



Estrategia de Marketing Empresa Cyclistic

Rodrigo Bautista

2023-07-24

Objetivo

Estrategia de marketing de Cyclistic se basaba en la construcción de un reconocimiento de marca general y en atraer a amplios segmentos de consumidores. Diseñar estrategias de marketing orientadas a convertir a los ciclistas ocasionales en miembros anuales. Para hacer eso, se necesita entender mejor cómo difieren los miembros anuales y los ciclistas ocasionales, y el ¿por qué? los ciclistas ocasionales comprarían una membresía y cómo los medios digitales podrían afectar sus tácticas de marketing.

Contents

Preguntar	2
Instrucción de la tarea empresarial	2
Preparar	2
Descripción de las fuentes de los datos utilizadas	2
Procesar	3
Limpieza de los datos y su manipulación de los datos	3
Analizar y Visualización	5
Resumen del análisis	5
Actuar	29
Tres recomendaciones mas importantes	29

```
library(readr)
library(dplyr)
library(ggplot2)
library(data.table)
library(lubridate)
library(generics)
library(stringi)
library(descriptr)
library(scales)
```

¿En qué se diferencian los socios anuales y los ciclistas ocasionales con respecto al uso de las bicicletas de Cyclistic?

Preguntar

Instrucción de la tarea empresarial

- Tarea Empresarial
 1. *Problema a resolver.*
 - Maximizar el numero de miembros anuales en Cyclistic. Actualmente, Los miembros anuales son más rentables que los ciclistas ocasionales, por lo que es crucial convertir a los miembros ocasionales a miembros anuales y así impulsar el crecimiento y el éxito futuro de la empresa.
 2. *Decisiones Empresariales*
 - Con la ayuda del equipo de análisis de datos, y el conocimiento que tenemos, nos permitirá realizar un análisis exhaustivo del usos de las modalidades de las bicicletas de Cyclistic. Al comprender las diferencias en términos de **patrones de uso, duración de los viajes, frecuencia de uso, días con mayor demanda y de menor demanda** , y la **preferencia de los usuarios**, y así podremos identificar oportunidades claves para diseñar una estrategia de marketing efectiva que convierta los ciclistas ocasionales en miembros anuales.

Al terminar el análisis, podremos proporcionar recomendaciones basadas en el análisis, respaldadas por visualizaciones profesionales de los datos, podemos ayudar al equipo de Marketing (Dir. Lily Moreno) a tomar decisiones informadas y desarrollar tácticas de marketing específicas para atraer y retener a los ciclistas ocasionales, gracias a los análisis y recomendaciones, ya que los resultados de los análisis estarán respaldados por datos concretos y específicos para cada punto establecido, lo que aumentaran la confianza del equipo ejecutivo de Cyclistic para aprobar y llevar a cabo las estrategias recomendadas y así responder a:

1. ¿En qué se diferencian los socios anuales y los ciclistas ocasionales con respecto al uso de las bicicletas de Cyclistic?
2. ¿Por qué los ciclistas ocasionales comprarían membresías anuales de Cyclistic?
3. ¿Cómo puede usar Cyclistic los medios digitales para influenciar a los ciclistas ocasionales a convertirse en miembros?

Preparar

Descripción de las fuentes de los datos utilizadas

Para este análisis, los datos utilizados son datos históricos de los viajes realizados por meses, son proporcionados por la compañía Motivate. Los datos se encuentran en varios archivos que están especificados por meses, los cuales contiene los datos de manera ordenada. La fuente de los datos es confiable, ya que se proporciona por una empresa dedicada a guardar datos de esta empresa.

En términos de autorización, privacidad, seguridad y accesibilidad, se ha asegurado que los datos se utilicen de acuerdo con las políticas y regulaciones de protección de datos aplicables. No se utilizará información de identificación personal ni se realizará ningún intento de conectar los datos con información personal.

Para verificar la integridad de los datos, se pueden realizar diferentes acciones, como comprobar que los archivos descargados estén completos y sin errores, revisar la consistencia de los campos y valores, y comparar los datos con información adicional o registros previos para detectar discrepancias o anomalías.

El análisis de los datos históricos de los viajes de Cyclistic ayudará a identificar tendencias y patrones en el uso de las bicicletas. Esto permitirá comprender mejor los diferentes tipos de clientes y su comportamiento, lo que puede ser útil para tomar decisiones de negocio informadas y optimizar los servicios ofrecidos por Cyclistic. Los datos cuenta con varias celdas vacías, por lo tanto, representa un gran riesgo para tomar cualquier decisión basada en los datos por lo cual es necesario filtra los datos y ordenar de tal manera en disminuir los sesgos y los errores que pueda tener al momento de realizar el análisis de los datos.

Procesar

Liempieza de los datos y su manipulacion de los datos

Primeramente cargaremos los datos para poder visualizar los datos y comprobar si existe algún tipo de error los datos corresponde a los últimos 12 meses por lo cual se utilizaran los datos correspondiente a los meses que van desde Julio del 2022 a Junio del 2023. (se cargaran los datos)

```
#Se extraen todas los documentos y ase aguarden en un alista
list_datos <- list.files("E:\\Cyclistic", full.names= TRUE)
list_datos
```

```
## [1] "E:\\Cyclistic/202207-divvy-tripdata.csv"
## [2] "E:\\Cyclistic/202208-divvy-tripdata.csv"
## [3] "E:\\Cyclistic/202209-divvy-publictripdata.csv"
## [4] "E:\\Cyclistic/202210-divvy-tripdata.csv"
## [5] "E:\\Cyclistic/202211-divvy-tripdata.csv"
## [6] "E:\\Cyclistic/202212-divvy-tripdata.csv"
## [7] "E:\\Cyclistic/202301-divvy-tripdata.csv"
## [8] "E:\\Cyclistic/202302-divvy-tripdata.csv"
## [9] "E:\\Cyclistic/202303-divvy-tripdata.csv"
## [10] "E:\\Cyclistic/202304-divvy-tripdata.csv"
## [11] "E:\\Cyclistic/202305-divvy-tripdata.csv"
## [12] "E:\\Cyclistic/202306-divvy-tripdata.csv"
```

Para limpiar los datos es necesario utilizar varias funciones entre ellas **lapply** (permite aplicar una función sobre la lista para poder leer los archivos .csv) y el comando **read_csv** para poder visualizar los datos en un “marco de datos”, se le asignara una nueva columna llamada “month”, “diferencia de horas”, y los “días” y se aguardara en un nuevo archivo.

Se utiliza la función `stringr::str_replace` el cual nos ayudara a simplificar el nombre de la columna “month” y que solo se muestre la fecha del mes. Calculamos la diferencias de tiempo entre el tiempo de inicio del viaje y el final del viaje en bicicleta, posteriormente este dato que obtenemos se transformara a un dígito y se formateara de nuevo pero en formato de horas:minutos:segundos.

```
# Se almacenara el marco de datos procesados para cada archivo
resultados <- lapply(list_datos, function(x) {
  ##Leemos los dato de tipo .csv
  datos_limpios <- read_csv(x)
  ##Creamos una nueva columna con los meses
  datos_limpios$month <- x
  ## se modifica el texto en que se aparece el mes en este caso se elimina la
#la primera parte y la ultima
  datos_limpios$month <- stringr::str_replace( datos_limpios$month, pattern =
    'E:\\\\Cyclistic/',
    replacement = '')
  datos_limpios$month <- stringr::str_replace( datos_limpios$month, pattern =
    '-divvy-tripdata.csv',
    replacement = '')
  datos_limpios$month <- stringr::str_replace( datos_limpios$month, pattern =
    '-divvy-publictripdata.csv',
    replacement = '')
  ##obtenemos la diferencia y la almacenamos en otra columna
  datos_limpios$diff <- datos_limpios$ended_at - datos_limpios$started_at
  ## la diferencia obtenidas se formateara para tener un buen orden en los datos
  datos_limpios$diff_double <- as.double( datos_limpios$diff, units = "auto")
  datos_limpios <- datos_limpios[datos_limpios$diff_double >= 0, ]
})
```

```

datos_limpios$diff_format <- hms::hms(seconds_to_period( datos_limpios$diff))
#creamos otra columna con los días de la semana y posterior se formateara
# en tipo string
datos_limpios$day_of_week <- as.integer(format(datos_limpios$started_at, "%u"))

datos_limpios <- mutate(datos_limpios, dia_semana = case_when(
  day_of_week == 1 ~ "Lunes",
  day_of_week == 2 ~ "Martes",
  day_of_week == 3 ~ "Miércoles",
  day_of_week == 4 ~ "Jueves",
  day_of_week == 5 ~ "Viernes",
  day_of_week == 6 ~ "Sábado",
  day_of_week == 7 ~ "Domingo",
  TRUE ~ NA_character_
))
## se almacena en un data frame
datos_limpios
})

```

Se combinara todos los registros en un solo archivo “marco datos” verticalmente y se le asignara a la variable de datos_limpios, cambiaremos los nombres las columnas para tener un mejor manejo de los datos.

```

#los datos se almacenaran en un solo archivo
datos_limpios <- do.call(rbind, resultados)
# Eliminar filas con datos nulos
datos_limpios <- datos_limpios[complete.cases(datos_limpios), ]
# Actualizar los nombres de las columnas
colnames(datos_limpios) <- c("ride_id", "rideable_type", "started_at", "ended_at", "start_station_name"

```

Las columnas member_casual y rideable_type en factores, esta conversión nos ayudara para que estas columnas se traten de manera categórica y no por numéricamente, final mente para observar que los cambios han sido efectivos se muestra las columnas de los datos.

```

#Las columnas member_casual y rideable_type en factores
datos_limpios$member_casual <- as.factor(datos_limpios$member_casual)
datos_limpios$rideable_type <- as.factor(datos_limpios$rideable_type)

colnames(datos_limpios)

```

```

## [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"     "month"              "diff"
## [16] "diff_double"       "diff_format"        "day_of_week"
## [19] "dia_semana"

```

Analizar y Visualización

Resumen del análisis

Visualizamos los datos ya limpios filtraremos datos para tener una mejor precisión de los datos.

```
#eliminamos filas que contengan datos nulos
datos_limpios <- na.omit(datos_limpios)
#obtenemos un resumen de los datos
glimpse(datos_limpios)

## Rows: 4,409,013
## Columns: 19
## $ ride_id          <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258~
## $ rideable_type    <fct> classic_bike, classic_bike, classic_bike, classic_b~
## $ started_at       <dtm> 2022-07-05 08:12:47, 2022-07-26 12:53:38, 2022-07--
## $ ended_at         <dtm> 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     <fct> member, casual, casual, casual, member, member, mem~
## $ month             <chr> "202207", "202207", "202207", "202207", "202207", "~
## $ diff              <drtn> 705 secs, 113 secs, 463 secs, 3509 secs, 1578 secs~
## $ diff_double       <dbl> 705, 113, 463, 3509, 1578, 523, 689, 1853, 333, 687~
## $ diff_format       <time> 00:11:45, 00:01:53, 00:07:43, 00:58:29, 00:26:18, ~
## $ day_of_week       <int> 2, 2, 7, 7, 3, 5, 1, 4, 7, 7, 5, 6, 2, 3, 1, 7, 4, ~
## $ dia_semana        <chr> "Martes", "Martes", "Domingo", "Domingo", "Miércoles~
```

