Case study: Cyclistic bike-share analysis

Elizaveta Kalacheva

14 August 2024

Introduction

The organization had developed a large bicycle park throughout Chicago. Bicycle can be unlocked in any station and returned to any other station at any moment. The organization's main strategy was price flexibility: single trip, day trip, and annual membership. Only one trip and a day trip plane are casual users. Financial analytics indicate that annual members are more profitable than casual users. Consequently, the directors believe that increasing the number of annual members will boost profits. They plan to implement a new marketing strategy aimed at converting casual users into annual members.

Objectives

- Identify existing differences in bicycle usage between casual and member users
- Develop a new marketing strategy to convert casual bike users into members.

Data Collecation

For this analysis, historical data of all trips during the year 2023 will be utilized. This data is publicly available from the organization's website.The data consisted of separate datasets for each month:

```
jan_23 <- read.csv("202301-divvy-tripdata.csv", sep=",", header=TRUE)
feb_23 <- read.csv("202302-divvy-tripdata.csv", sep=",", header=TRUE)

mar_23 <- read.csv("202303-divvy-tripdata.csv", sep=",", header=TRUE)
abr_23 <- read.csv("202304-divvy-tripdata.csv", sep=",", header=TRUE)
may_23 <- read.csv("202305-divvy-tripdata.csv", sep=",", header=TRUE)

jul_23 <- read.csv("202307-divvy-tripdata.csv", sep = ",", header = TRUE)
jun_23 <- read.csv("202306-divvy-tripdata.csv", sep=",", header = TRUE)
aug_23 <- read.csv("202308-divvy-tripdata.csv", sep=",", header=TRUE)

sep_23 <- read.csv("202309-divvy-tripdata.csv", sep=",", header=TRUE)
oct_23 <- read.csv("202310-divvy-tripdata.csv", sep=",", header=TRUE)
nov_23 <- read.csv("202311-divvy-tripdata.csv", sep=",", header=TRUE)
dec_23 <- read.csv("202312-divvy-tripdata.csv", sep=",", header=TRUE)
```

Data preprocessing

These are packages that were used in this analysis:

```
library(readr) #for read read.csv
library(tidyr) #for data manipulation
library(dplyr) #data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(ggplot2) #creating plots and graphs
library(lubridate) #date-time manipulation
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(FNN) #k-nearest neighbors (KNN)
library(class) #for knn function fill_station_name_knn
```

```
##
## Attaching package: 'class'
```

```
## The following objects are masked from 'package:FNN':
##
##     knn, knn.cv
```

```
library(viridis) #color palettes
```

```
## Loading required package: viridisLite
```

```
library(leaflet) #creating interactive maps
library(sf) #handling spatial data
```

```
## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

```r
library(htmlwidgets) #saving interactive visualizations as HTML files
library(webshot) #taking screenshots of web content
library(patchwork) #combining multiple ggplot2 plots with functions like plot_layout
library(scales) #format numbers for better readability
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:viridis':
##
##     viridis_pal

## The following object is masked from 'package:readr':
##
##     col_factor
```

Since the data was provided as separate datasets for each month, it was necessary to merge them into a single dataframe to facilitate easier manipulation.

```r
trips <- bind_rows(jan_23, feb_23, mar_23, abr_23, may_23, jun_23, jul_23, aug_23,
                   sep_23, oct_23, nov_23, dec_23)
```

There are empty rows only in four columns, such as station names and station IDs. Therefore, the dataframe will be split into two dataframes for further analysis. The dates_trips dataframe will consist of dates for further analysis, while the stations dataframe will contain information about stations and their coordinates.

```r
dates_trips <- trips %>%
  select(ride_id, rideable_type, started_at, ended_at, member_casual)

stations <- trips %>%
 select(ride_id, start_station_name, start_station_id, end_station_name, end_station_id,
        start_lat, start_lng, end_lat, end_lng)

stations <- bind_rows(
  stations %>%
    select(ride_id, start_station_name, start_station_id, start_lat, start_lng) %>%
    rename(
      station = start_station_name,
      id_station = start_station_id,
      latitude = start_lat,
      longitude = start_lng
    ),

stations %>%
  select(ride_id, end_station_name, end_station_id, end_lat, end_lng) %>%
    rename(
      station = end_station_name,
      id_station = end_station_id,
      latitude = end_lat,
      longitude = end_lng
    )
)
```

```r
print(head(stations))
```

```
##            ride_id                         station   id_station latitude
## 1 F96D5A74A3E41399   Lincoln Ave & Fullerton Ave TA1309000058 41.92407
## 2 13CB7EB698CEDB88         Kimbark Ave & 53rd St TA1309000037 41.79957
## 3 BD88A2E670661CE5        Western Ave & Lunt Ave       RP-005 42.00857
## 4 C90792D034FED968         Kimbark Ave & 53rd St TA1309000037 41.79957
## 5 3397017529188E8A         Kimbark Ave & 53rd St TA1309000037 41.79957
## 6 58E68156DAE3E311 Lakeview Ave & Fullerton Pkwy TA1309000019 41.92607
##   longitude
## 1 -87.64628
## 2 -87.59475
## 3 -87.69048
## 4 -87.59475
## 5 -87.59475
## 6 -87.63886
```

Creating the unique_station dataframe is the next step, as it is needed to identify all stations used by casual and annual members throughout the year.

```r
unique_stations <- stations %>%
  distinct(station, .keep_all = TRUE)
```

The column with dates in the dates_trips dataframe will be converted to date format.

```r
dates_trips$started_at <- ymd_hms(dates_trips$started_at)
dates_trips$ended_at <- ymd_hms(dates_trips$ended_at)
```

Moreover, the day of the week, month, and time will be split into different columns for further analysis.

```r
dates_trips$day_of_week <- wday(dates_trips$started_at, label = TRUE, week_start = 7)

dates_trips <- dates_trips %>%
  mutate(month = month(started_at, label = TRUE),
         hour = factor(hour(started_at),
                       levels = 0:23,
                       labels = c("12 AM", "1 AM", "2 AM", "3 AM", "4 AM", "5 AM",
                                  "6 AM", "7 AM", "8 AM", "9 AM", "10 AM", "11 AM",
                                  "12 PM", "1 PM", "2 PM", "3 PM", "4 PM", "5 PM",
                                  "6 PM", "7 PM", "8 PM", "9 PM", "10 PM", "11 PM")))
```

In the following code chunk, the days of the week and months were ordered sequentially and the time was organized by hour.

