso 5: Compartir

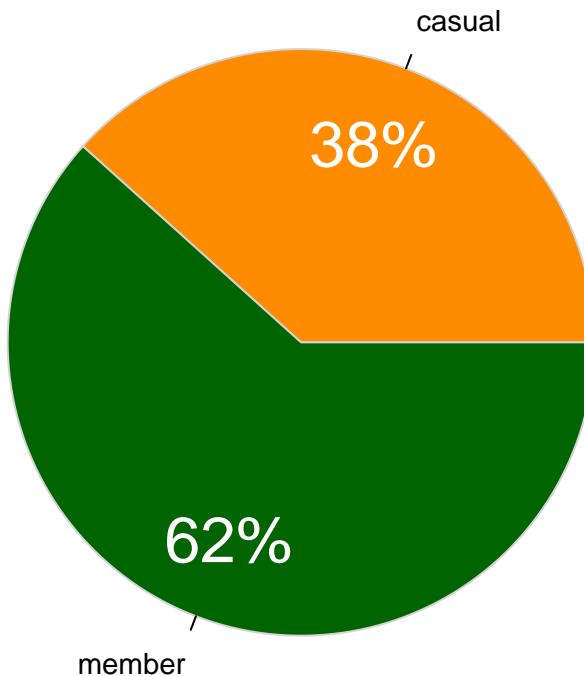
```
# Pie chart 1: número de viajes por tipo de usuario
table_users_type <- table(total_trips_v2$member_casual)
table_users_type

##
## casual member
## 2004749 3223830

PieChart(table_users_type, hole = 0, values = "%"
         , fill = c("darkorange", "darkgreen")
         , main = "% de viajes por tipo de usuario, oct 2021 - sept 2022", values_size = 2)

## >>> Note: table_users_type is not in a data frame (table)
## >>> Note: table_users_type is not in a data frame (table)
```

% de viajes por tipo de usuario, oct 2021 – sept 2022



```
## >>> suggestions
## piechart(table_users_type, hole=0) # traditional pie chart
## piechart(table_users_type, values="%") # display %'s on the chart
## piechart(table_users_type) # bar chart
## plot(table_users_type) # bubble plot
## plot(table_users_type, values="count") # lollipop plot
##
## --- table_users_type ---
##
##               casual    member      Total
## Frequencies:   2004749  3223830     5228579
```

```

## Proportions:      0.383      0.617      1.000
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 284237.550, df = 1, p-value = 0.000
# Convirtiendo "ride_length" de formato difftime a tipo numérico para hacer el histograma

class(total_trips_v2$ride_length)

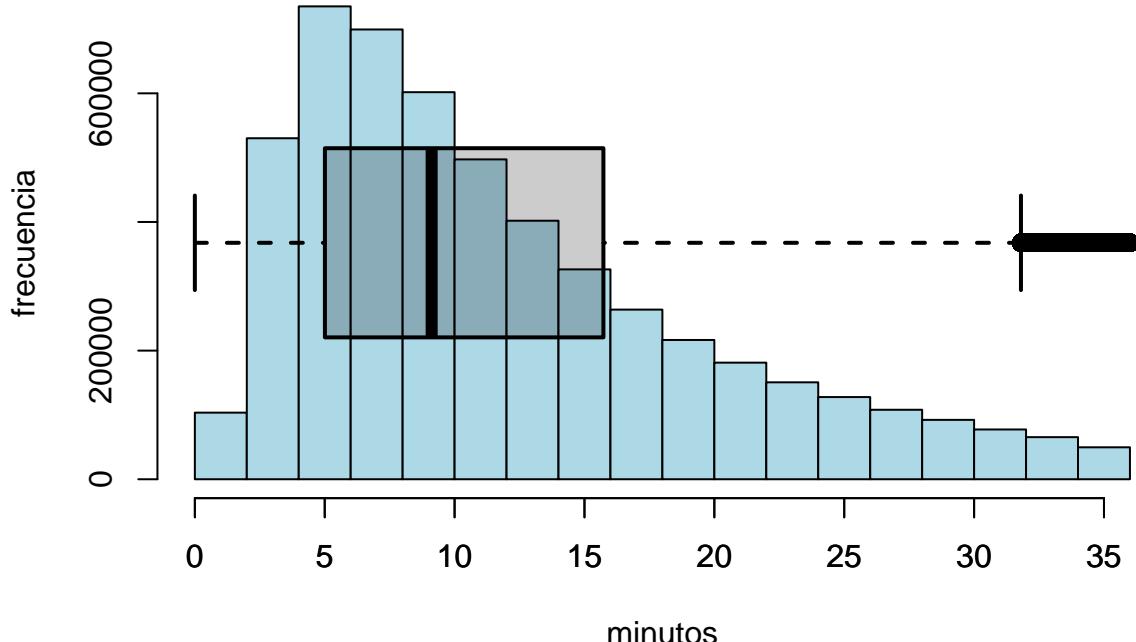
## [1] "difftime"

total_trips_v2$ride_length_num <- as.numeric(total_trips_v2$ride_length)
#total_trips_v2$ride_length_num

# Histograma + plotbox 1: variable "ride_length"
hist(x = total_trips_v2$ride_length_num
      , main = "Histograma de tiempo de viaje\noctubre 2021 - septiembre 2022"
      , xlab = "minutos"
      , ylab = "frecuencia", col = "lightblue")
axis(1) # añade eje x
par(new = TRUE)
boxplot(total_trips_v2$ride_length_num, horizontal = TRUE, axes = FALSE, lwd = 2
       , col = rgb(0, 0, 0, alpha = 0.2))