Una vez visualizados en el resumen, creamos una tabla para tener una mejor comprensión de los datos, en este caso se selecciono una parte de las columnas con mayor importancia para la decisiones del análisis.

```
#seleccionamos algunas columnas
datos_impo <- select( datos_limpios, month, start_station_name,
                      end_station_name, started_at, ended_at, member_casual,
                      diff_format, dia_semana)

# Generar la tabla con knitr::kable() para los datos seleccionados
tabla <- knitr::kable(head(datos_impo,10), col.names = c("Mes", "Estación inicial",
                                                         "Estación final", "Inicio T",
                                                         "Fin T", "Tipo", "Duration",
                                                         "Dia")) %>%
  kableExtra::kable_styling(full_width = TRUE, font_size = 8, position = "right")
```

tabla

Mes	Estación inicial	Estación final	inicioT	Fin T	Tipo	Duration	Dia
202207	Ashland Ave & Blackhawk St	Kingsbury St & Kinzie St	2022-07-05 08:12:47	2022-07-05 08:24:32	member	00:11:45	Martes
202207	Buckingham Fountain (Temp)	Michigan Ave & 8th St	2022-07-26 12:53:38	2022-07-26 12:55:31	casual	00:01:53	Martes
202207	Buckingham Fountain (Temp)	Michigan Ave & 8th St	2022-07-03 13:58:49	2022-07-03 14:06:32	casual	00:07:43	Domingo
202207	Buckingham Fountain (Temp)	Woodlawn Ave & 55th St	2022-07-31 17:44:21	2022-07-31 18:42:50	casual	00:58:29	Domingo
202207	Wabash Ave & Grand Ave	Sheffield Ave & Wellington Ave	2022-07-13 19:49:06	2022-07-13 20:15:24	member	00:26:18	Miércoles
202207	Desplaines St & Randolph St	Clinton St & Roosevelt Rd	2022-07-01 17:04:35	2022-07-01 17:13:18	member	00:08:43	Viernes
202207	Marquette Ave & 89th St	East End Ave & 87th St	2022-07-18 18:11:01	2022-07-18 18:22:30	member	00:11:29	Lunes
202207	Wabash Ave & Grand Ave	Dearborn Pkwy & Delaware Pl	2022-07-28 20:38:18	2022-07-28 21:09:11	casual	00:30:53	Jueves
202207	Wabash Ave & Grand Ave	Dearborn Pkwy & Delaware Pl	2022-07-10 22:55:59	2022-07-10 23:01:32	member	00:05:33	Domingo
202207	Ashland Ave & Blackhawk St	Orleans St & Merchandise Mart Plaza	2022-07-10 09:35:58	2022-07-10 09:47:25	member	00:11:27	Domingo

Filtraremos los datos para representar las diferentes tendencias en las distintas etapas de uso de las bicicletas. A continuación, se muestra un resumen de las variables que reflejan las diferencias de tiempo, es decir, los valores promedio del uso de las bicicletas sin importar el tipo de usuario y el día con mas demanda en todo el año, lo cual nos proporcionara las tendencias en todo el año.

```
# Calcular la media de diff_format
media_diff <- seconds_to_period(mean(datos_limpios$diff_double))

# Calcular el máximo de diff_format
max_diff <- seconds_to_period(max(datos_limpios$diff_double))

# Calcular la moda de dia_semana
moda_dia_semana <- names(table(datos_limpios$dia_semana))[which.max(table(datos_limpios$dia_semana))]

#Creamos la tablas de las tendencias
tendencias <- data.frame(Media = media_diff, Max = max_diff, Moda = moda_dia_semana)

tabla_tendencias <- tendencias %>%
  group_by(Moda) %>%
  summarize(Media = media_diff, Max = max_diff)

tabla <- tabla_tendencias %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE,
    position = "right"
```

)

tabla

Moda	Media	Max
Sábado	16M 3.11362180152571S	22d 5H 55M 27S

Calculamos el promedio de tiempo por cada tipo de usuario, en todo el año, esto nos servira para comprende el comportamiento de los ciclistas, y asi dar una estimacion si es rentable para los usuarios casuales.

```
#creamos los datos promedios de cada miembro
tabla_promedio_miembros <- datos_limpios %>%
  group_by(member_casual) %>%
  summarize(promedio_ride_length = mean(diff_double)) %>%
  mutate(promedio_ride_length = seconds_to_period(promedio_ride_length))
```

```
tablaPro <- tabla_promedio_miembros %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE,
    position = "right"
  )
```

tablaPro

member_casual	promedio_ride_length
casual	22M 29.7995819987318S
member	12M 6.07420301307536S

Crearemos una tabla en donde se puede ver el promedio de duración de viajes por días y filtrando por el tipo de usuario, lo cual servira para ver que dias son los que tiene un promedio mayor y asi poder hacer estrategias de marketing.

```
#tabla para el promedio de cada día en todo el año
tabla_promedio_por_dia <- datos_limpios %>%
  group_by(dia_semana, member_casual) %>%
  summarize(promedio_ride_length = mean(diff_double)) %>%
  mutate(promedio_ride_length = seconds_to_period(promedio_ride_length))
```

```
tablaPro_dia <- tabla_promedio_por_dia %>%
  kable() %>%
  kableExtra::kable_styling(full_width = TRUE)
```

tablaPro_dia

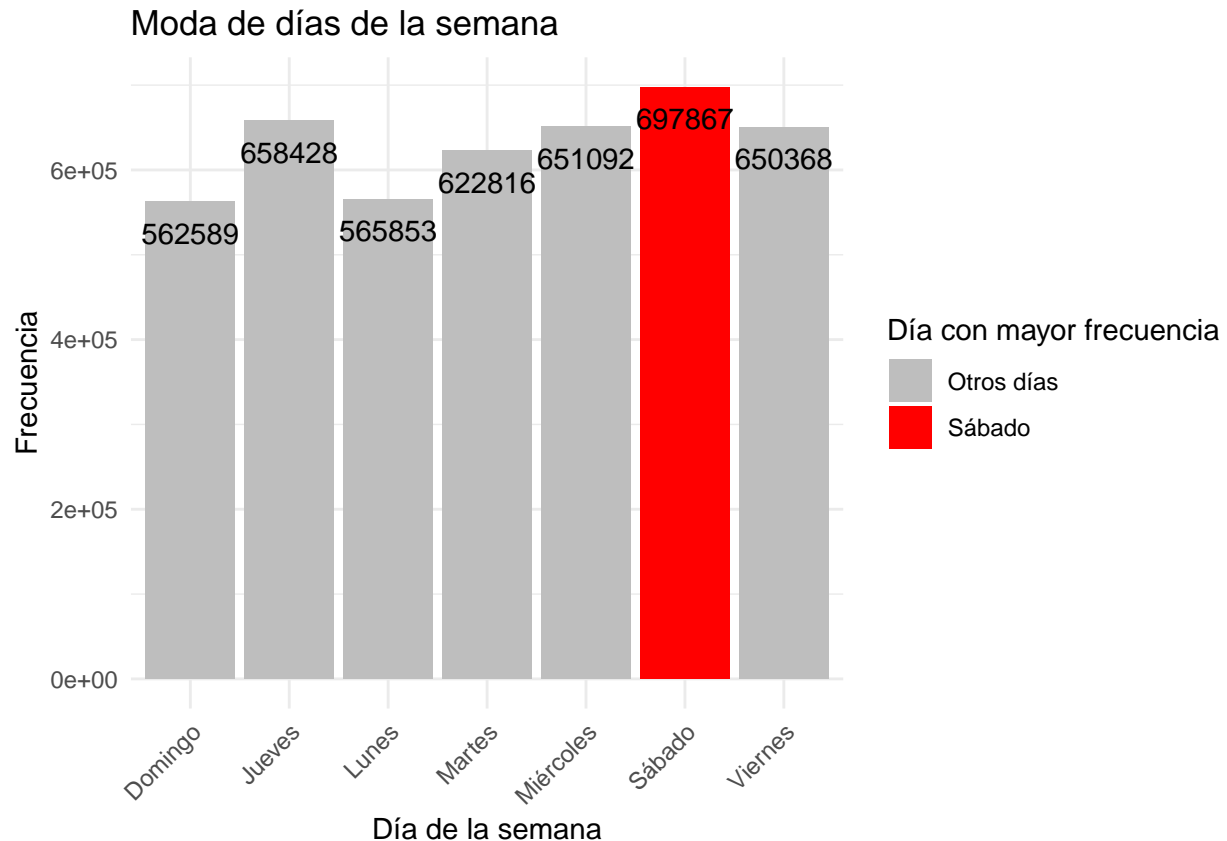
dia_semana	member_casual	promedio_ride_length
Domingo	casual	25M 45.7134072629692S
Domingo	member	13M 25.4612353502732S
Jueves	casual	19M 41.5756124407574S
Jueves	member	11M 37.7582597988919S
Lunes	casual	22M 29.9760611887004S
Lunes	member	11M 32.2094004042099S
Martes	casual	19M 57.6464438227065S
Martes	member	11M 31.9443383841652S
Miércoles	casual	19M 22.3083101154955S
Miércoles	member	11M 35.9165009895797S
Sábado	casual	25M 33.3288477279102S
Sábado	member	13M 42.1686884601365S
Viernes	casual	21M 42.0398903907383S
Viernes	member	11M 57.1177419151006S

Como pudimos apreciar en la tabla anterior, el día con mayor uso es el “Sábado”. Para comprender la diferencia con los demás días de la semana a lo largo del año, podemos observar la siguiente gráfica.

```
# Obtener el día con mayor frecuencia (moda)
data_moda <- as.data.frame(table(datos_limpios$dia_semana))
data_moda <- data_moda[order(-data_moda$Freq), ]
colnames(data_moda) <- c("dia_semana", "frecuencia")
dia_moda <- data_moda$dia_semana[1]

# Crear el gráfico de barras con resalte en el día de moda
grafico_moda <- ggplot(data_moda, aes(x = dia_semana, y = frecuencia,
                                       fill = dia_semana == moda_dia_semana)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = frecuencia), color = "black", vjust = 2)+
  labs(title = "Moda de días de la semana", x = "Día de la semana",
       y = "Frecuencia") +
  scale_fill_manual(values = c("gray", "red"),
                    labels = c("Otros días", moda_dia_semana),
                    guide = guide_legend(title = "Día con mayor frecuencia")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Mostrar el gráfico
grafico_moda
```

Si bien esto no nos proporciona una conclusión por sí solo, nos da una estimación del promedio de uso de las bicicletas sin importar el tipo de usuario por lo tanto. Calculamos la media de las diferencias de la duración del viaje, el máximo tiempo de recorrido, y calculamos la moda del día en que se utiliza las bicicletas por el tipo de usuario, en todo el año y filtramos los datos del tipo de bicicletas mas usadas por mes.

```
#####FUNCION
# Definir una función para convertir segundos a horas y minutos en el formato HH:MM
segundos_a_horas_y_minutos <- function(segundos) {
  horas <- floor(segundos / 3600)
  minutos <- floor((segundos % 3600) / 60)
  formato_hh_mm <- sprintf("%02d:%02d", horas, minutos)
  return(formato_hh_mm)
}

# Obtener el nombre completo del mes en el formato "YYYY-MM" a partir del número del mes
nombre_mes <- function(num_mes) {
  return(format(as.Date(paste0(num_mes, "01"), format = "%Y-%m-%d"), "%Y-%m"))
}

# Filtrar datos por tipo de usuario 'member' y calcular resumen
resumen_member <- datos_limpios %>%
  filter(member_casual == "member") %>%
  group_by(month) %>%
  summarise(
    media_diff_hh_mm = segundos_a_horas_y_minutos(mean(diff_double)),
    max_diff_hh_mm = segundos_a_horas_y_minutos(max(diff_double)),
    moda_dia_semana = names(table(dia_semana))[which.max(table(dia_semana))],
```

```

    total_viajes = n()
  ) %>%
  arrange(desc(total_viajes)) # Ordenar de mayor a menor según el total de viajes

# Filtrar datos por tipo de usuario 'casual' y calcular resumen
resumen_casual <- datos_limpios %>%
  filter(member_casual == "casual") %>%
  group_by(month) %>%
  summarise(
    media_diff_hh_mm = segundos_a_horas_y_minutos(mean(diff_double)),
    max_diff_hh_mm = segundos_a_horas_y_minutos(max(diff_double)),
    moda_dia_semana = names(table(dia_semana))[which.max(table(dia_semana))],
    total_viajes = n()
  ) %>%
  arrange(desc(total_viajes)) # Ordenar de mayor a menor según el total de viajes

tabla_casual <- resumen_casual %>%
  group_by(month) %>%
  summarize(moda_dia_semana, media_diff_hh_mm, max_diff_hh_mm, total_viajes )

tablaCasu <- tabla_casual %>%
  arrange(desc(total_viajes)) %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )

```

tablaCasu

month	moda_dia_semana	media_diff_hh_mm	max_diff_hh_mm	total_viajes
202207	Sábado	00:25	533:55	311670
202208	Sábado	00:23	80:48	270089
202209	Sábado	00:21	24:54	220913
202306	Sábado	00:24	185:52	219793
202305	Domingo	00:24	202:16	177037
202210	Sábado	00:20	137:23	151324
202304	Sábado	00:22	39:09	110537
202211	Jueves	00:17	24:45	73536
202303	Miércoles	00:16	24:56	46792
202302	Domingo	00:17	24:22	32776
202212	Jueves	00:14	24:35	31505
202301	Martes	00:14	24:03	29621