```r
custom_order <- c("Sun", "Sat", "Fri", "Thu", "Wed",  "Tue", "Mon")
custom_month <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct",
                  "Nov", "Dec")
custom_hour <- c("12 AM", "1 AM", "2 AM", "3 AM", "4 AM", "5 AM", "6 AM", "7 AM", "8 AM",
                 "9 AM", "10 AM", "11 AM", "12 PM", "1 PM", "2 PM", "3 PM", "4 PM", "5 PM", "6 PM",
                 "7 PM", "8 PM", "9 PM", "10 PM", "11 PM")
```

Exploratory Data Analysis

Initially, the work started with the stations dataframe. Due to the large volume of data, it was necessary to identify the top 20 most frequently used stations. However, it was found that the top station is unknown:

```
stations$coordinate <- paste(stations$latitude, stations$longitude, sep = " ")

stations_top <- stations %>%
  group_by(coordinate) %>%
  summarise(count = n(),
            station = first(station)) %>%
  arrange(desc(count)) %>%
  slice_head(n = 21)
print(stations_top)
```

```
## # A tibble: 21 x 3
##    coordinate                     count station
##    <chr>                          <int> <chr>
##  1 41.892278 -87.612043          110354 "Streeter Dr & Grand Ave"
##  2 41.880958 -87.616743           67762 "DuSable Lake Shore Dr & Monroe St"
##  3 41.911722 -87.626804           64143 "DuSable Lake Shore Dr & North Blvd"
##  4 41.90096039 -87.62377664       59424 "Michigan Ave & Oak St"
##  5 41.902973 -87.63128            57397 "Clark St & Elm St"
##  6 41.88917683258 -87.6385057718  54297 "Kingsbury St & Kinzie St"
##  7 41.912133 -87.634656           52883 "Wells St & Concord Ln"
##  8 41.926277 -87.630834           52735 "Theater on the Lake"
##  9 41.89 -87.63                   50808 ""
## 10 41.88338 -87.64117             50115 "Clinton St & Washington Blvd"
## # i 11 more rows
```

The k-nearest neighbors (KNN) method was used to identify the name of the unknown station based on its coordinates. The FNN package and the `unique_stations` dataframe, which includes the coordinates of known stations, were used to find the nearest known stations and predict the name of the unknown station.

```
top <- stations_top %>%
  separate(coordinate, into = c("latitude", "longitude"), sep = " ")

#unknown station
stations_with_missing <- data.frame(
  lat = c(41.89, 41.91),
  lng = c(-87.63, -87.63),
  station = c(NA, NA)
)
#knn function
fill_station_name_knn <- function(unknown_coords, unique_stations, k = 3) {
  knn_result <- knn(
    train = unique_stations[, c("latitude", "longitude")],
    test = unknown_coords,
    cl = unique_stations$station,
    k = k
  )
  return(knn_result)
}
```

```r
#extract coordinates for station with missing name
unknown_coords <- stations_with_missing %>%
  filter(is.na(station)) %>%
  select(lat, lng)

#fill the station with missing name
filled_station_name <- fill_station_name_knn(unknown_coords, unique_stations)

#to see the name of station
filled_station <- data.frame(
  lat = unknown_coords$lat,
  lng = unknown_coords$lng,
  filled_station_name = filled_station_name
)



#fill in the names for stations with missing names
stations_top <- top %>%
  mutate(station = ifelse(latitude == 41.89 & longitude == -87.63, "Wabash Ave & Grand Ave", station)) %
  mutate(station = ifelse(latitude == 41.91 & longitude == -87.63, "DuSable Lake Shore Dr & North Blvd"

stations_top <- stations_top %>%
  group_by(station) %>%
  summarise(count = sum(count),
            latitude = first(latitude),
            longitude = first(longitude)) %>%
  arrange(desc(count))

print(stations_top)
```

```
## # A tibble: 20 x 4
##    station                           count latitude         longitude
##    <chr>                             <int> <chr>            <chr>
##  1 Streeter Dr & Grand Ave          110354 41.892278        -87.612043
##  2 DuSable Lake Shore Dr & North Blvd 106404 41.911722      -87.626804
##  3 DuSable Lake Shore Dr & Monroe St  67762 41.880958       -87.616743
##  4 Michigan Ave & Oak St              59424 41.90096039      -87.62377664
##  5 Clark St & Elm St                  57397 41.902973        -87.63128
##  6 Kingsbury St & Kinzie St           54297 41.88917683258   -87.6385057718
##  7 Wells St & Concord Ln              52883 41.912133        -87.634656
##  8 Theater on the Lake                52735 41.926277        -87.630834
##  9 Wabash Ave & Grand Ave             50808 41.89            -87.63
## 10 Clinton St & Washington Blvd       50115 41.88338         -87.64117
## 11 Wells St & Elm St                  47973 41.903222        -87.634324
## 12 Millennium Park                    46708 41.8810317       -87.62408432
## 13 University Ave & 57th St           46309 41.791478        -87.599861
## 14 Broadway & Barry Ave               45506 41.9375823160063 -87.6440978050232
## 15 Ellis Ave & 60th St                44226 41.78509714636   -87.6010727606
## 16 Indiana Ave & Roosevelt Rd         43021 41.867888        -87.623041
## 17 Clark St & Armitage Ave            42920 41.918306        -87.636282
## 18 Wilton Ave & Belmont Ave           42792 41.9402319181086 -87.6529437303543
## 19 N Sheffield Ave & W Wellington Ave 41838 41.94            -87.65
```

```
## 20 Clinton St & Madison St              41599 41.8827519656856 -87.641190290451
```

After identifying the top station names, the next step was to calculate the ride duration. This was done by determining the difference between the start and end times of each ride.

```
dates_trips$ride_length <- dates_trips$ended_at - dates_trips$started_at
```

Null and negative values in the ride duration are considered errors; therefore, they will be removed from the dataframe.

```
dates_trips <- dates_trips %>% filter(ride_length > 0)
```

The estimation of the average ride length for each type of member is the next step. Additionally, converting the ride length into minutes facilitates data manipulation and prepares for estimating the average and total ride lengths for each type of bike.

```
analisis <- dates_trips %>%
  select(ride_id, rideable_type, member_casual, ride_length, day_of_week, month, hour)

analisis$ride_length_min <- analisis$ride_length/60

mean_length <- analisis %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_length_min),
            max = max(ride_length_min),
            min = min(ride_length_min))
mean_length$length_mean <- paste0(round(mean_length$mean), " min")
mean_length$length_max <- paste0(round((mean_length$max/60)/24), " day")
mean_length$length_min <- paste0(round(mean_length$min), " min")

type_total_length <- analisis%>%
  group_by(rideable_type) %>%
  summarise(length = sum(ride_length_min))

type_avg_length <- analisis %>%
  group_by(rideable_type) %>%
  summarise(mean_length = mean(ride_length_min))

print(mean_length)
```

```
## # A tibble: 2 x 7
##   member_casual mean          max        min   length_mean length_max length_min
##   <chr>         <drtn>        <drtn>     <drt> <chr>       <chr>      <chr>
## 1 casual        28.25396 secs 98489.067~ 0.01~ 28 min      68 day     0 min
## 2 member        12.52779 secs  1559.667~ 0.01~ 13 min      1 day      0 min
```

In this step, the dataframe was filtered to create two separate dataframes:one for annual members and other for casual users. This approach allowed for separate and more detailed analysis of each user group.