```

Histograma de tiempo de viaje octubre 2021 – septiembre 2022



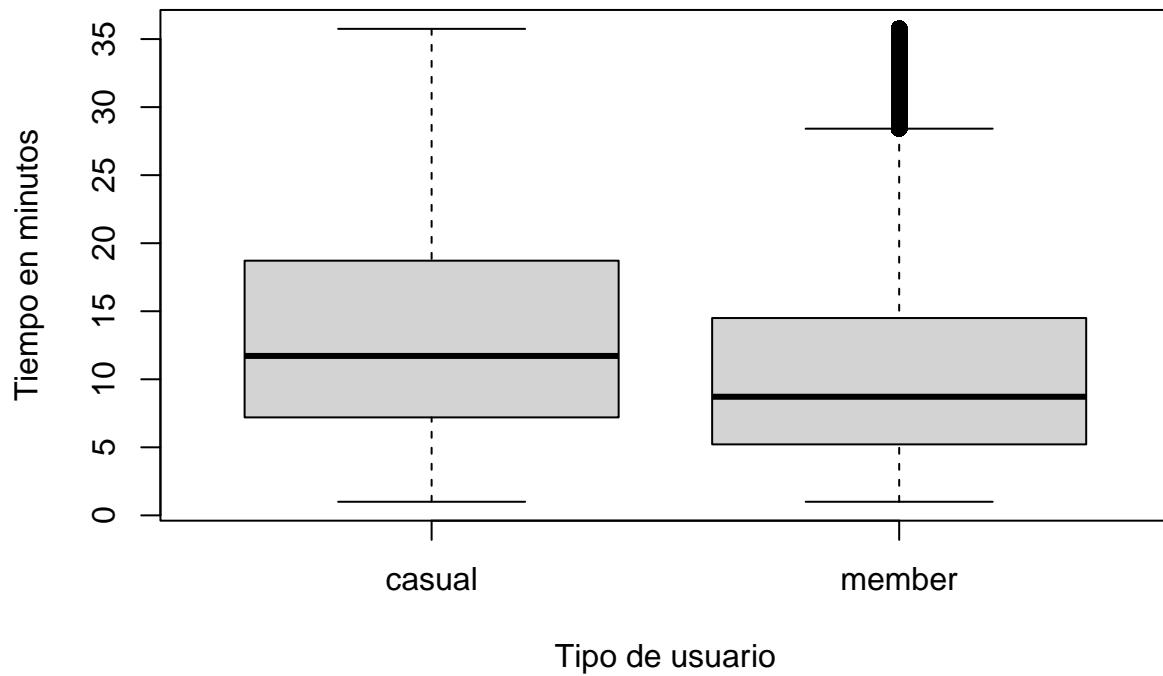
```

# Boxplot 3: variable "ride_length" por usuario, "member_casual"
boxplot(total_trips_v2$ride_length ~ total_trips_v2$member_casual
       , main = "Tiempo de viaje por tipo de usuario"
       , xlab = "Tipo de usuario"

```

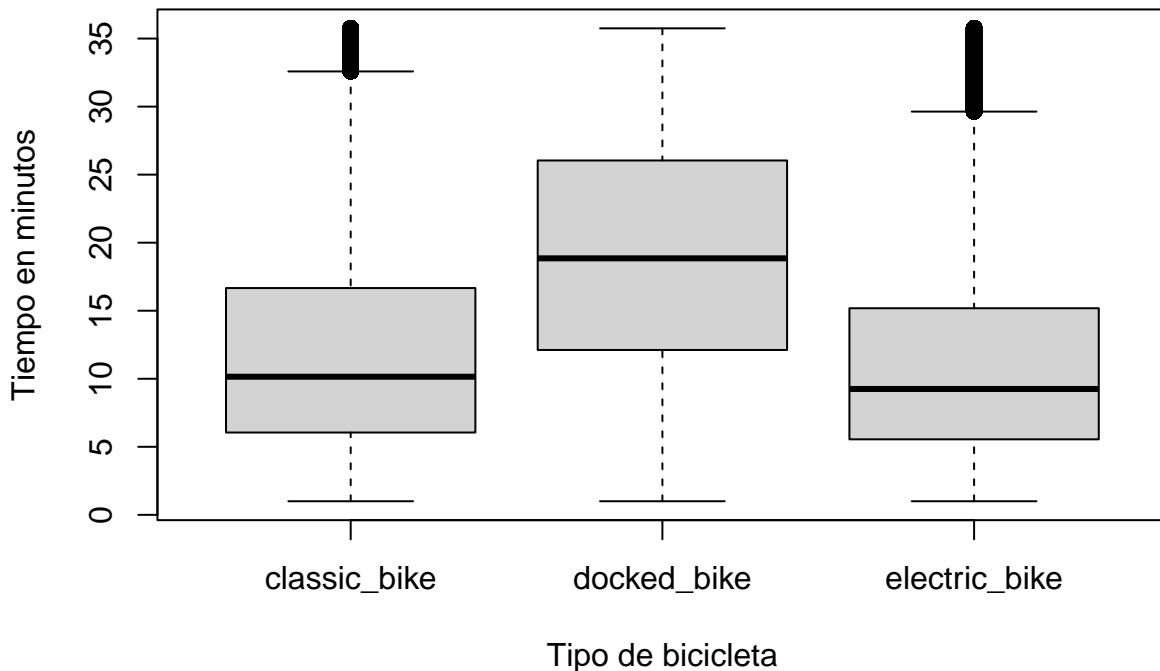
```
, ylab = "Tiempo en minutos")
```

Tiempo de viaje por tipo de usuario



```
# Boxplot 4: variable "ride_length" por tipo de bicicleta, "rideable_type"
boxplot(total_trips_v2$ride_length ~ total_trips_v2$rideable_type
       , main = "Tiempo de viaje por tipo de bicicleta"
       , xlab = "Tipo de bicicleta"
       , ylab = "Tiempo en minutos")
```