Como podemos observar en las tablas creadas se puede apreciar que en el caso de usuario casual tiene que el mes (202207) con mas viajes con una media de duración de 00:25 y el uso máximo es de 533:55 y el día que mas se usa es Martes con un total de viajes entoldo el año es de 311670

```

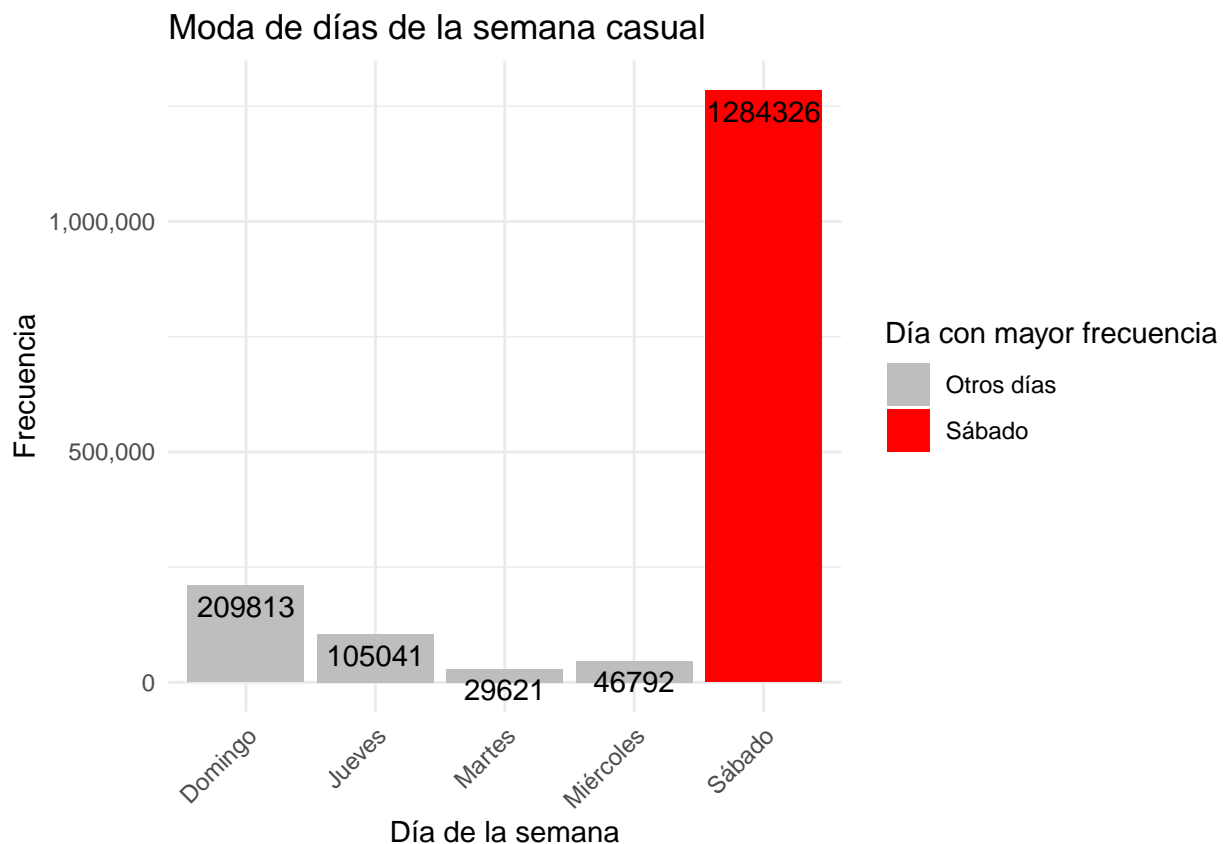
##Grafico para el tipo casual resumen de la tabla
datos_suma_por_dias <- resumen_casual %>%
  group_by(modad_dia_semana) %>%
  summarize(total_viajes = sum(total_viajes), .groups = "drop")

# Obtener el día con mayor frecuencia (moda)
moda_dia_semana_casual <- datos_suma_por_dias$moda_dia_semana[which.max(datos_suma_por_dias$total_viajes)]
moda_dia_semanas <- moda_dia_semana_casual

```

```
# Crear el gráfico de barras
grafico_moda_casual <- ggplot(datos_suma_por_dias, aes(x = moda_dia_semana, y = total_viajes, fill = moda_dia_semana)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = total_viajes), color = "black", vjust = 1.5) + # Ajustar la posición vertical
  labs(title = "Moda de días de la semana casual", x = "Día de la semana", y = "Frecuencia") +
  scale_fill_manual(values = c("gray", "red"),
                    labels = c("Otros días", "Sábado"),
                    guide = guide_legend(title = "Día con mayor frecuencia")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = scales::comma)

# Mostrar el gráfico
grafico_moda_casual
```



En este gráfico se puede observar que el tipo de usuario casual lo ocupa el servicio de bicicletas los días sábados y los domingos con una frecuencia de 1,284,326 en todo el año.

```
# Filtrar datos por tipo de usuario 'casual' y calcular resumen
tabla_mem <- resumen_member %>%
  group_by(month) %>%
  summarize(moda_dia_semana, media_diff_hh_mm, max_diff_hh_mm, total_viajes)

tablaMem <- tabla_mem %>%
  arrange(desc(total_viajes)) %>%
  kable() %>%
  kableExtra::kable_styling(
```

```

    full_width = TRUE
  )

```

tablaMem

month	moda_dia_semana	media_diff_hh_mm	max_diff_hh_mm	total_viajes
202208	Martes	00:13	24:05	335224
202207	Sábado	00:13	24:36	330996
202306	Jueves	00:12	24:05	314962
202209	Jueves	00:12	24:31	314227
202305	Martes	00:12	22:33	286185
202210	Lunes	00:11	24:16	262943
202304	Jueves	00:11	23:53	213657
202211	Miércoles	00:10	24:53	182226
202303	Miércoles	00:10	24:40	153655
202301	Martes	00:10	24:57	118663
202302	Martes	00:10	24:28	116784
202212	Jueves	00:10	23:26	103898

En el caso de usuario con la membresia tiene el mes (202208) con mas viajes con una media de duración de 00:13 y el uso máximo es de 24:05 y el día que mas se usa es Martes con un total de viajes en todo el año es de 335224

```

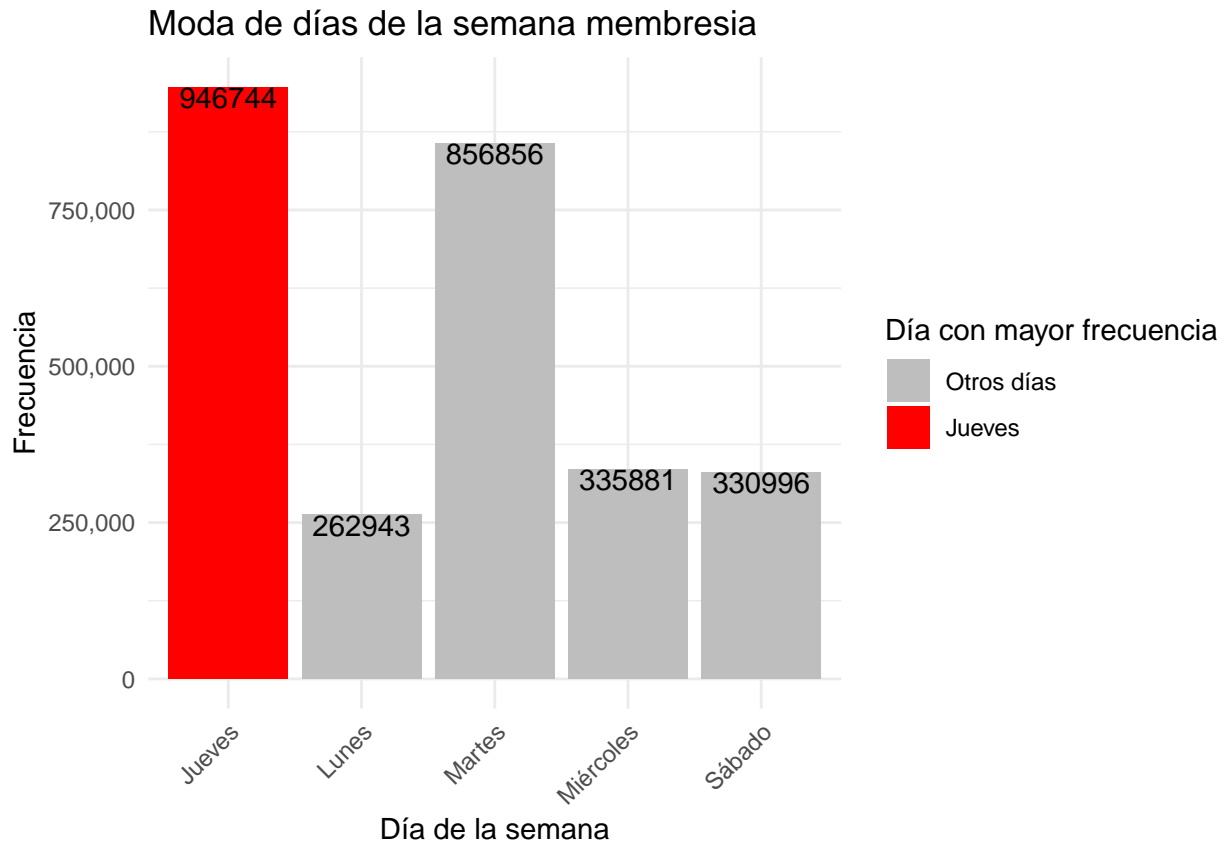
datos_suma_por_dias <- resumen_member %>%
  group_by(modad_dia_semana) %>%
  summarize(total_viajes = sum(total_viajes), .groups = "drop")

# Obtener el día con mayor frecuencia (moda)
moda_dia_semana_mem <- datos_suma_por_dias$moda_dia_semana[which.max(datos_suma_por_dias$total_viajes)]
moda_dia_semanas <- moda_dia_semana_mem

# Crear el gráfico de barras
grafico_moda_mem <- ggplot(datos_suma_por_dias, aes(x = moda_dia_semana, y = total_viajes, fill = moda_dia_semana)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = total_viajes), color = "black", vjust = 1) + # Ajustar la posición vertical de las etiquetas
  labs(title = "Moda de días de la semana membresia", x = "Día de la semana", y = "Frecuencia") +
  scale_fill_manual(values = c("gray", "red"),
                    labels = c("Otros días", moda_dia_semana_mem),
                    guide = guide_legend(title = "Día con mayor frecuencia")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = scales::comma)

# Mostrar el gráfico
grafico_moda_mem

```



En este gráfico se puede observar que el tipo de usuario membresía lo ocupa el servicio de bicicletas los días jueves y los martes.

Como podemos observar en este gráfico muestra la tendencia en que los usuarios sin importar de su estatus utilizan las bicicletas, en este caso el uso de las bicicletas mas utilizadas son las clásicas, y las menos utilizadas son las electric bike.