```
members <- analisis %>%
  filter(member_casual == "member")

casual <- analisis %>%
  filter(member_casual == "casual")
```

In addition, it is necessary to know the most popular days for bike usage. The analysis of the distribution of rides during the week requires correctly ordering and grouping the days to identify these days.

```
count_data <- analisis %>%
  group_by(member_casual, day_of_week) %>%
  summarize(count = n(), .groups = 'drop')

count_data$day_of_week <- factor(count_data$day_of_week, levels = custom_order)
```

Prepared two separate dataframes: one aggregating data by month for all users, and another aggregating data by month and user type to analyse usage patterns for each user type.

```
month <- analisis %>%
  group_by(month) %>%
  summarise(count=n(), mean_ride_length = mean(ride_length_min, na.rm = TRUE))

monthly <- analisis %>%
  group_by(member_casual, month) %>%
  summarise(count = n(), mean_ride_length = mean(ride_length_min, na.rm = TRUE))
```

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

```
#Reordering months to follow the calendar order and calculating the average ride duration

month$month <- factor(month$month, levels = custom_month)
month$mean_ride_length <- round(month$mean_ride_length)

monthly$month <- factor(monthly$month, levels = custom_month)
monthly$mean_ride_length <- round(monthly$mean_ride_length)
```

Preparation of the dataframe for hourly analysis_including calculating the number of rides per hour and reordering them to follow chronological order.

```
hour_data <- analisis %>%
  group_by(member_casual, hour) %>%
  summarize(count = n(), .groups = 'drop')
hour_data$hour <- factor(hour_data$hour, levels = custom_hour)

hour_general <- analisis %>%
  group_by(hour) %>%
  summarise(count=n())
hour_general$hour <- factor(hour_general$hour, levels = custom_hour)
```

Calculating the average ride length for each user by month

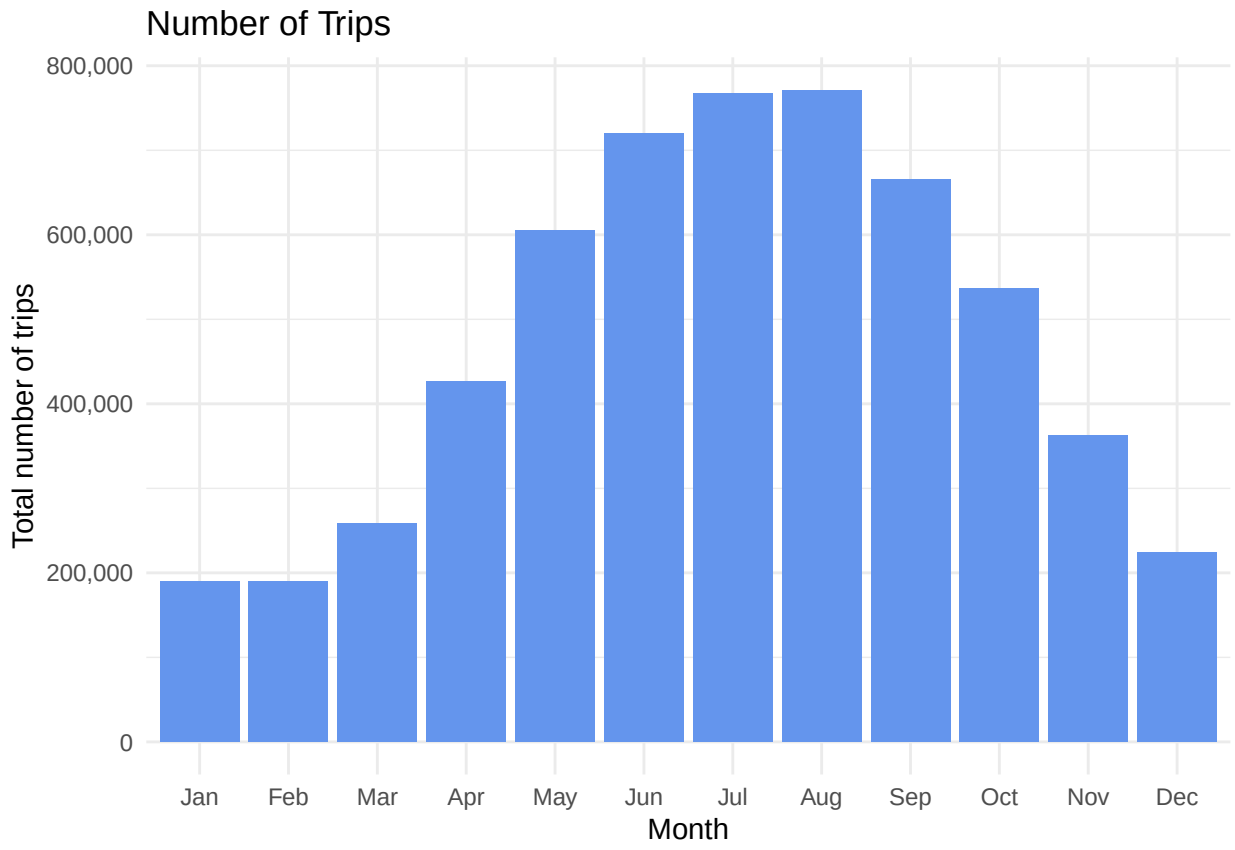
```
member_average <- members %>%
  select(ride_length_min, month) %>%
  group_by(month) %>%
  summarise(count=n(), mean_ride_length = mean(ride_length_min))
member_average$month <- factor(member_average$month, levels = custom_month)
member_average$mean_ride_length <- round(member_average$mean_ride_length)
```

```r
casual_average <- casual %>%
  select(ride_length_min, month) %>%
  group_by(month) %>%
  summarise(count=n(), mean_ride_length = mean(ride_length_min))
casual_average$month <- factor(casual_average$month, levels = custom_month)
casual_average$mean_ride_length <- round(casual_average$mean_ride_length)
```

Data Visualization

The bar chart below displays the distribution of trips throughout the year:

```r
ggplot(month, aes(x = month, y = count)) +
  geom_bar(stat = "identity", fill = "cornflowerblue") +
  scale_y_continuous(labels = comma) +
  labs(y = "Total number of trips", x = "Month", title = "Number of Trips") +
  theme_minimal()
```



The graph shows that the total number of trips peaked in August and gradually declined throughout autumn, with a significant drop in January and February.

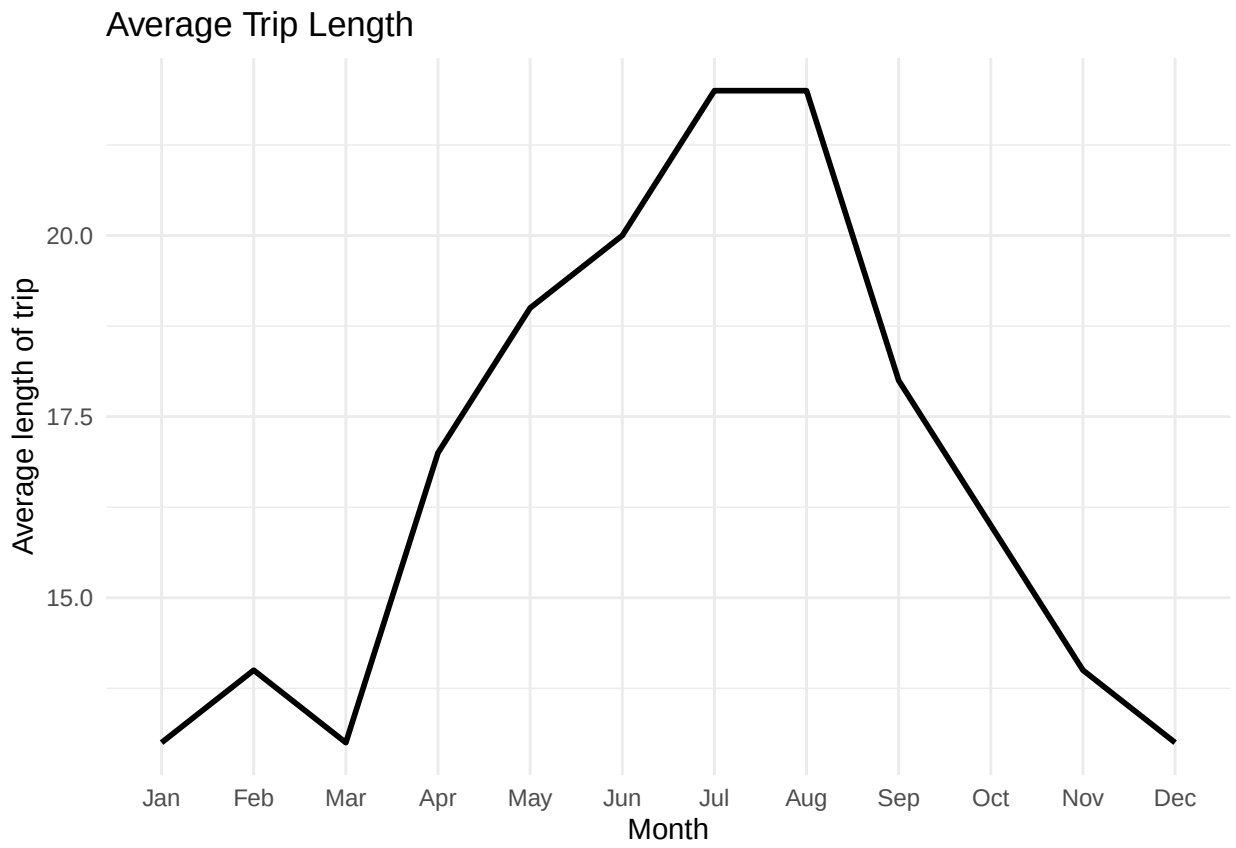
The next graph is the distribution of average length of trips:

```r
ggplot() +
  geom_line(data = month, aes(x = month, y = mean_ride_length, group = 1), color = "black", size = 1) +
```

```
  labs(y = "Average length of trip", x = "Month", title = "Average Trip Length") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