Tiempo de viaje por tipo de bicicleta



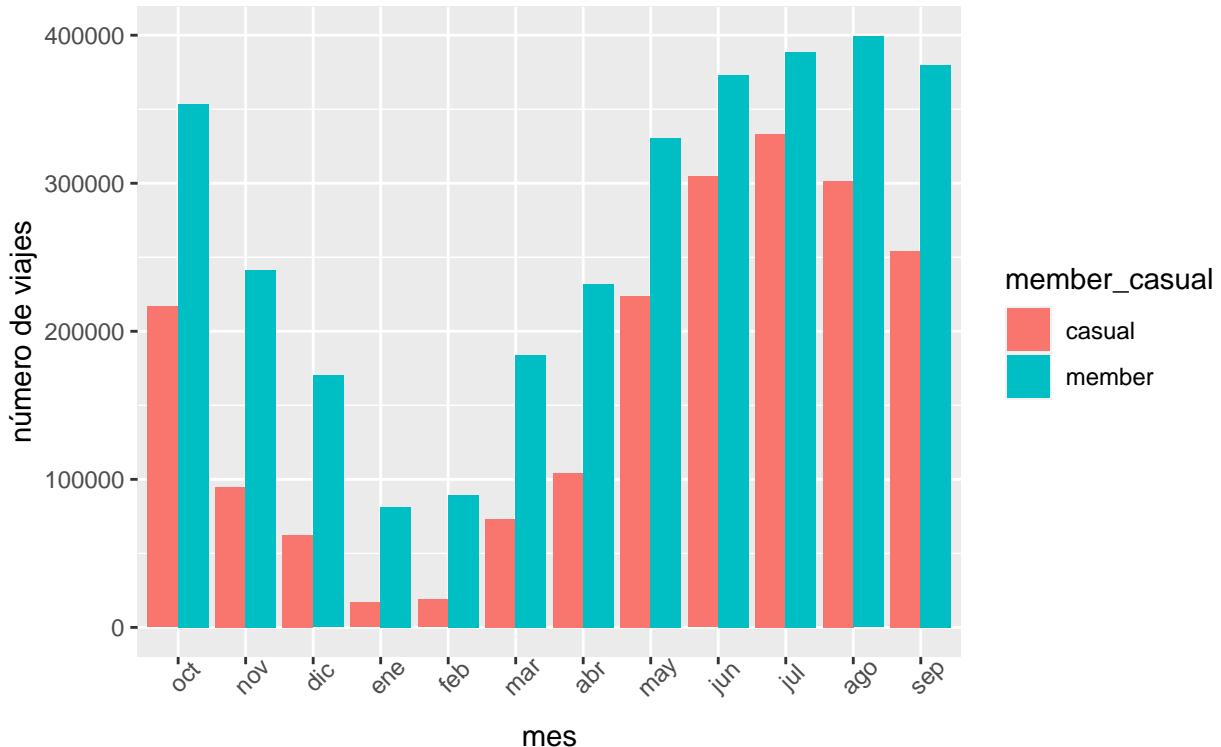
```
# Reordenando la tabla en la columna "day_of_week" en secuencia de domingo a sábado.
total_trips_v2$month <- ordered(total_trips_v2$month
                                , levels=c("oct", "nov", "dic", "ene", "feb", "mar"
                                          , "abr", "may", "jun", "jul", "ago", "sep"))

# Gráfico de barras 1: combinando "month" con "member_casual" del número de viajes
# Nota: los meses tienen formato abreviado. Los meses oct, nov y dic son del año 2021,
# los restantes corresponden al 2022

total_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()) %>%
  arrange(month, member_casual) %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  labs(title = "Gráfico mensual del número de viajes por tipo de usuario"
       , subtitle = "octubre 2021 - septiembre 2022"
       , x = "mes", y = "número de viajes" ) +
  geom_col(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45))

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

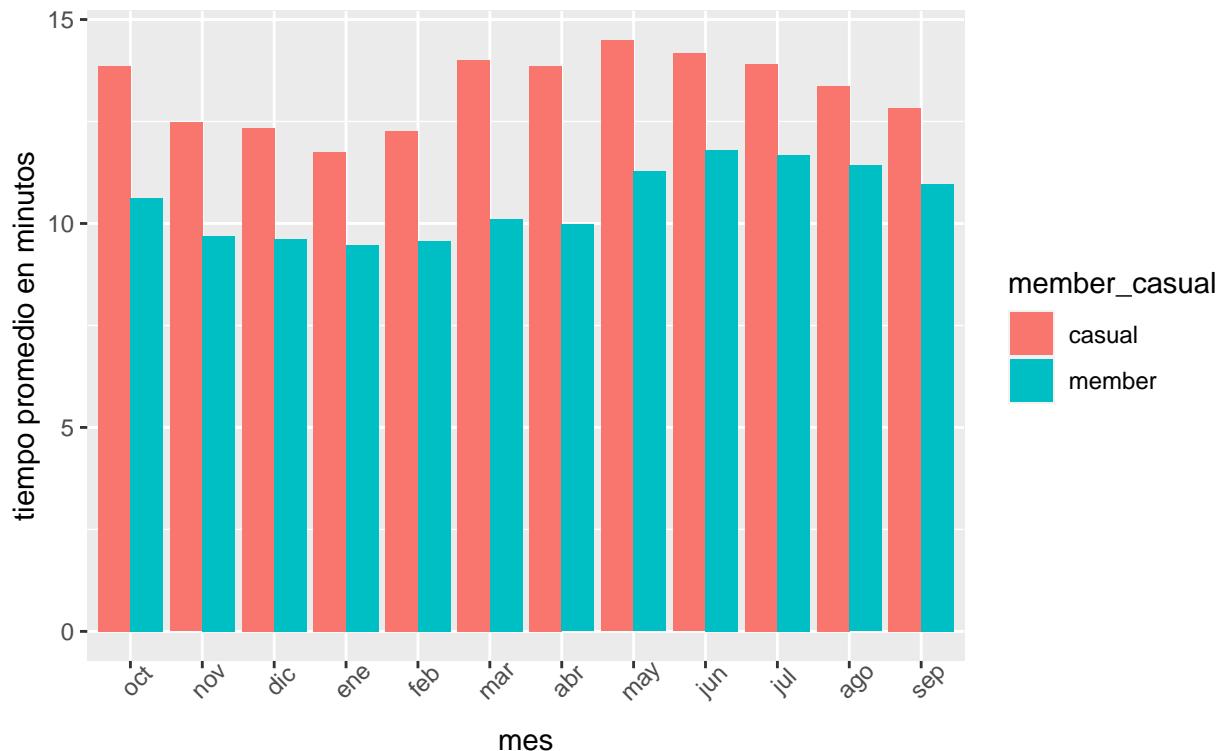
Gráfico mensual del número de viajes por tipo de usuario
octubre 2021 – septiembre 2022



```
# Gráfico de barras 2:
#Similar al gráfico anterior. En lugar del número de viajes, el eje de la "y" es para el
#promedio de duración de viaje. Gráfico de barras NO apiladas con variables agrupadas.
#Los meses de octubre a diciembre son del 2021, los restantes corresponden al 2022
total_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  labs(title = "Gráfico mensual del tiempo en promedio de viaje por tipo de usuario",
       subtitle = "octubre 2021 - septiembre 2022",
       x = "mes", y = "tiempo promedio en minutos" ) +
  geom_col(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45))

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

Gráfico mensual del tiempo en promedio de viaje por tipo de usuario
octubre 2021 – septiembre 2022



```
# Gráfico de barras 3: combinando "month" con "rideable_type" del número de viajes
# Nota: los meses tienen formato de carácter. Los meses 10 al 12 son del año 2021,
# los restantes corresponden al 2022
ggplot(data = total_trips_v2) +
  geom_bar(mapping = aes(x = month, fill = rideable_type)) +
  labs(title = "Gráfico mensual del número de viajes por tipo de usuario\ny tipo de bicicleta",
       subtitle = "octubre 2021 - septiembre 2022",
       x = "mes", y = "número de viajes") +
  facet_wrap(~member_casual) +
  theme(axis.text.x = element_text(angle = 45))
```