Obtendremos el mes con mas demanda, con los datos ya filtrados anterior mente crearemos una tabla para mostrar el mes con mas demanda en todo el año.

```
# Calcular el número total de viajes por mes y tipo de usuario
viajes_por_mes_usuario <- datos_limpios %>%
  group_by(month, member_casual) %>%
  summarize(total_viajes = n(), .groups = "drop")

# Encontrar el mes con el máximo número de viajes
mes_max_viajes <- viajes_por_mes_usuario %>%
  filter(total_viajes == max(total_viajes)) %>%
  pull(month) %>%
  unique()

# Obtener el tipo de usuario que realizó la mayoría de los viajes en ese mes
tipo_usuario_max_viajes <- viajes_por_mes_usuario %>%
  filter(month %in% mes_max_viajes) %>%
  group_by(month, member_casual) %>%
  summarize(total_viajes = sum(total_viajes), .groups = "drop") %>%
  filter(total_viajes == max(total_viajes)) %>%
  pull(member_casual)
```

```
datos_2 <- data.frame(mes_max_viajes, tipo_usuario_max_viajes)
tablaMES <- datos_2 %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )
```

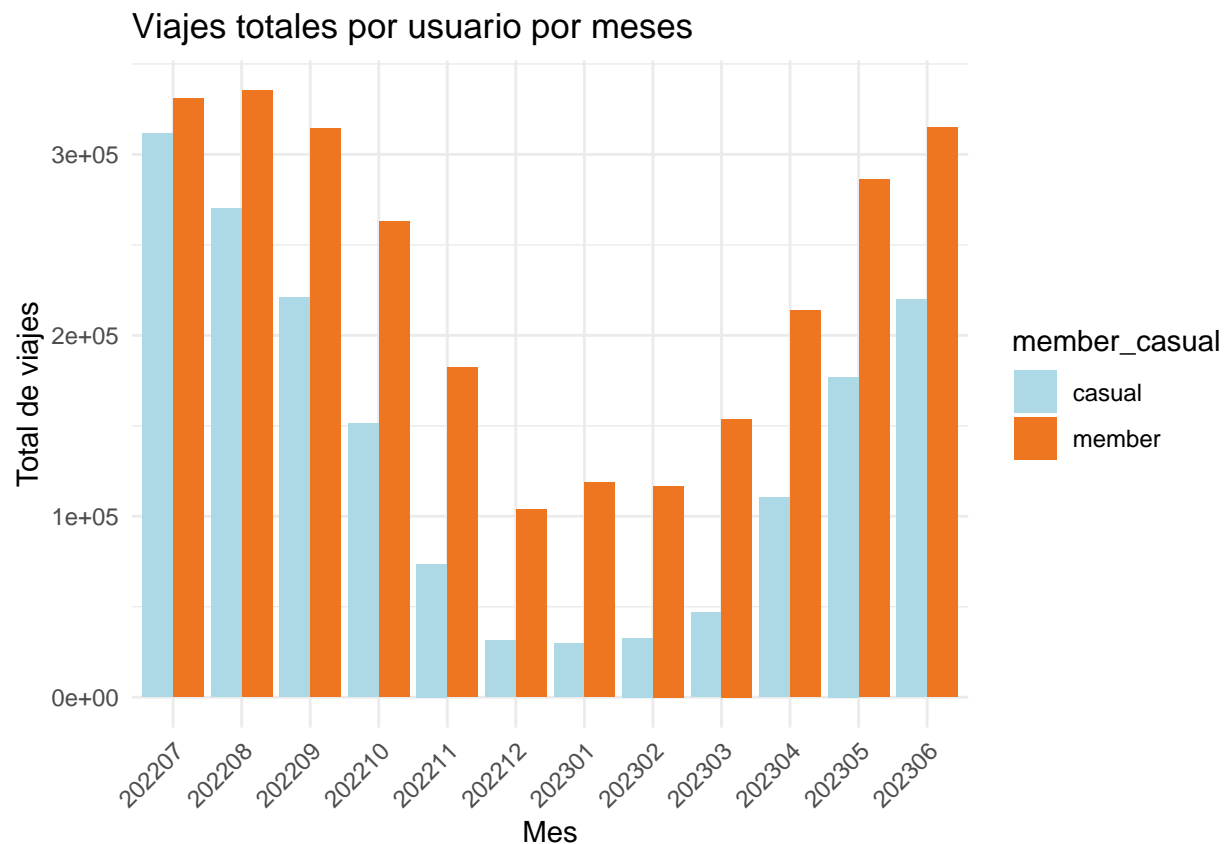
tablaMES

mes_max_viajes	tipo_usuario_max_viajes
202208	member

El mes con mas uso es el mes de agosto del 2022, y el usuario con mas uso es el usuario que tiene la membresia.

Para comprobar esto , se creara distintas tablas y gráficas don de respaldan esta información, en la primera gráfica representan los viajes totales en dada mes.

```
# Gráfico 1: Viajes totales por usuario por meses
ggplot(viajes_por_mes_usuario, aes(x = month, y = total_viajes, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Mes", y = "Total de viajes", title = "Viajes totales por usuario por meses") +
  scale_fill_manual(values = c("casual" = "lightblue", "member" = "#EE7621")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

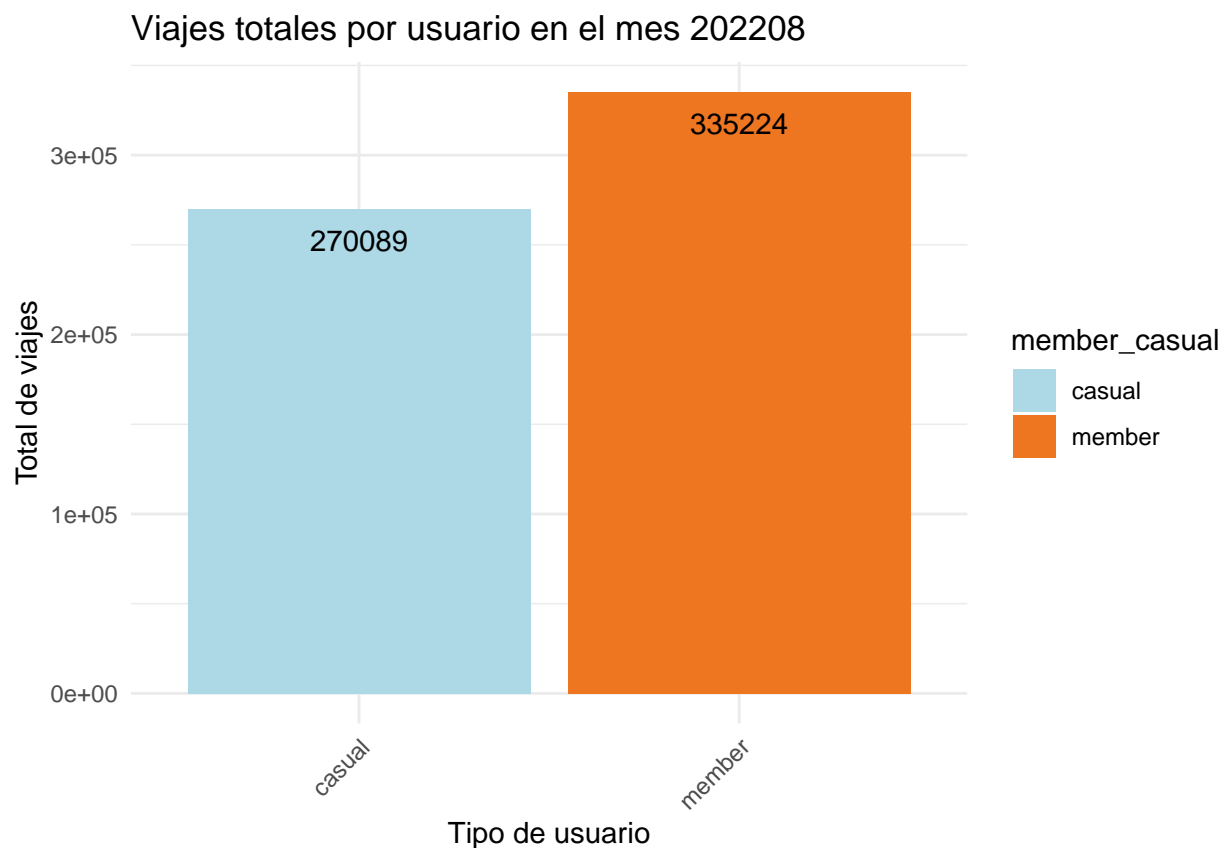


Como pondremos observar la información antes vistas es correcta ya que a comparación con otros meses el mes de agosto tiene una mayor frecuencia de usos por parte de los ciclistas con anualidad.

Se desglosa este mes para tener una mejor visualización del total de ciclistas en ese mes

```
# Filtrar los datos para incluir únicamente el mes con más viajes
viajes_mes_max_viajes <- viajes_por_mes_usuario %>%
  filter(month == mes_max_viajes)

# Gráfico: 2 Viajes totales por usuario en el mes con más viajes
ggplot(viajes_mes_max_viajes, aes(x = member_casual, y = total_viajes, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = total_viajes), color = "black", vjust = 2) +
  labs(x = "Tipo de usuario", y = "Total de viajes", title = paste("Viajes totales por usuario en el mes", mes_max_viajes)) +
  scale_fill_manual(values = c("casual" = "lightblue", "member" = "#EE7621")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Pero si este mes desglosamos por días, nos da una visión más amplia para realizar una decisión y crear una hipótesis para la estrategia de marketing,

```
viajes_mes_max_demanda <- datos_limpios %>%
  filter(month == mes_max_viajes)

# Calcular el número de viajes por día y tipo de usuario
viajes_por_dia <- viajes_mes_max_demanda %>%
  group_by(dia_semana, member_casual) %>%
  summarize(total_viajes = n(), .groups = "drop")

dia_max_demanda <- viajes_por_dia %>%
```

```

group_by(member_casual) %>%
  filter(total_viajes == max(total_viajes)) %>%
  pull(dia_semana)

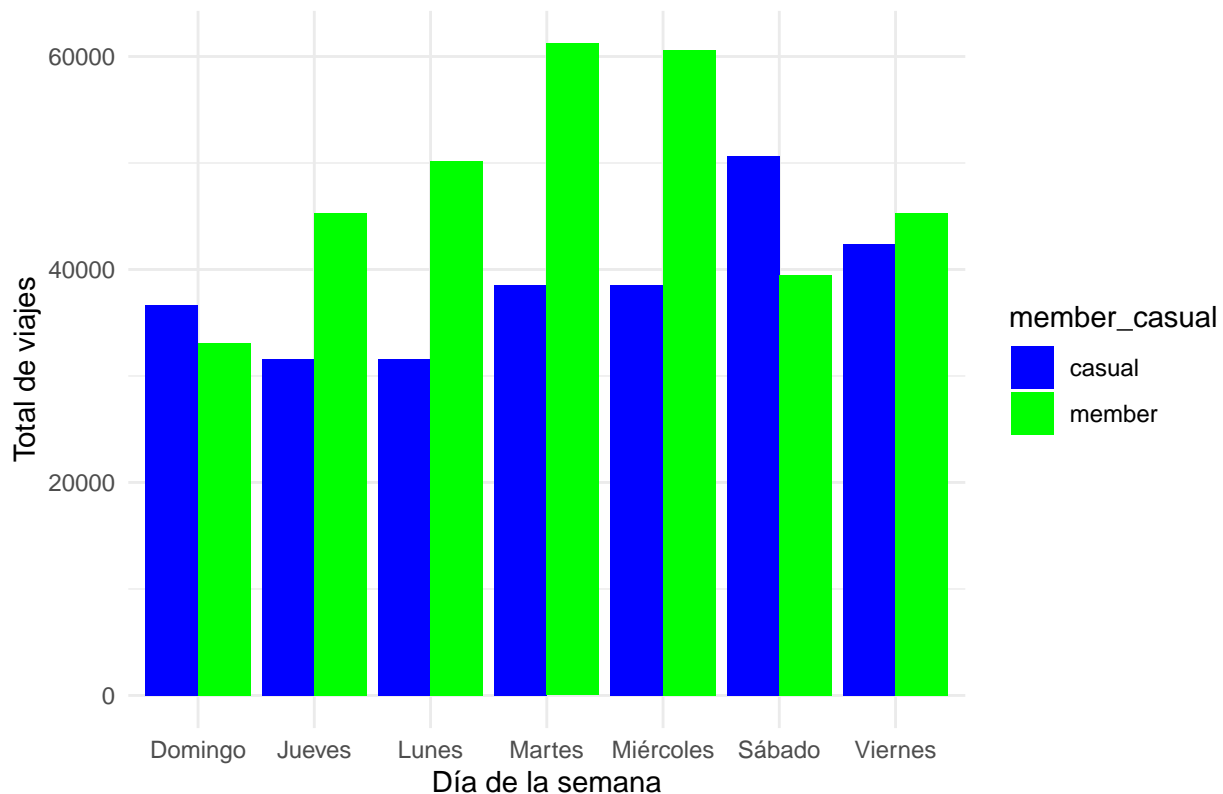
# Agregar columna para indicar si es el día de mayor demanda
viajes_por_dia <- viajes_por_dia %>%
  mutate(es_max_demanda = ifelse(dia_semana == dia_max_demanda, "Max Demanda", "No Max Demanda"))

# Filtrar los días con máximo número de viajes por tipo de usuario
viajes_max_por_usuario_dia <- viajes_por_dia %>%
  group_by(member_casual) %>%
  filter(total_viajes == max(total_viajes)) %>%
  ungroup()

ggplot(viajes_por_dia, aes(x = dia_semana, y = total_viajes, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Día de la semana", y = "Total de viajes", title = paste("Viajes por días en el mes", mes_ma
  scale_fill_manual(values = c("casual" = "blue", "member" = "green")) +
  theme_minimal()