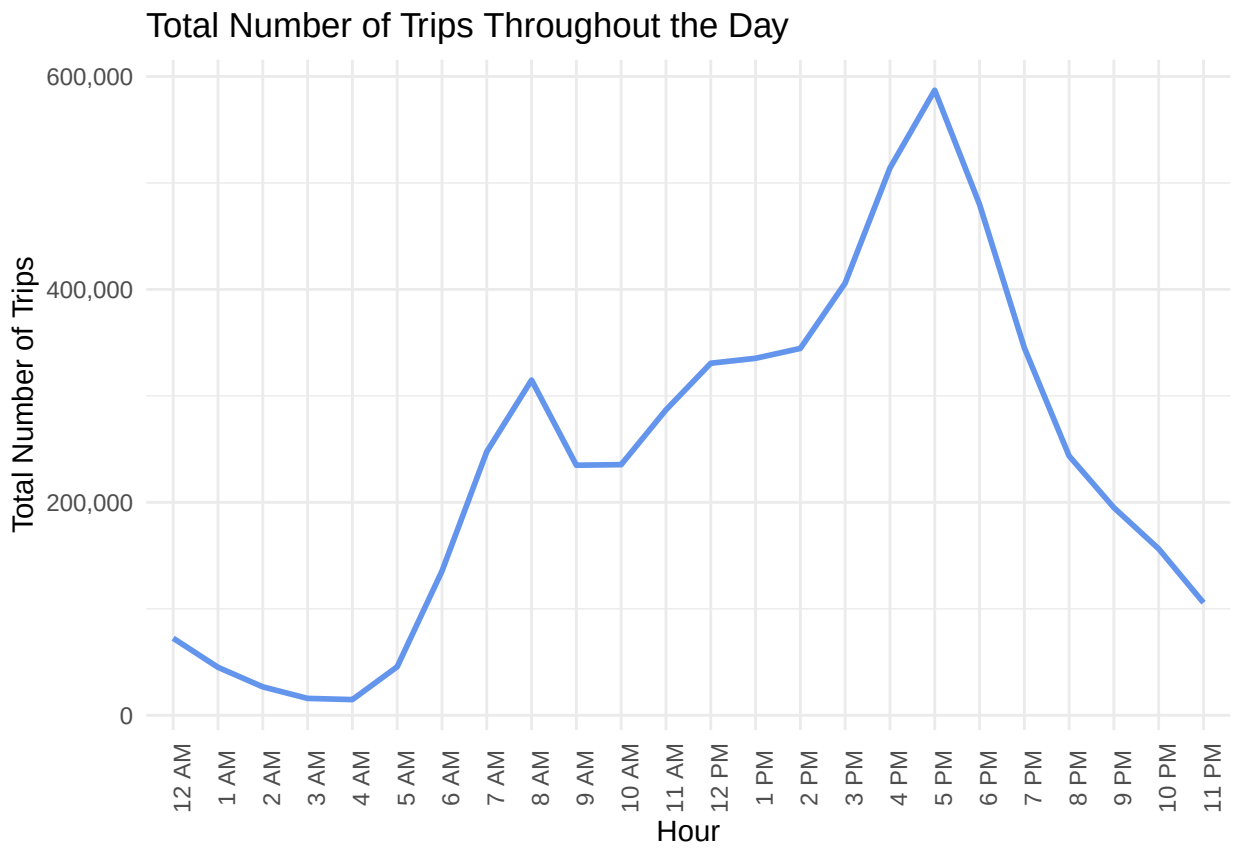
```
## Don’t know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



The line graph shows that the average ride length, similar to the number of trips, peaked during the summer months of July and August, and then plummeted during autumn, with a significant drop in December, January, and March.

It is now interesting to examine the distribution of trips throughout the day and at different times to determine which periods are most utilized:

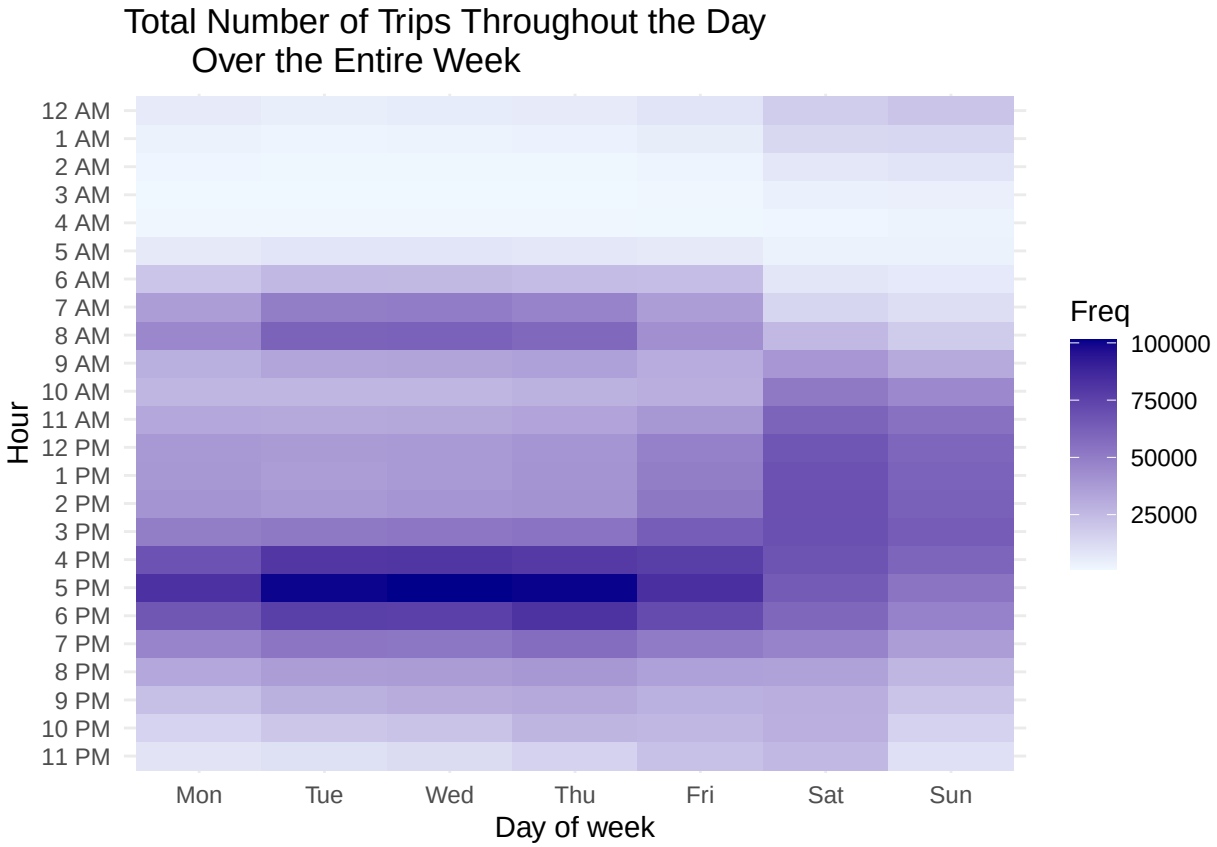
```
ggplot() +
  geom_line(data = hour_general, aes(x = hour, y = count, group = 1),
            color = "cornflowerblue", size = 1) +
  scale_y_continuous(labels = comma) +
  labs(y = "Total Number of Trips", x = "Hour", title = "Total Number of Trips Throughout the Day") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```



The graph above shows a sharp increase in the number of rides in the early morning, reaching its peak at 8 AM. There is a slight fluctuation afterward, leading to a second peak around 5 PM. Following this, the graph gradually declines, hitting its lowest point between 3 and 4 AM.

The next heatmap provides a more detailed view of the peak times throughout each day of the week:

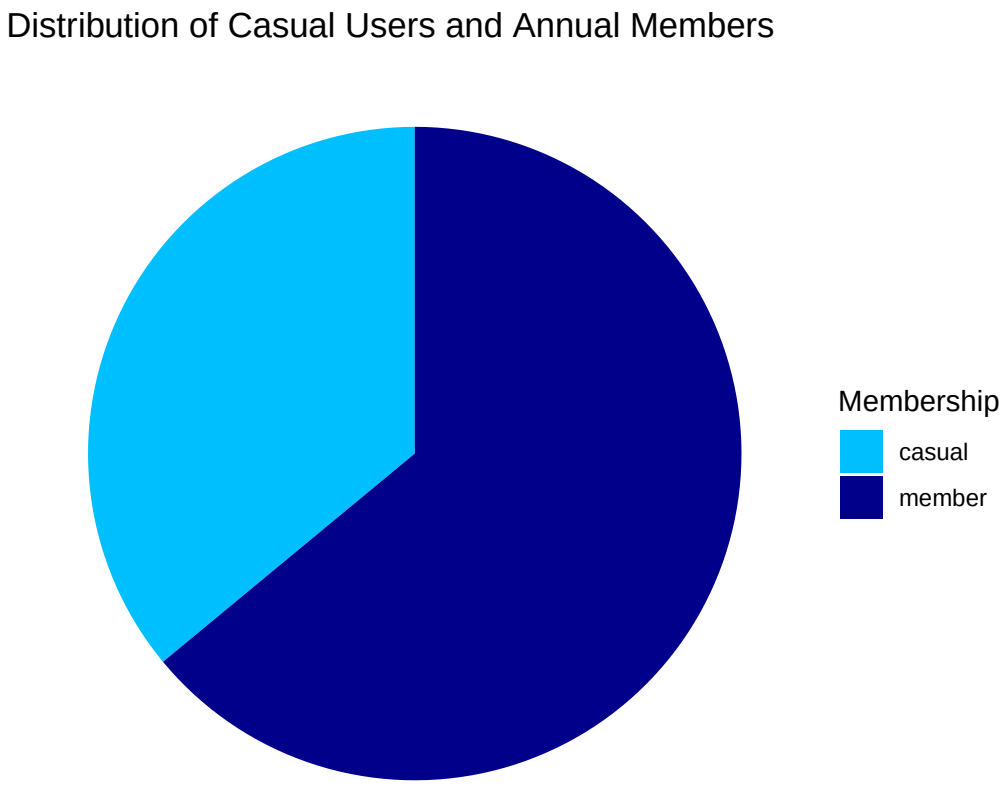
```
analisis$hour <- factor(analisis$hour, levels = c("11 PM", "10 PM", "9 PM", "8 PM", "7 PM",
                                                  "6 PM", "5 PM", "4 PM", "3 PM", "2 PM",
                                                  "1 PM", "12 PM", "11 AM",
                                                  "10 AM", "9 AM", "8 AM", "7 AM", "6 AM",
                                                  "5 AM", "4 AM", "3 AM", "2 AM",
                                                  "1 AM", "12 AM"))
analisis$day_of_week <- factor(analisis$day_of_week, levels = c("Mon", "Tue", "Wed", "Thu",
                                                                "Fri", "Sat", "Sun"))

analisis_freq <- analisis %>%
  count(day_of_week, hour, name = "Freq")

ggplot(analisis_freq, aes(day_of_week, hour, fill = Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "aliceblue", high = "darkblue") +
  theme_minimal() +
  labs(y = "Hour", x = "Day of week", title = "Total Number of Trips Throughout the Day
       Over the Entire Week")
```