Gráfico mensual del número de viajes por tipo de usuario y tipo de bicicleta

octubre 2021 – septiembre 2022

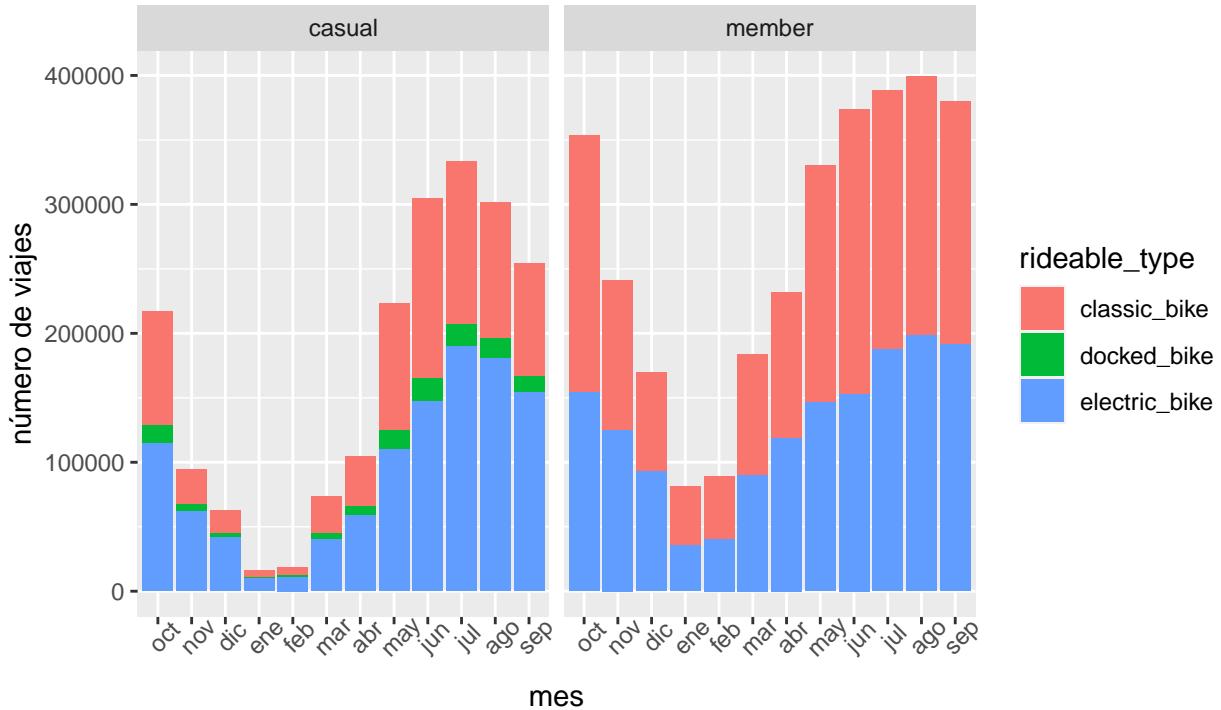
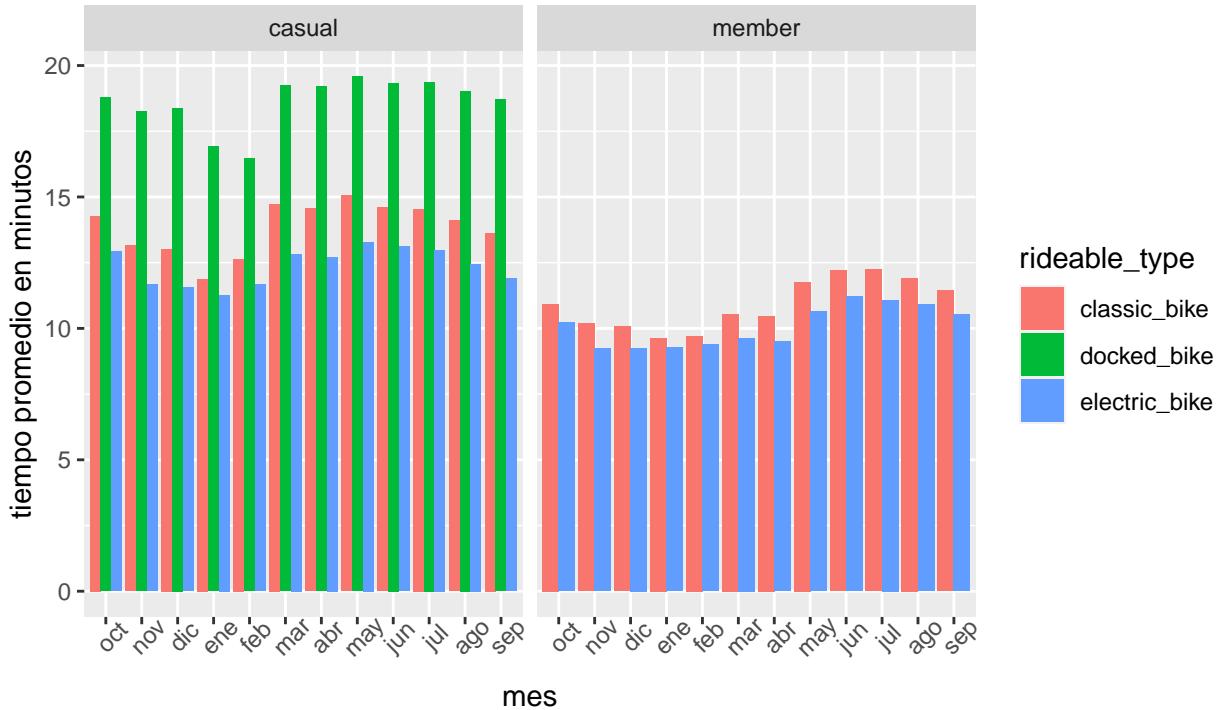


Gráfico del tiempo promedio de viaje mes,
tipo de bicicleta y tipo de usuario
octubre 2021 – septiembre 2022