```

Viajes por días en el mes 202208 por tipo de usuario



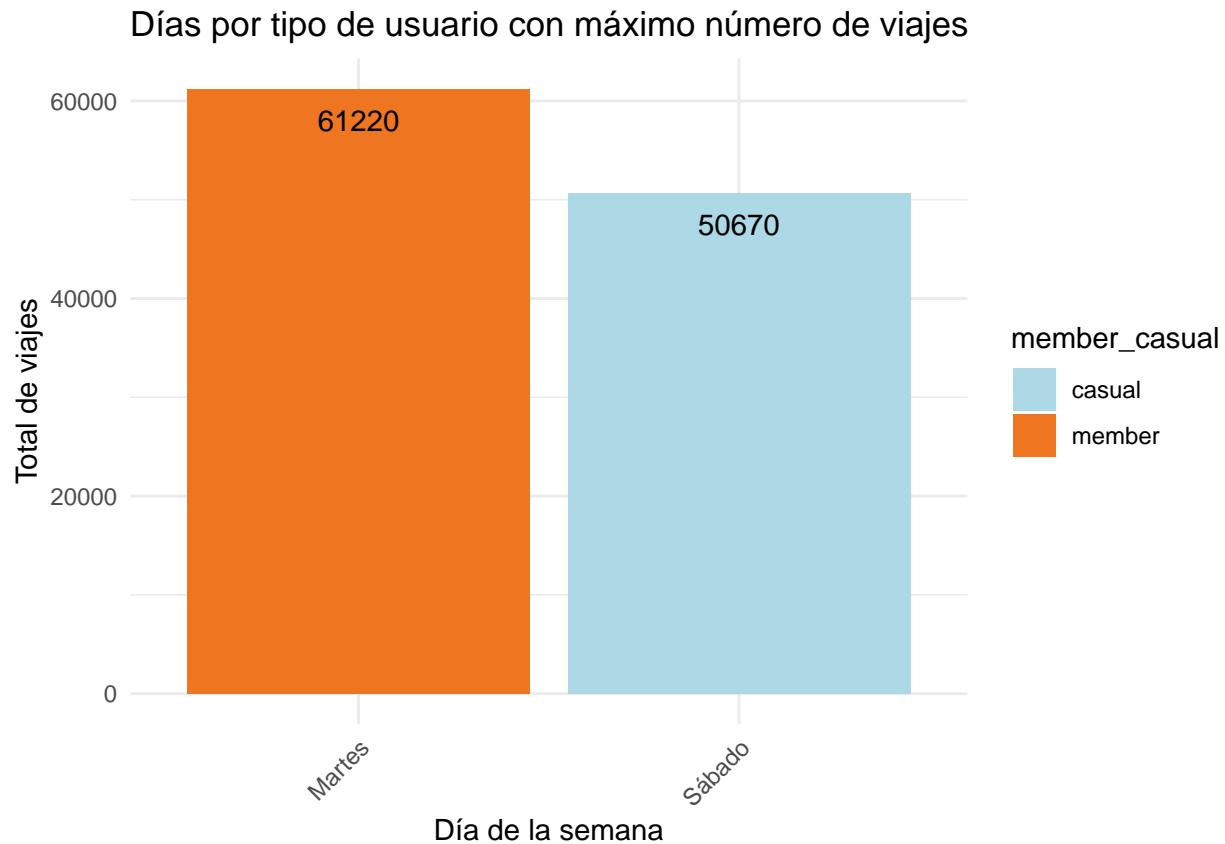
```

# Gráfico: Días por tipo de usuario con máximo número de viajes
ggplot(viajes_max_por_usuario_dia, aes(x = dia_semana, y = total_viajes, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = total_viajes), color = "black", vjust = 2) +
  labs(x = "Día de la semana", y = "Total de viajes", title = "Días por tipo de usuario con máximo número

```



```
scale_fill_manual(values = c("casual" = "lightblue", "member" = "#EE7621")) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



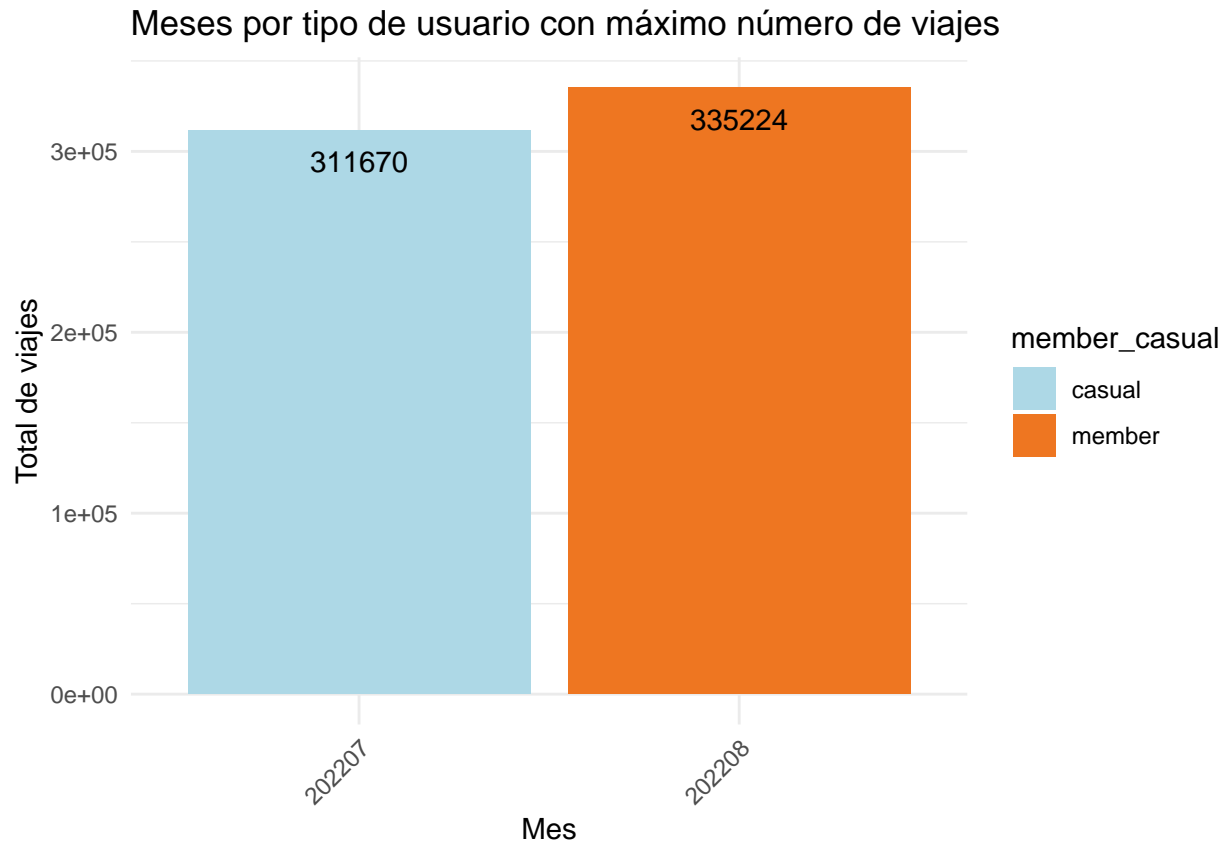
como podemos observar en estos dos gráficos los días con mas viajes son los sábados por los usuarios casuales, mientras que el día martes es el día con mas viajes por parte de los usuarios con anualidad. Podemos concluir que los ciclistas ocasionales utilizan las bicicletas en fines de semana mientras que los usuarios con anualidad entre semana.

Pero si lo vemos por separado el mes con más viajes por el tipo de usuario son julio y agosto con una mayor demanda a comparación de los demás meses.

Gráfico 3: Meses por tipo de usuario con máximo número de viajes

```
viajes_max_por_usuario <- viajes_por_mes_usuario %>%
  group_by(member_casual) %>%
  filter(total_viajes == max(total_viajes)) %>%
  ungroup()
```

```
ggplot(viajes_max_por_usuario, aes(x = month, y = total_viajes, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = total_viajes), color = "black", vjust = 2)+
  labs(x = "Mes", y = "Total de viajes", title = "Meses por tipo de usuario con máximo número de viajes") +
  scale_fill_manual(values = c("casual" = "lightblue", "member" = "#EE7621")) +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Con los datos de la estaciones podremos observar en que estación los usuarios comenzaron su recorrido y así podremos ver la estaciones con mas demanda y también la estaciones de destino con mas demanda.

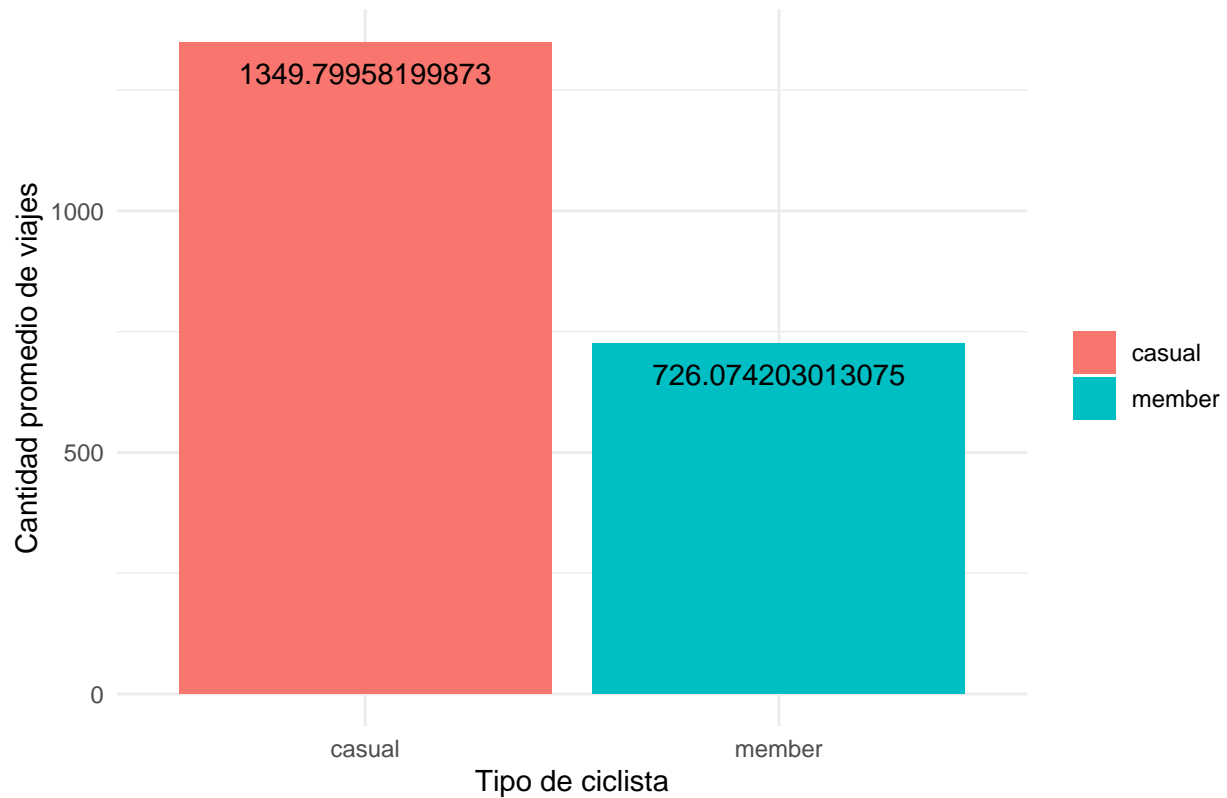
Pero si vemos el promedio de viajes por usuario la diferencia entre ambos, el ciclista ocasional tiene un mayor porcentaje como podemos observar en los siguiente gráficos,

```
# Calcular los valores promedio de viajes por socio/ocasional en el dataframe "datos limpios"
promedio_viajes <- datos_limpios %>%
  group_by(member_casual) %>%
  summarize(avg_viajes = mean(diff_double))

# Crear el gráfico de barras
grafico_barras <- ggplot(promedio_viajes, aes(x = member_casual, y = avg_viajes, fill = member_casual))
  geom_bar(stat = "identity") +
  geom_text(aes(label = avg_viajes), color = "black", vjust = 2) +
  labs(title = "Cantidad promedio de viajes por member y ciclistas ocasionales",
       x = "Tipo de ciclista",
       y = "Cantidad promedio de viajes") +
  theme_minimal() +
  theme(legend.title = element_blank())

print(grafico_barras)
```