The most demanded time is between 4 PM and 6 PM from Tuesday to Thursday, while on the weekend, the most popular hours shift to between 12 PM and 4 PM.

There are two types of membership: casual users and annual members. The distribution of each type is shown in the pie chart below.

```
colors <- c("classic_bike" = "cyan3", "docked_bike" = "darkslategray1",
            "electric_bike"="deepskyblue2", "member" = "darkblue", "casual"="deepskyblue")
labels_biketype <- c("classic_bike" = "classic bike", "docked_bike" = "docked bike",
                     "electric_bike"="electric bike", "1" = "casual", "2" = "member")

ggplot(analisis, aes(x = "", fill = member_casual)) +
  geom_bar(width = 1, stat = "count") +
  coord_polar(theta = "y") +
  theme_void() +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  labs(fill = "Membership") +
  labs(title = "Distribution of Casual Users and Annual Members")
```

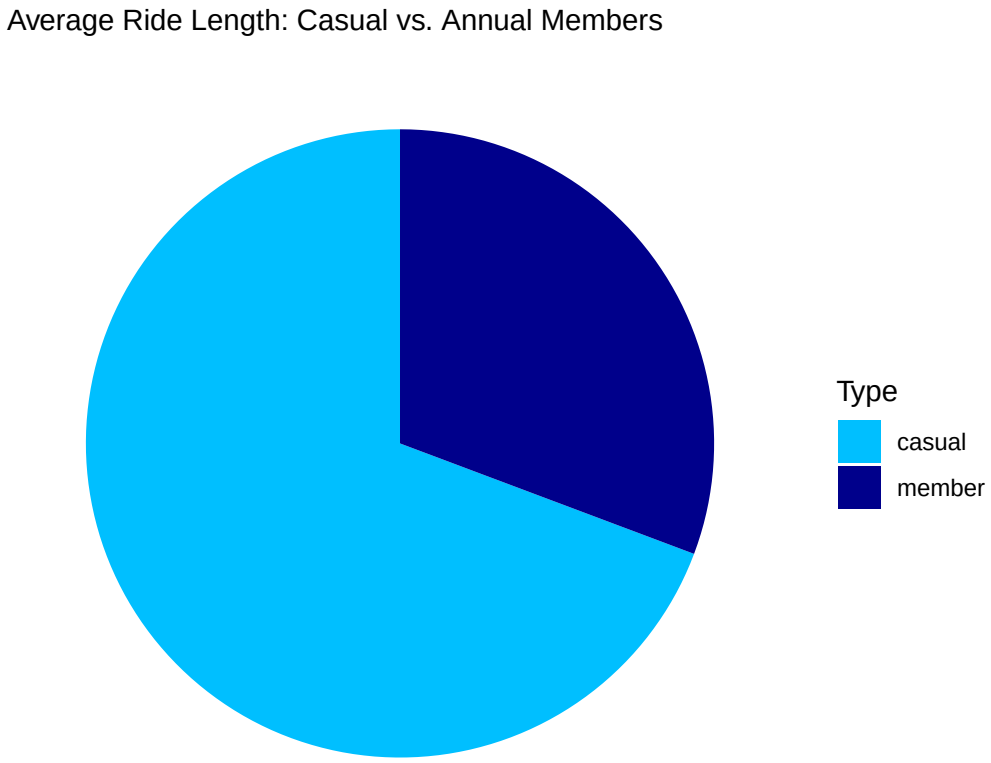


The given pie chart shows that the majority of users, 63%, are annual members, while remaining are casual users.

The pie chart below represents which type of users has the largest average ride time:

```
ggplot(mean_length, aes(x = "", y = mean, fill = member_casual)) +
  geom_bar(width = 1, stat = "identity", color = "transparent") +
  coord_polar("y", start = 0) +
  labs(subtitle = "Average Ride Length: Casual vs. Annual Members",
       x = NULL,
       y = NULL,
       fill = "Type") +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  theme_minimal() +
  theme(axis.text.x = element_blank(), panel.border = element_blank(),
        panel.grid = element_blank())
```

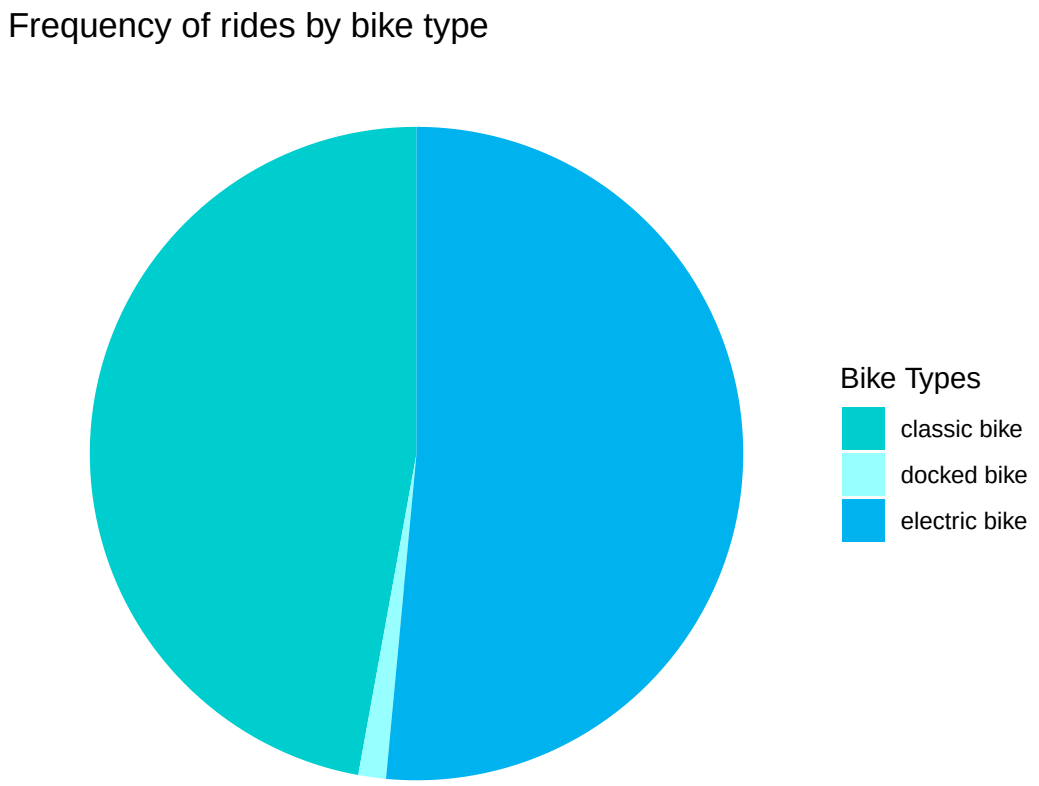
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