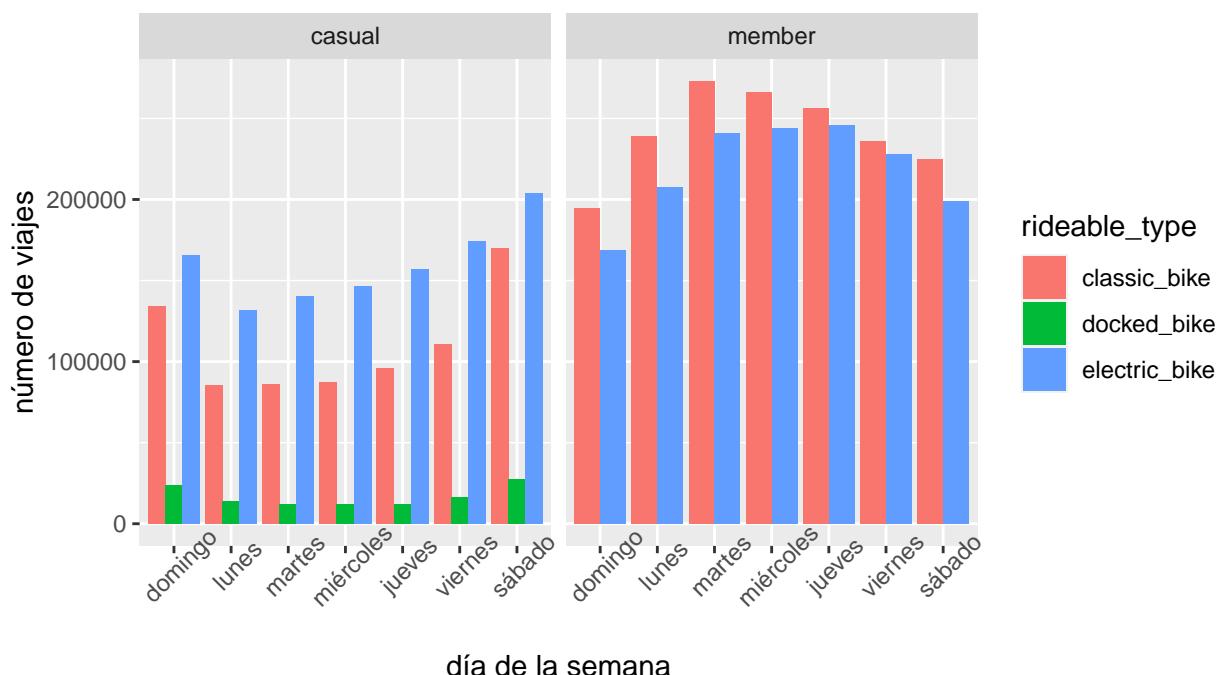
```
# Gráfico de barras 5
# Let's visualize the number of rides by rider type + facet(member_casual)
total_trips_v2 %>%
  group_by(member_casual, day_of_week, rideable_type) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = rideable_type)) +
  labs(title = "Gráfico del número de viajes por día de la semana,
  \ntipo de usuario y tipo de bicicleta"
  , subtitle = "octubre 2021 - septiembre 2022"
  , x = "día de la semana", y = "número de viajes" ) +
  facet_wrap(~member_casual) +
  theme(axis.text.x = element_text(angle = 45)) +
  geom_col(position = "dodge")

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

Gráfico del número de viajes por día de la semana,

tipo de usuario y tipo de bicicleta

octubre 2021 – septiembre 2022



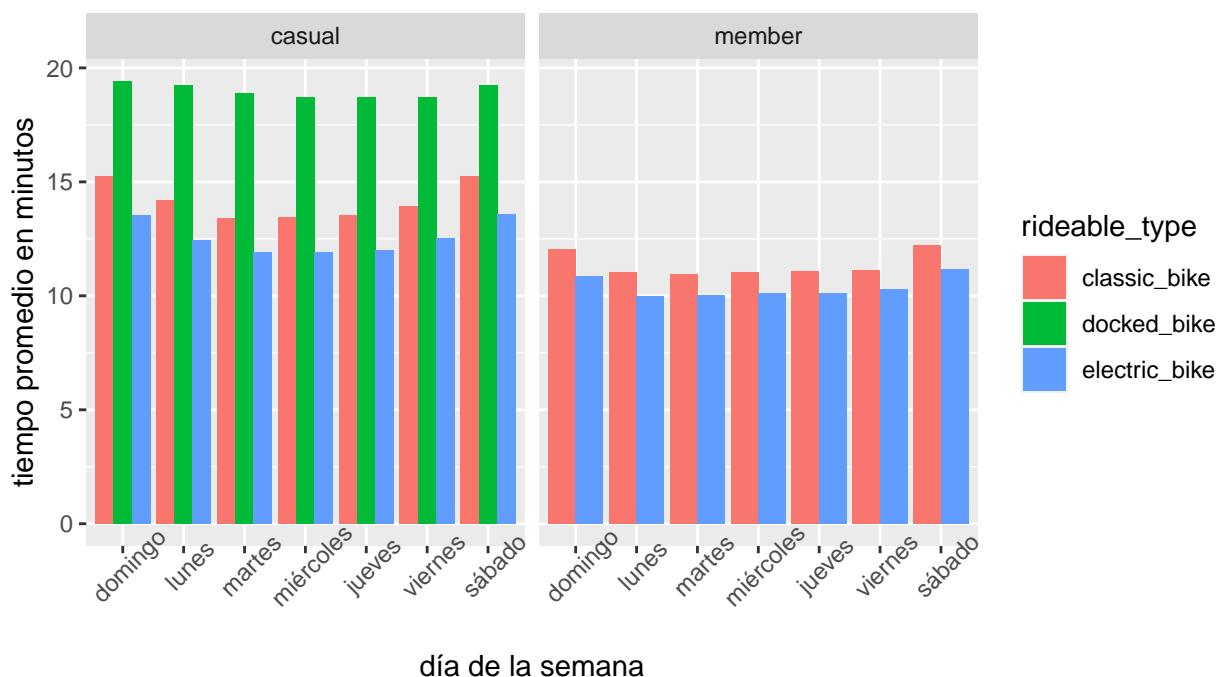
```
# Gráfico de barras 6
# Let's visualize the "average_duration" by rider type + facet(member_casual)
total_trips_v2 %>%
  group_by(member_casual, day_of_week, rideable_type) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(rideable_type, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = rideable_type)) +
  labs(title = "Tiempo promedio de viaje por día de la semana,
  \ny tipo de bicicleta y tipo de usuario"
  , subtitle = "octubre 2021 - septiembre 2022"
  , x = "día de la semana", y = "tiempo promedio en minutos" ) +
  facet_wrap(~member_casual) +
  theme(axis.text.x = element_text(angle = 45)) +
  geom_col(position = "dodge")