Cantidad promedio de viajes por member y ciclistas ocasionales

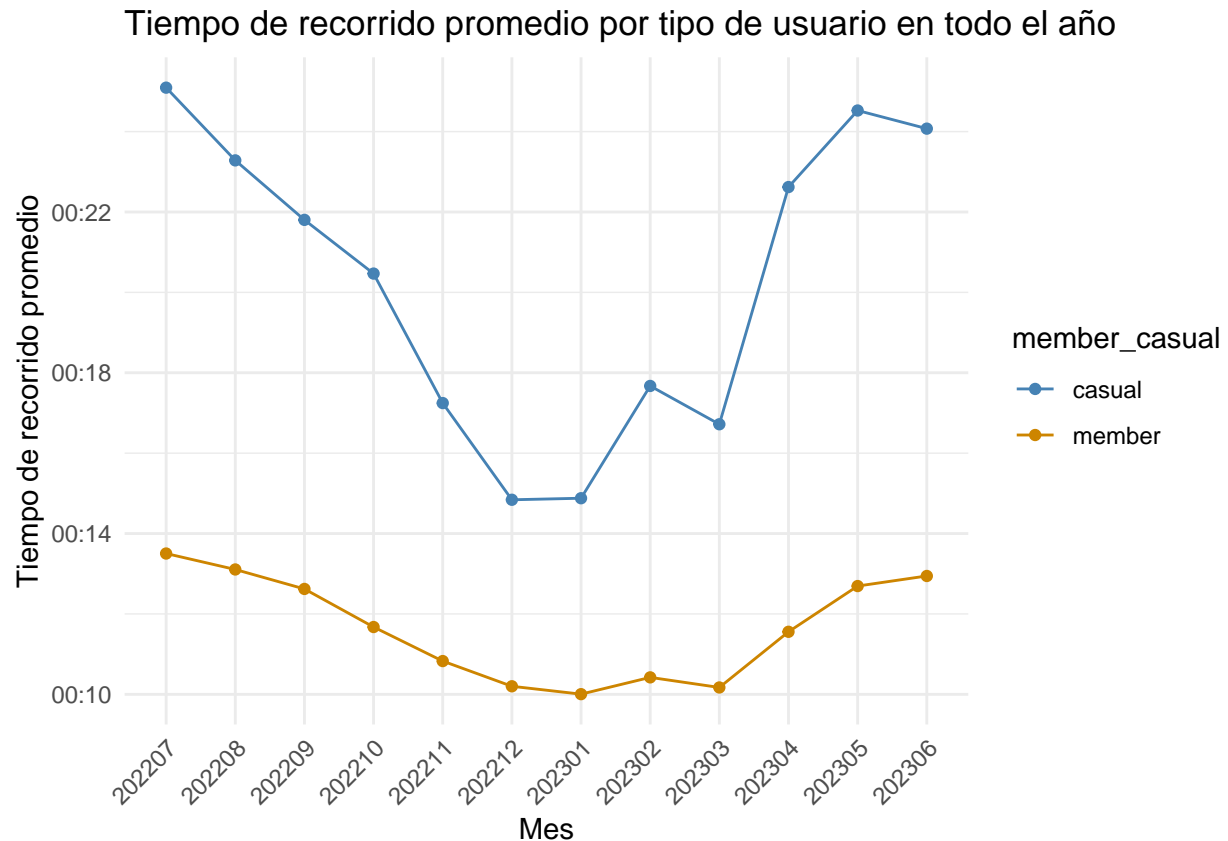


pero esto no quiere decir que los usuarios con anualidad tenga el tiempo con mas uso.

Entonces calculamos el promedio del tiempo en que los usuarios hicieron un recorrido y para tener una mejor visualización crearemos una gráfica de líneas donde representa el promedio de la duración de viajes por el tipo de usuario

```
# Calcular el tiempo de recorrido promedio por tipo de usuario en todo el año
tiempo_recorrido_promedio <- datos_limpios %>%
  group_by(month, member_casual) %>%
  summarize(tiempo_promedio = mean(diff_format), .groups = "drop") %>%
  arrange(month) # Ordenar por mes

# Crear el gráfico de líneas
ggplot(tiempo_recorrido_promedio, aes(x = month, y = tiempo_promedio, color = member_casual)) +
  geom_line(aes(group = member_casual)) + # Agregar la línea y especificar el mapeo de color
  geom_point() + # Agregar puntos para cada mes
  labs(x = "Mes", y = "Tiempo de recorrido promedio", title = "Tiempo de recorrido promedio por tipo de")
  scale_color_manual(values = c("casual" = "#4682B4", "member" = "#CD8500")) +
  scale_x_discrete() + # Forzar que el eje x muestre valores discretos (meses)
  scale_y_time(labels = scales::label_time(format = "%H:%M")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Esta gráfica representa que los usuarios casuales tienen un promedio mas grande uso, a comparación de los usuarios con anualidades

Pero queda otra duda, “¿Cual es la estación con mas demanda?”, esto se pude observar a través de la siguiente tabla y la siguiente gráfica en donde se pude observar que la estación con mas uso es la estación “Streeter & Grand Ave” la tiene una mayor demanda por los usuarios.

```
# Calcular el número de usuarios por estación de inicio
estaciones_inicio <- datos_limpios %>%
  group_by(start_station_name) %>%
  summarize(num_usuarios = n())
# Ordenar los datos por el número de usuarios descendente
estaciones_inicio <- estaciones_inicio[order(estaciones_inicio$num_usuarios, decreasing = TRUE), ]
# Tomar las 4 estaciones de inicio con mayor demanda
estaciones_inicio_mas_demanda <- head(estaciones_inicio, 4)
```

```
tablaEstaIN <- estaciones_inicio_mas_demanda %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE,
  )
```

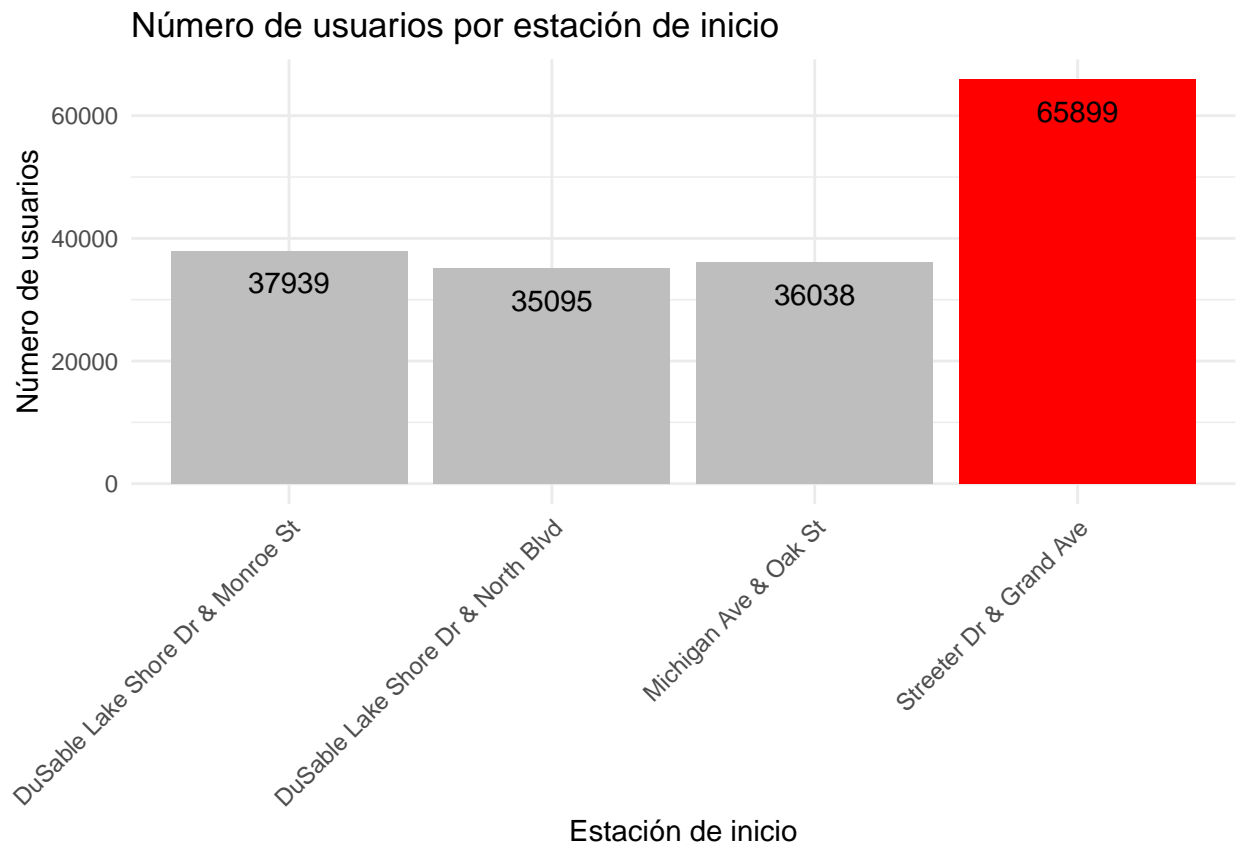
tablaEstaIN

start_station_name	num_usuarios
Streeter Dr & Grand Ave	65899
DuSable Lake Shore Dr & Monroe St	37939
Michigan Ave & Oak St	36038
DuSable Lake Shore Dr & North Blvd	35095

```
# Seleccionar la estación de inicio con mayor demanda
estacion_inicio_mas_demandada <- estaciones_inicio_mas_demanda[which.max(estaciones_inicio_mas_demanda$num_usuarios)]

# Crear el gráfico de barras
grafico_barras <- ggplot(estaciones_inicio_mas_demanda, aes(x = start_station_name, y = num_usuarios, fill = "gray")) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = num_usuarios), color = "black", vjust = 2) +
  labs(title = "Número de usuarios por estación de inicio",
       x = "Estación de inicio",
       y = "Número de usuarios") +
  scale_fill_manual(values = c("gray", "red"), guide = FALSE) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Mostrar el gráfico de barras
grafico_barras
```



Esta estación es de vital importancia estratégica debido a su gran influencia y alto tráfico de usuarios, como se puede observar en el siguiente gráfico en donde se representan las estaciones de destino. Al ser un punto de destino para numerosos usuarios.

```
# Calcular el número de usuarios por estación de final
estaciones_final <- datos_limpios %>%
  group_by(end_station_name) %>%
  summarize(num_usuarios = n())
# Ordenar los datos por el número de usuarios descendente
estaciones_final <- estaciones_final[order(estaciones_final$num_usuarios, decreasing = TRUE), ]
# Tomar las 4 estaciones de final con mayor demanda
estaciones_final_mas_demanda <- head(estaciones_final, 4)
```