From the pie chart it can be seen that casual users spend approximately 1,86 times more time riding that member subscriptions.

Similarly, there are three types of bikes provided by the company. It will be interesting to examine the distribution of ride frequency across these bike types:

```
colors <- c("classic_bike" = "cyan3", "docked_bike" = "darkslategray1",
            "electric_bike"="deepskyblue2")
labels_biketype <- c("classic_bike" = "classic bike", "docked_bike" = "docked bike",
                     "electric_bike"="electric bike")
ggplot(analisis, aes(x = "", fill = rideable_type)) +
  geom_bar(width = 1, stat = "count") +
  coord_polar(theta = "y") +
  theme_void() +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  labs(fill = "Bike Types") +
  labs(title = "Frequency of rides by bike type")
```



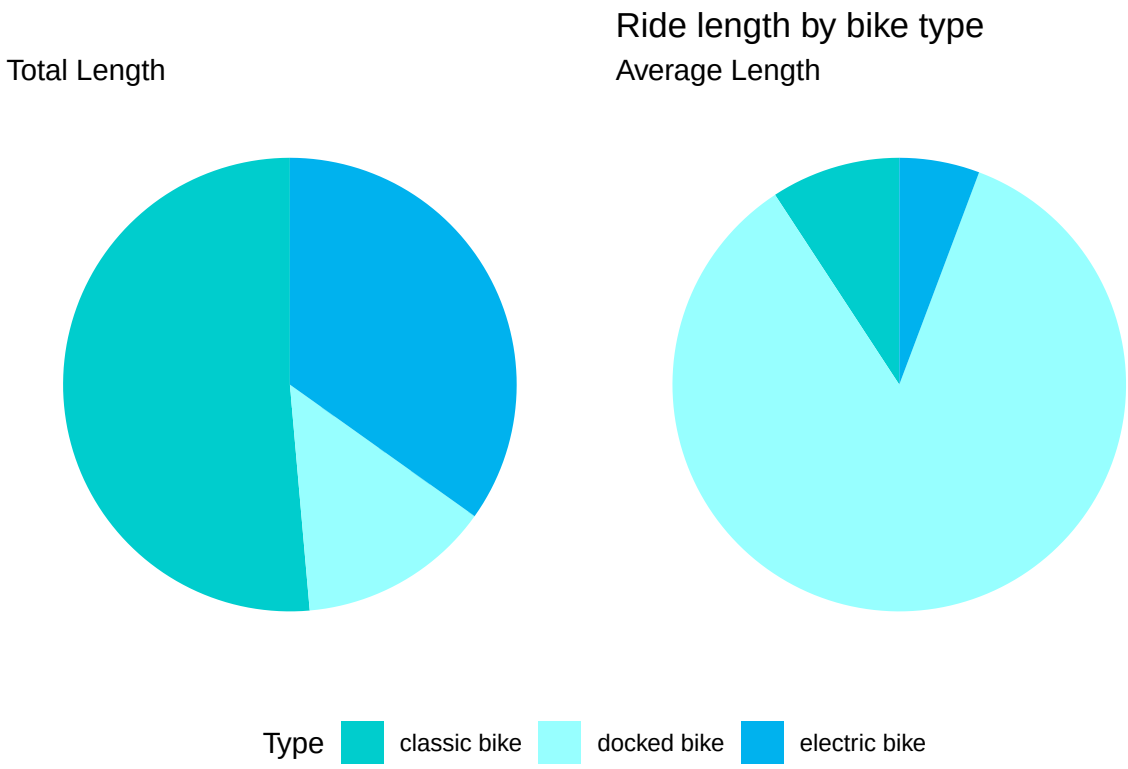
The most commonly used bike is the electric bike, while the docked bike is used less frequently. However, the docked bike category includes both classic and electric bikes, meaning that all types of bikes are docked.

In addition, the chart below represents the distribution of bike usage based on the total and average length of rides:

```
graph1 <- ggplot(type_total_length, aes(x = "", y = length, fill = rideable_type)) +
  geom_bar(width = 1, stat = "identity", color = "transparent") +
  coord_polar("y", start = 0) +
  labs(subtitle = "Total Length",
       x = NULL,
       y = NULL,
       fill = "Type") +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  theme_minimal() +
  theme(axis.text.x = element_blank(), panel.border = element_blank(),
        panel.grid = element_blank())

graph2 <- ggplot(type_avg_length, aes(x = "", y = mean_length, fill = rideable_type)) +
  geom_bar(width = 1, stat = "identity", color = "transparent") +
  coord_polar("y", start = 0) +
  labs(subtitle = "Average Length",
       x = NULL,
       y = NULL,
       fill = "Type") +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  theme_minimal() +
```

```
  theme(axis.text.x = element_blank(), panel.border = element_blank(),
        panel.grid = element_blank())

grpah_ride_length <- graph1 + graph2 + labs(title = "Ride length by bike type") +
  plot_layout(guides = 'collect') & theme(legend.position = "bottom")
print(grpah_ride_length)
```