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

Tiempo promedio de viaje por día de la semana,

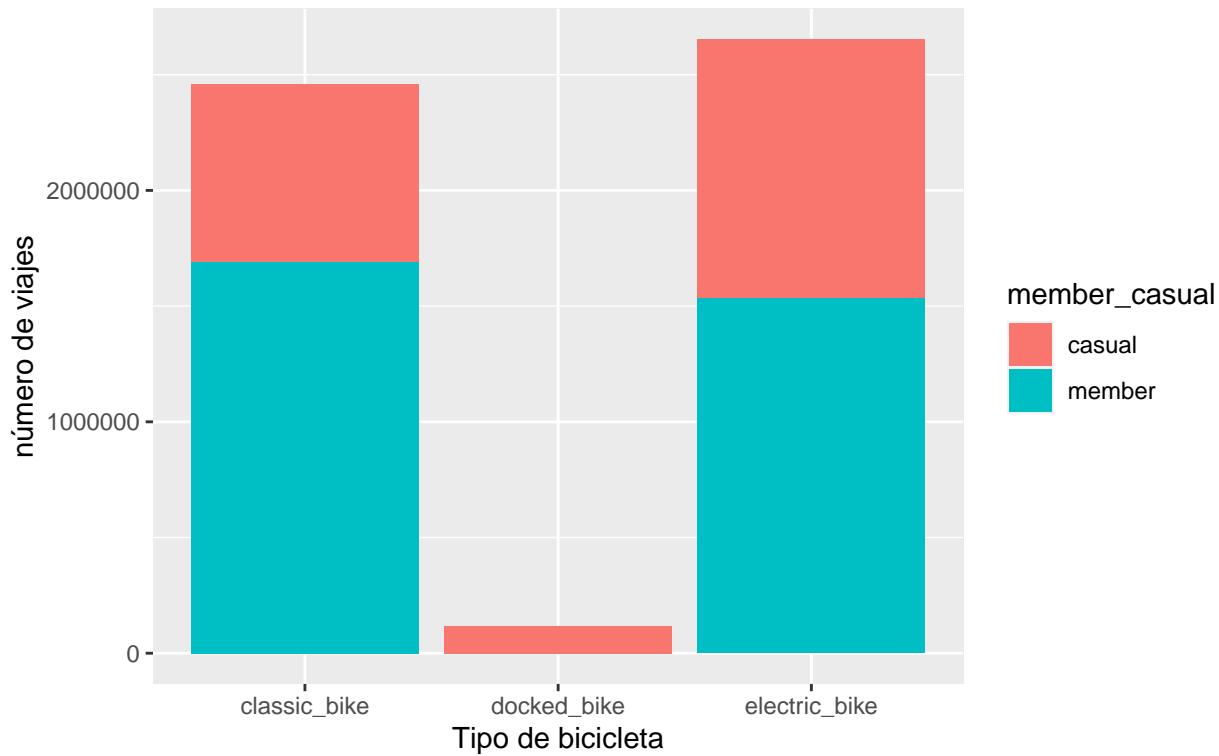
y tipo de bicicleta y tipo de usuario

octubre 2021 – septiembre 2022



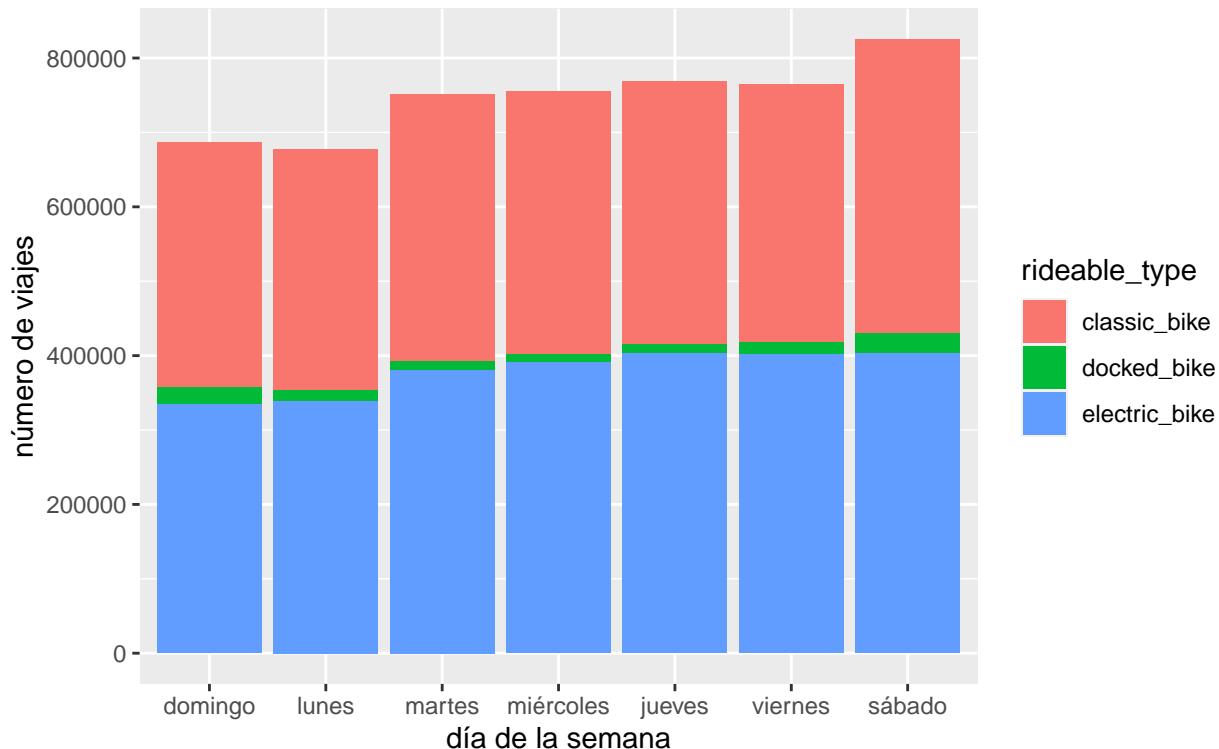
```
# Gráfico de barras 7: apilamiento combinando "rideable_type" con "member_casual"
# del número de viajes
ggplot(data = total_trips_v2) +
  geom_bar(mapping = aes(x = rideable_type, fill = member_casual)) +
  labs(title = "Gráfico del número de viajes por tipo de bicicleta y tipo de usuario",
       subtitle = "octubre 2021 - septiembre 2022",
       x = "Tipo de bicicleta", y = "número de viajes" )
```

Gráfico del número de viajes por tipo de bicicleta y tipo de usuario
octubre 2021 – septiembre 2022



```
# Gráfico de barras 8: apilamiento combinando "day_of_week" con "rideable_type" del número de viajes
ggplot(data = total_trips_v2) +
  geom_bar(mapping = aes(x = day_of_week, fill = rideable_type)) +
  labs(title = "Gráfico del número de viaje por día de la semana y tipo de bicicleta",
       subtitle = "octubre 2021 - septiembre 2022",
       x = "día de la semana", y = "número de viajes" )
```