```
# Mostrar las estaciones de final con más demanda
tablaEstaFIN <- estaciones_final_mas_demanda %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )
```

tablaEstaFIN

end_station_name	num_usuarios
Streeter Dr & Grand Ave	67543
DuSable Lake Shore Dr & North Blvd	38030
Michigan Ave & Oak St	36979
DuSable Lake Shore Dr & Monroe St	36806

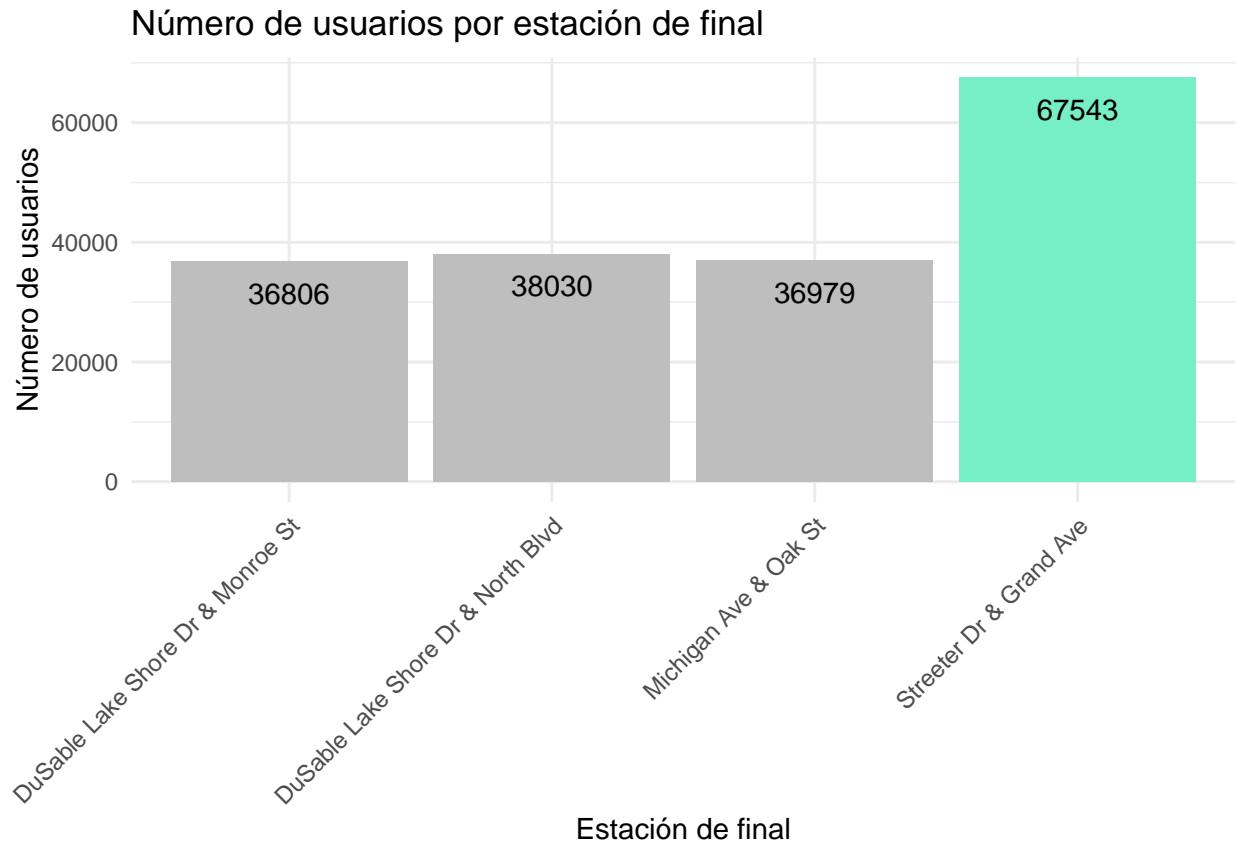
```
# Calcular el número de usuarios por estación de final
estaciones_final <- datos_limpios %>%
  group_by(end_station_name) %>%
  summarize(num_usuarios = n())

# Ordenar los datos por el número de usuarios descendente
estaciones_final <- estaciones_final[order(estaciones_final$num_usuarios, decreasing = TRUE), ]
estaciones_final <- head(estaciones_final, 4)

# Tomar la estación de final con mayor demanda
estacion_mas_demanda <- estaciones_final$end_station_name[1]

# Crear el gráfico de barras
grafico_barras <- ggplot(estaciones_final, aes(x = end_station_name, y = num_usuarios, fill = end_station_name)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = num_usuarios), color = "black", vjust = 2) +
  labs(title = "Número de usuarios por estación de final",
       x = "Estación de final",
       y = "Número de usuarios") +
  scale_fill_manual(values = c("gray", "#76EEC6"), guide = FALSE) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Mostrar el gráfico de barras
grafico_barras
```



Pero estos datos son poco precisos para dar una respuesta para la pregunta, en este caso se recomienda crear datos separarlos por el tipo de usuario. Como podemos observar el tipo de usuario casual la estación con mas influencia es la de “Streeter & Grand Ave” y para el caso del usuario con anualidad es “Kingsbury St & Kinzie St”

```
# Calcular el número de usuarios por estación de inicio y tipo de usuario
estaciones_inicio_por_usuario <- datos_limpios %>%
  group_by(start_station_name, member_casual) %>%
  summarize(num_usuarios = n()) %>%
  ungroup()

# Ordenar los datos por el número de usuarios descendente
estaciones_inicio_por_usuario <- estaciones_inicio_por_usuario %>%
  arrange(desc(num_usuarios))

# Tomar las 4 estaciones de inicio con mayor demanda para cada tipo de usuario
estaciones_inicio_mas_demanda_por_usuario <- estaciones_inicio_por_usuario %>%
  group_by(member_casual) %>%
  slice_max(order_by = num_usuarios, n = 5)

# Separar los datos por tipo de usuario
estaciones_inicio_demanda_casual <- estaciones_inicio_mas_demanda_por_usuario %>%
  filter(member_casual == "casual") %>%
  select(start_station_name, num_usuarios)

estaciones_inicio_demanda_member <- estaciones_inicio_mas_demanda_por_usuario %>%
  filter(member_casual == "member") %>%
  select(start_station_name, num_usuarios)
```

```
select(start_station_name, num_usuarios)

# Mostrar las estaciones de inicio con más demanda para el casual
tablaInicioCasual <- estaciones_inicio_demanda_casual %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )

tablaInicioCasual
```

member_casual	start_station_name	num_usuarios
casual	Streeter Dr & Grand Ave	49774
casual	DuSable Lake Shore Dr & Monroe St	29076
casual	Michigan Ave & Oak St	22385
casual	Millennium Park	21953
casual	DuSable Lake Shore Dr & North Blvd	20318

```
# Mostrar las estaciones de inicio con más demanda para el member
tablaInicioMember <- estaciones_inicio_demanda_member %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )

tablaInicioMember
```

member_casual	start_station_name	num_usuarios
member	Kingsbury St & Kinzie St	23490
member	Clark St & Elm St	21682
member	Clinton St & Washington Blvd	21016
member	Wells St & Concord Ln	19747
member	Loomis St & Lexington St	19587

Pero en el caso de destino es igual se mantiene la estación , Streeter Dr & Grand Ave y la estación Kingsbury St & Kinzie St. Ambas estaciones mantienen una mayor demanda y relevancia.

```
# Calcular el número de usuarios por estación de destino y tipo de usuario
estaciones_destino_por_usuario <- datos_limpios %>%
  group_by(end_station_name, member_casual) %>%
  summarize(num_usuarios = n()) %>%
  ungroup()

# Ordenar los datos por el número de usuarios descendente
estaciones_destino_por_usuario <- estaciones_destino_por_usuario %>%
  arrange(desc(num_usuarios))

# Tomar las 4 estaciones de destino con mayor demanda para cada tipo de usuario
estaciones_destino_mas_demanda_por_usuario <- estaciones_destino_por_usuario %>%
  group_by(member_casual) %>%
  slice_max(order_by = num_usuarios, n = 4)

# Separar los datos por tipo de usuario
estaciones_destino_demanda_casual <- estaciones_destino_mas_demanda_por_usuario %>%
  filter(member_casual == "casual") %>%
```



```

select(end_station_name, num_usuarios)

estaciones_destino_demanda_member <- estaciones_destino_mas_demanda_por_usuario %>%
  filter(member_casual == "member") %>%
  select(end_station_name, num_usuarios)

# Mostrar las estaciones de destino con más demanda para el TipoUsuario1
tablaDestinoCasual <- estaciones_destino_demanda_casual %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )

```

tablaDestinoCasual

member_casual	end_station_name	num_usuarios
casual	Streeter Dr & Grand Ave	53249
casual	DuSable Lake Shore Dr & Monroe St	27000
casual	Millennium Park	23974
casual	Michigan Ave & Oak St	23898

```

# Mostrar las estaciones de destino con más demanda para el TipoUsuario2
tablaDestinoMember <- estaciones_destino_demanda_member %>%
  kable() %>%
  kableExtra::kable_styling(
    full_width = TRUE
  )

```

tablaDestinoMember

member_casual	end_station_name	num_usuarios
member	Kingsbury St & Kinzie St	23465
member	Clinton St & Washington Blvd	21992
member	Clark St & Elm St	21828
member	Wells St & Concord Ln	20645

Ahora falta ver cual es la bicicleta mas popular entre los usuarios, por lo tanto Crearemos una gráfica que muestra los tipos de bicicletas junto con la cantidad de viajes realizados por cada una de ellas en cada mes. De esta manera, podremos identificar fácilmente cuál tipo de bicicleta es la más popular entre los usuarios y cuál tiene una menor demanda a lo largo del tiempo.

```

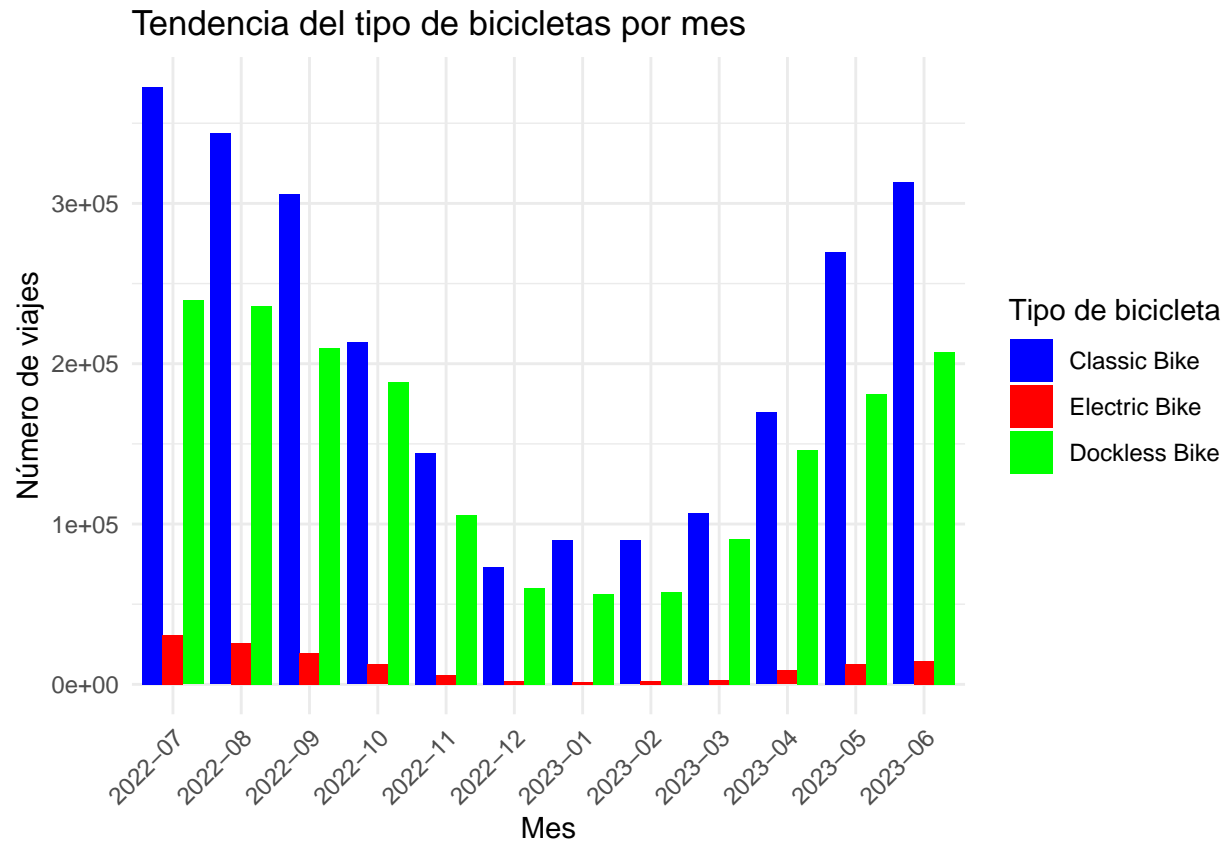
# Crear un nuevo dataframe con el conteo de rideable_type por mes
conteo_rideable <- datos_limpios %>%
  mutate(month = format(started_at, "%Y-%m")) %>%
  group_by(month, rideable_type) %>%
  summarize(numero_viajes = n()) %>%
  ungroup()

# Crear el gráfico de barras apiladas
grafico_rideable <- ggplot(conteo_rideable, aes(x = month, y = numero_viajes, fill = rideable_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Tendencia del tipo de bicicletas por mes", x = "Mes", y = "Número de viajes") +

```

```
scale_fill_manual(values = c("blue", "red", "green"),
                  labels = c("Classic Bike", "Electric Bike", "Dockless Bike"),
                  guide = guide_legend(title = "Tipo de bicicleta")) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Mostrar el gráfico
grafico_rideable
```



Como podemos observar la bicicleta mas utilizada por los usuarios en cada mes es la bicicleta classic, y la bicicleta con un bajo uso es la electric, pero para tener una mejor precisión podemos separarla en dos gráficas una para cada tipo de usuario.

```
# Crear un nuevo dataframe con el conteo de rideable_type por mes y tipo de usuario
conteo_rideable_usuario <- datos_limpios %>%
  mutate(month = format(started_at, "%Y-%m")) %>%
  group_by(month, member_casual, rideable_type) %>%
  summarize(numero_viajes = n()) %>%
  ungroup()
```

```
# Obtener el total de viajes por cada bicicleta para cada tipo de usuario y mes
total_viajes_bicicleta_mes <- conteo_rideable_usuario %>%
  group_by(month, member_casual, rideable_type) %>%
  summarize(total_viajes = sum(numero_viajes))
```

```
# Crear dos tablas separadas por tipo de usuario con las diferentes bicicletas y el total de viajes por
```

```

tabla_miembros <- total_viajes_bicicleta_mes %>%
  filter(member_casual == "member") %>%
  select(month, rideable_type, total_viajes) %>%
  rename(Tipo_de_usuario = member_casual, Total_de_viajes = total_viajes)

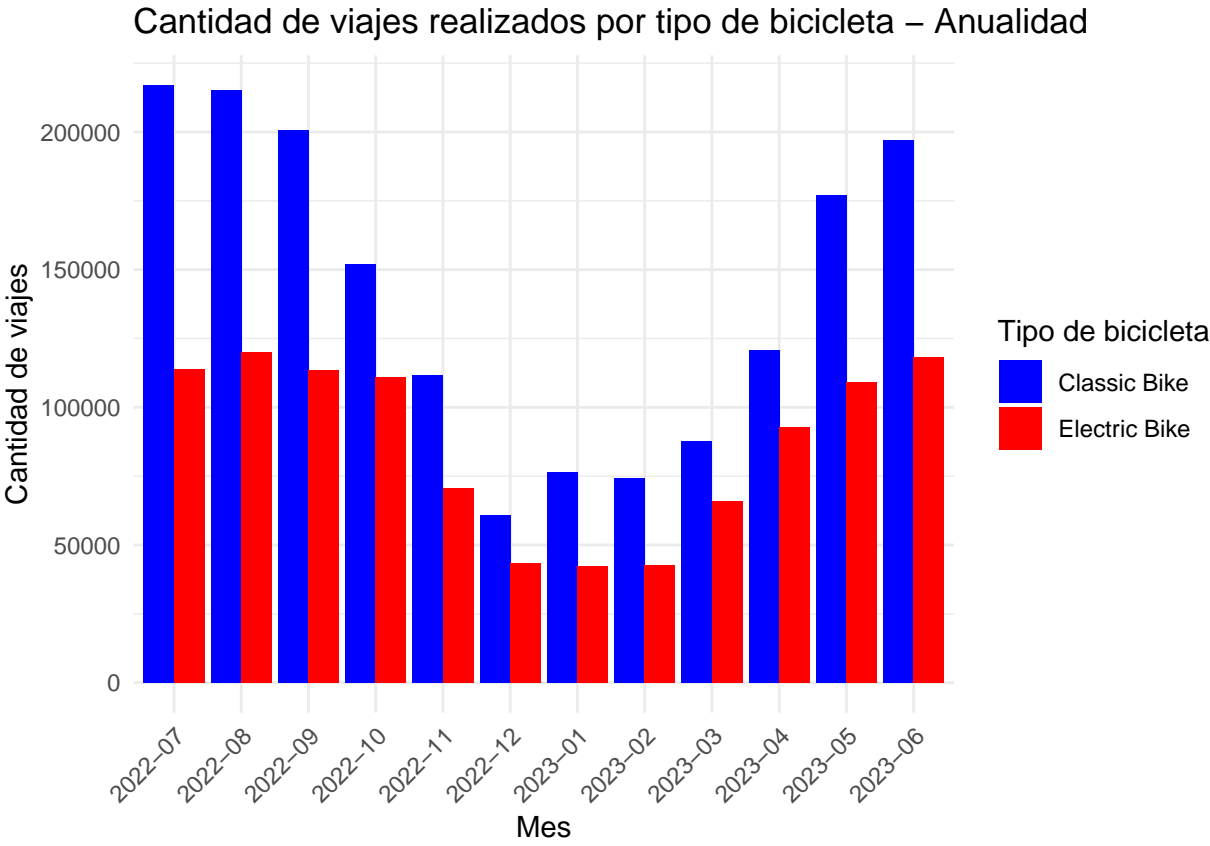
tabla_casuales <- total_viajes_bicicleta_mes %>%
  filter(member_casual == "casual") %>%
  select(month, rideable_type, total_viajes) %>%
  rename(Tipo_de_usuario = member_casual, Total_de_viajes = total_viajes)

# Crear gráfico para miembros
grafico_miembros <- ggplot(tabla_miembros, aes(x = month, y = Total_de_viajes, fill = rideable_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Cantidad de viajes realizados por tipo de bicicleta - Anualidad",
       x = "Mes",
       y = "Cantidad de viajes",
       fill = "Tipo de bicicleta") +
  scale_fill_manual(values = c("blue", "red", "green"),
                    labels = c("Classic Bike", "Electric Bike", "Dockless Bike"),
                    guide = guide_legend(title = "Tipo de bicicleta")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

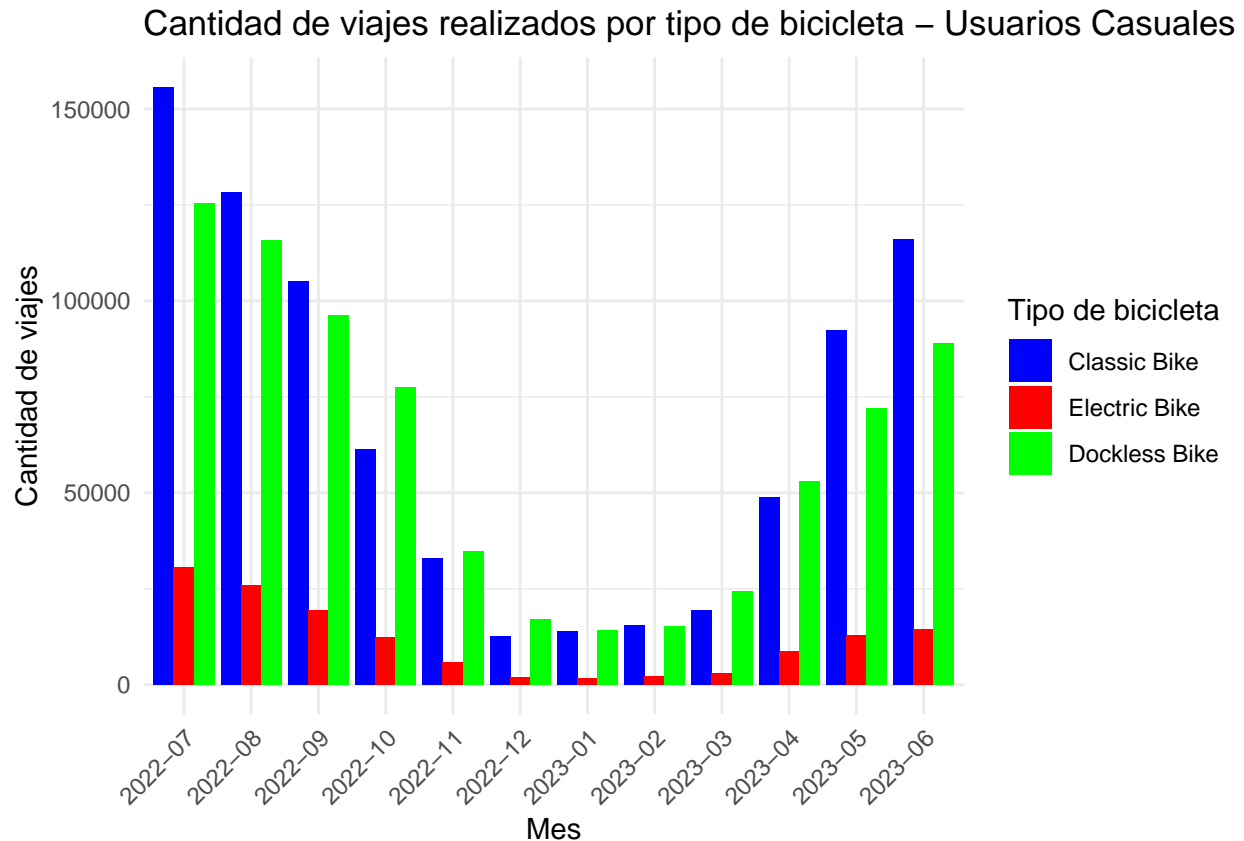
# Crear gráfico para usuarios casuales
grafico_casuales <- ggplot(tabla_casuales, aes(x = month, y = Total_de_viajes, fill = rideable_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Cantidad de viajes realizados por tipo de bicicleta - Usuarios Casuales",
       x = "Mes",
       y = "Cantidad de viajes",
       fill = "Tipo de bicicleta") +
  scale_fill_manual(values = c("blue", "red", "green"),
                    labels = c("Classic Bike", "Electric Bike", "Dockless Bike"),
                    guide = guide_legend(title = "Tipo de bicicleta")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Mostrar los gráficos
grafico_miembros

```



grafico_casuales



Podemos observar que los usuarios con membresía anual utilizan únicamente dos tipos de bicicletas: la bicicleta clásica y la bicicleta eléctrica. Por otro lado, los usuarios ocasionales hacen uso de las tres bicicletas disponibles, siendo las más utilizadas la bicicleta clásica y la bicicleta dockless. Estos datos reflejan una clara preferencia hacia la bicicleta clásica tanto para los usuarios con membresía anual como para los usuarios ocasionales, mientras que la bicicleta dockless también es bastante popular entre los usuarios ocasionales.

Actuar

Tres recomendaciones mas importantes

Como se puede apreciar en los gráficos anteriores, los usuarios ocasionales tienden a tener recorridos más largos en promedio cuando utilizan una bicicleta en comparación con los usuarios con membresía, independientemente del mes del año. Sin embargo, es importante destacar que existen ocasiones en las que los usuarios ocasionales utilizan las bicicletas con una menor frecuencia, pero sus tiempos promedio de recorrido son más largos.

Otra observación relevante se refiere a los días con mayor demanda. Los usuarios ocasionales muestran una mayor preferencia por utilizar las bicicletas durante los fines de semana, mientras que los usuarios con membresía tienen una mayor frecuencia de uso en los días entre semana.

Además, hemos identificado que el mes con más viajes realizados por parte de los usuarios con membresía es agosto, mostrando una alta actividad durante ese período del año. Por otro lado, es posible predecir que los usuarios ocasionales utilizan las bicicletas con mayor intensidad durante los meses vacacionales.

otra observación importante es la diferencia en el uso de tipos de bicicletas entre los usuarios con membresía anual y los usuarios ocasionales. Los datos muestran que los usuarios con membresía tienden a utilizar solo dos tipos de bicicletas, mientras que los usuarios ocasionales tienden a hacer uso de los tres tipos de bicicletas

disponibles.

Esta disparidad en la elección de bicicletas puede deberse a las preferencias y necesidades específicas de cada grupo de usuarios. Los usuarios con membresía anual pueden tener una mayor familiaridad y comodidad con ciertos tipos de bicicletas, lo que los lleva a utilizar principalmente esas opciones. Por otro lado, los usuarios ocasionales pueden variar sus elecciones de bicicletas según la disponibilidad, la duración del recorrido o sus preferencias personales.

Estas observaciones resaltan las diferencias en los patrones de uso entre los usuarios con membresía anual y los usuarios ocasionales, lo que puede ser de gran utilidad para adaptar y mejorar los servicios de bicicletas compartidas según las necesidades y preferencias de cada grupo de usuarios.

Las recomendaciones son

1. Ajustar publicidad en las estaciones de inicio con mas demanda por parte de los usuarios ocasionales, en donde se informe sobre los beneficios de la memebresía
2. Mejorar la experiencia de los usuarios, es decir tener la disponibilidad de bicicletas
3. Ajustar promociones en meses con mayor de manda de los usuarios ocasionales para promover ofertar atractivas para los usuarios ocasionales.