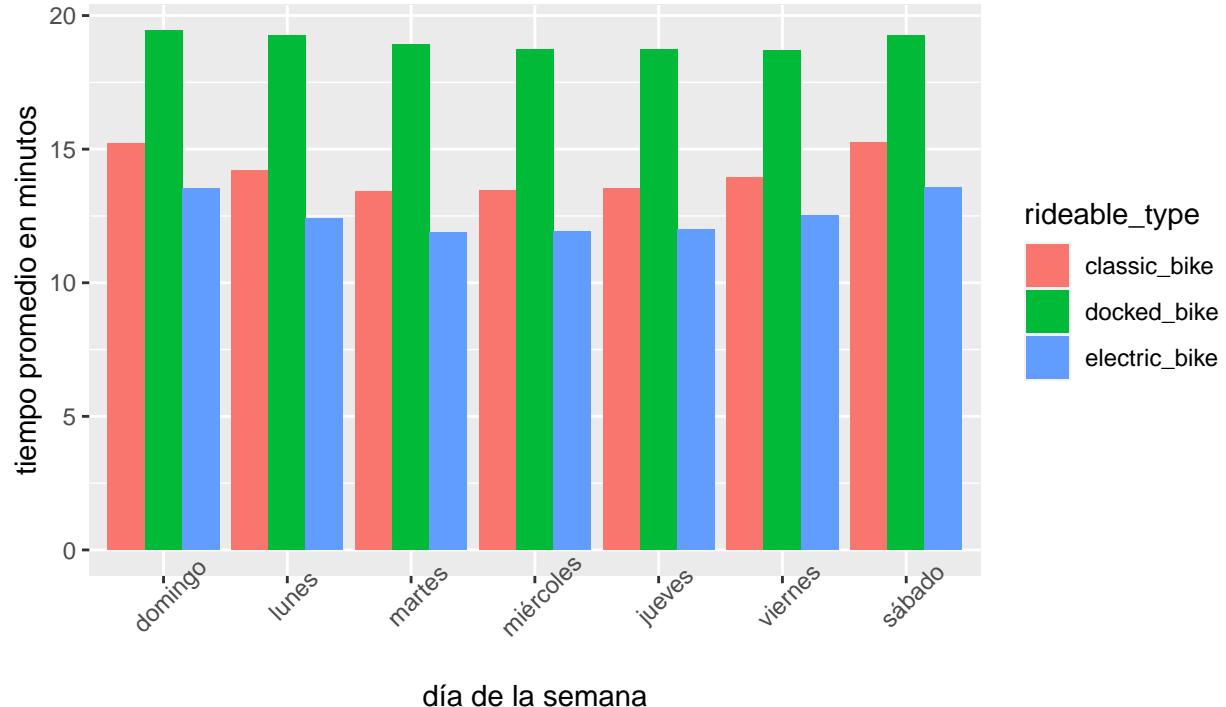
Gráfico del número de viaje por día de la semana y tipo de bicicleta
octubre 2021 – septiembre 2022



```
# Gráfico de barras 9: apilamiento combinando "day_of_week" con "rideable_type"
# del tiempo promedio de viaje
total_trips_v2 %>%
  group_by(member_casual, day_of_week, rideable_type) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(rideable_type, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = rideable_type)) +
  labs(title = "Gráfico del tiempo promedio de viaje por día de la semana,\nny tipo de bicicleta",
       subtitle = "octubre 2021 – septiembre 2022",
       x = "día de la semana", y = "tiempo promedio en minutos" ) +
  theme(axis.text.x = element_text(angle = 45)) +
  geom_col(position = "dodge")

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

Gráfico del tiempo promedio de viaje por día de la semana,
y tipo de bicicleta
octubre 2021 – septiembre 2022



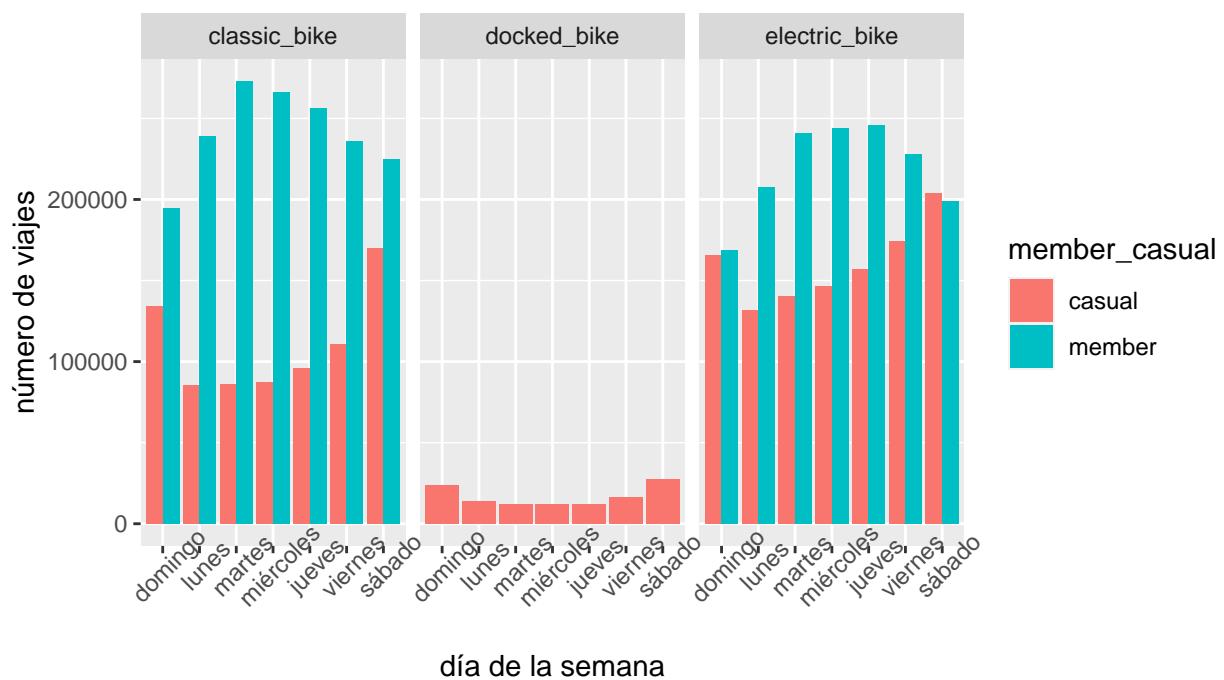
```
# Gráfico de barras 10
# Let's visualize the number of rides by rider type + facet(member_casual)
total_trips_v2 %>%
  group_by(member_casual, day_of_week, rideable_type) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  labs(title = "Gráfico del número de viajes por día de la semana,
  \ntipo de bicicleta y tipo de usuario"
  , subtitle = "octubre 2021 - septiembre 2022"
  , x = "día de la semana", y = "número de viajes" ) +
  facet_wrap(~rideable_type) +
  theme(axis.text.x = element_text(angle = 45)) +
  geom_col(position = "dodge")

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

Gráfico del número de viajes por día de la semana,

tipo de bicicleta y tipo de usuario

octubre 2021 – septiembre 2022



```
# Gráfico de barras 11
```

```
# Let's create a visualization for average duration
```

```
total_trips_v2 %>%
```

```
  group_by(member_casual, day_of_week, rideable_type) %>%
    summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
    arrange(member_casual, day_of_week) %>%
    ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
    labs(title = "Gráfico del tiempo promedio de viaje por día de la semana,
      \ntipo de bicicleta y tipo de usuario"
      , subtitle = "octubre 2021 - septiembre 2022"
      , x = "día de la semana", y = "tiempo promedio en minutos" ) +
    facet_wrap(~rideable_type) +
    theme(axis.text.x = element_text(angle = 45)) +
    geom_col(position = "dodge")
```

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

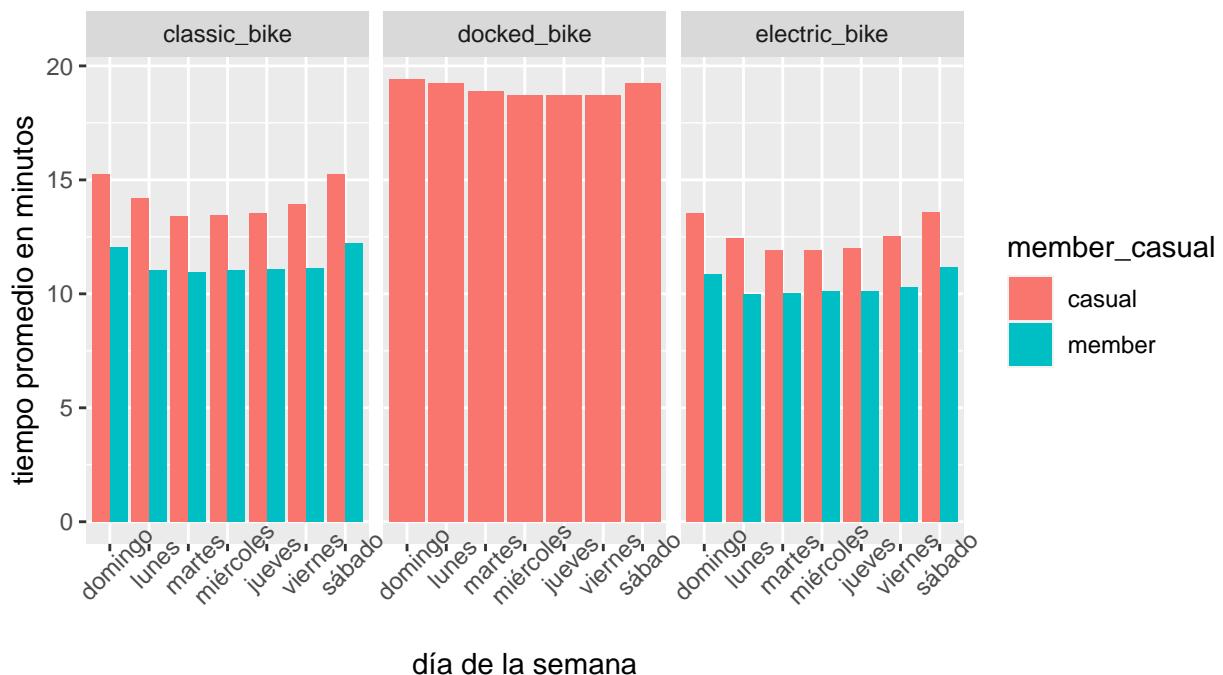
```
## Don't know how to automatically pick scale for object of type difftime.
```

```
## Defaulting to continuous.
```

Gráfico del tiempo promedio de viaje por día de la semana,

tipo de bicicleta y tipo de usuario

octubre 2021 – septiembre 2022



Paso 6: Actuar

Conclusiones.

1. En los últimos 12 meses, los miembros anuales hicieron más viajes (62% del total), en comparación con los ocasionales (38%), un 24% más.
2. El 75% de las veces, los tiempos de viajes duraron entre 1 y 15 minutos, aproximadamente.
3. A partir del minuto 10 de viaje, baja considerablemente el número de viajes con mayor duración.
4. Los días con más viajes: sábado, jueves y viernes.
5. Los días con menos viajes: lunes, miércoles y martes.
6. En promedio, el tiempo de viaje de los ocasionales es mayor que de los anuales, al menos 20% más: **lunes, 28% más; martes, 21.7% más; miércoles, 20.1% más; jueves, 21.3% más; viernes, 24.7% más; sábado 25.2% más; y domingo, 27.4% más.**
7. Sin embargo, los miembros anuales hacen más viajes que los ocasionales: **lunes, 93.4% más; martes, 116.0% más; miércoles, 107.0 más; jueves, 89.0% más; viernes, 54% más;** sábado 5.6% más; y domingo, 12.4% más.
8. En general, el día con más salidas es el sábado. Sábado, domingo y lunes son los días con mayor tiempo promedio de uso en el siguiente orden: docked, clásica y eléctrica.
9. Resumen del uso de bicicleta por tipo de usuario y tipo de bicicleta.

Tipo de usuario	Tipo de bicicleta	Tiempo prom. min	% de viajes
member	classic	11.30	32.3
member	electric	10.32	29.3
casual	electric	12.61	21.4
casual	classic	14.31	14.7

Tipo de usuario	Tipo de bicicleta	Tiempo prom. min	% de viajes
casual	docked	19.06	2.2

10. Los ocasionales tienen viajes entre 1 y 19 minutos de duración el 75% de las veces; los miembros anuales lo hacen hasta 13 minutos por viaje.
11. Hubo un gran descenso en el uso de las bicicletas de Cyclistic entre noviembre del 2021 y abril del 2022 por una ola de Covid-19 en el condado de Cook. Aún así se ve que los miembros anuales usan más las bicicletas que los ocasionales, todos los meses de estudio; aunque se observa un notorio descenso en agosto y septiembre del 2022 entre los ocasionales.
12. En cuanto a tiempo promedio de viaje, los ocasionales pasan un poco de más tiempo en las bicicletas clásicas que las eléctricas (11 y 14 minutos en promedio). Aunque las bicicletas clásicas las usan menos.
13. También, en cuanto a tiempo promedio de viaje, los miembros anuales ocupan en promedio entre 9 y 12 minutos las bicicletas clásicas que las eléctricas. Es de notar que manejan las bicicletas clásicas y las eléctricas en un porcentaje cercano al 30%.
14. Los miembros anuales no usan docked; los ocasionales muy poco, aunque el tiempo promedio (alrededor de 18 minutos) es mayor al viajar con clásicas y eléctricas.
15. Se nota un uso constante por promedio de tiempo de manejo en ambos tipos de miembros, tanto en la estadística mensual (salvo el periodo de cuarentena) y por día de la semana.
16. En general, las eléctricas son más usadas que las clásicas.
17. Las bicicletas clásicas son usadas, alrededor de 100% más, por los miembros anuales que los ocasionales. Aunque estos últimos lo usen 2 minutos más en promedio.
18. Las bicicletas eléctricas son usadas, alrededor de 25% más, por los miembros anuales que los ocasionales. Aunque estos últimos lo usen 2 minutos más en promedio.

Recomendaciones.

1. Armar una campaña para que los usuarios ocasionales se conviertan en miembros anuales en cuanto al uso de las bicicletas clásicas. Hay una área de oportunidad de incrementar el número de viajes en al menos 80% más en este tipo de bicicletas. En menor medida pero sin dejar de contemplar en la campaña a las eléctricas.
2. La campaña debe reflejar la necesidad de hacer el uso de la bicicleta entre semana, para ir al trabajo, por ejemplo. Reflejando los beneficios del ejercicio y otras relacionadas al tráfico, orientado a una población que esté dispuesta a pedalear entre 10 y 15 minutos, por ejemplo.
3. Hacer una campaña que premie al ciclista ocasional por el mayor tiempo de uso (distintos niveles de membresía por tiempo de manejo, por ejemplo).
4. Verificar si las bicicletas tipo docked pueden ser mantenidas, financieramente. Revisar si se trata de un nicho de mercado.