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

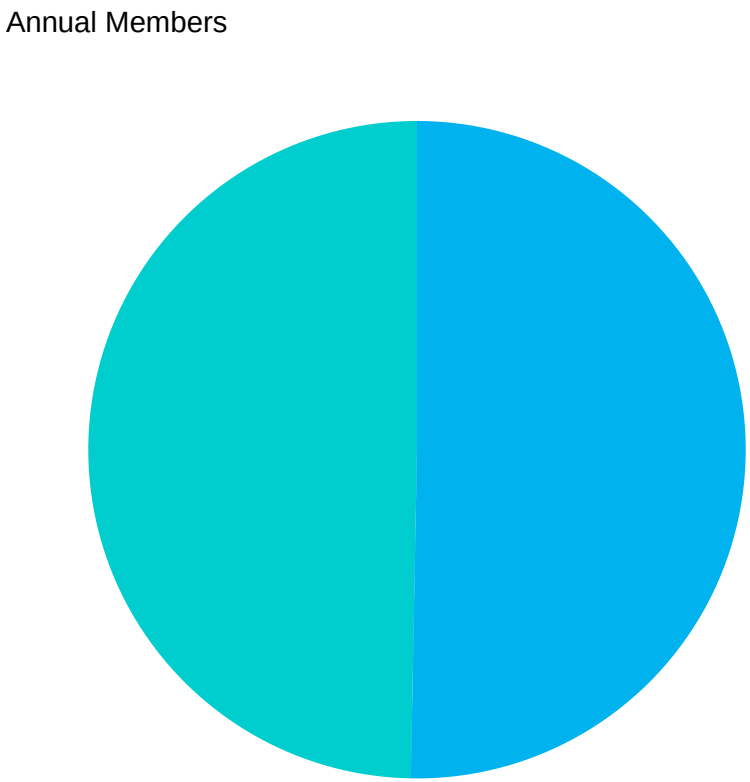


As mentioned earlier, the Chicago bike share system consist of docked type bikes, meaning that both electric and classic bikes must be returned to specific docking stations. Therefore, the average ride length by bike type alone may not fully capture user preferences. However, the pie chart depicting the total ride length shows that classic bikes are ridden for a longer duration than electric bikes.

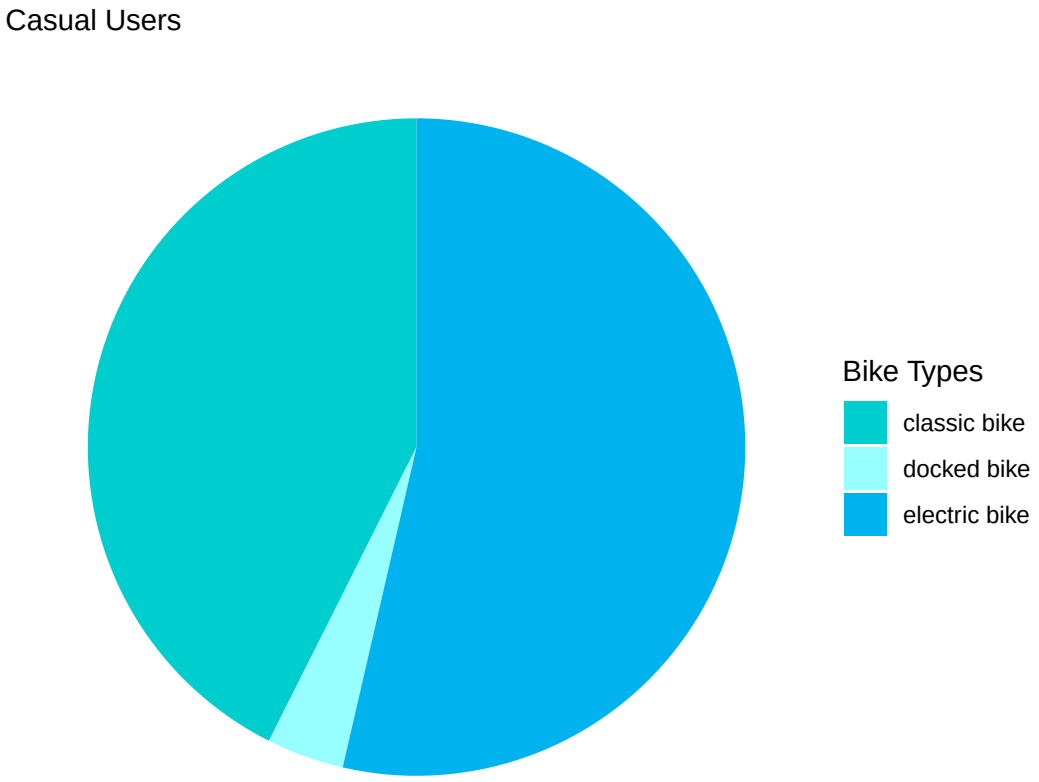
In the context of bike usage, it will also be interesting to examine the average ride length for different types of bikes. This analysis will help understand which bike types are preferred by different types of members:

```
graph3 <- ggplot(members, aes(x = "", fill = rideable_type)) +
  geom_bar(width = 1, stat = "count") +
  coord_polar(theta = "y") +
  theme_void() +
  scale_fill_manual(values = colors) +
  labs(subtitle = "Annual Members") +
```

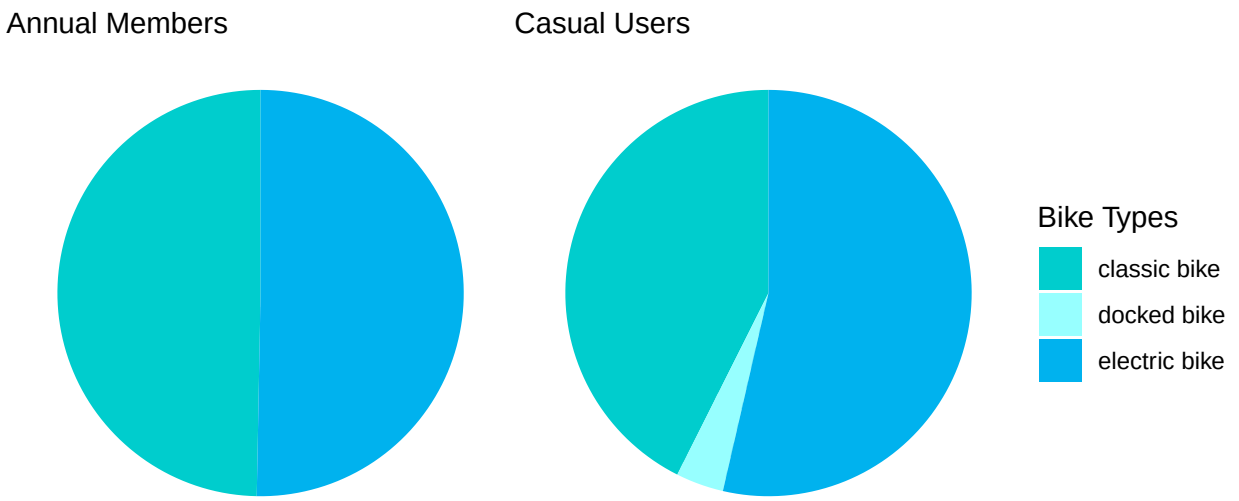
```
  theme(legend.position = "none")
print(graph3)
```




```
graph4 <- ggplot(casual, aes(x = "", fill = rideable_type)) +
  geom_bar(width = 1, stat = "count") +
  coord_polar(theta = "y") +
  theme_void() +
  scale_fill_manual(values = colors, labels = labels_biketype) +
  labs(fill = "Bike Types") +
  labs(subtitle = "Casual Users")
print(graph4)
```



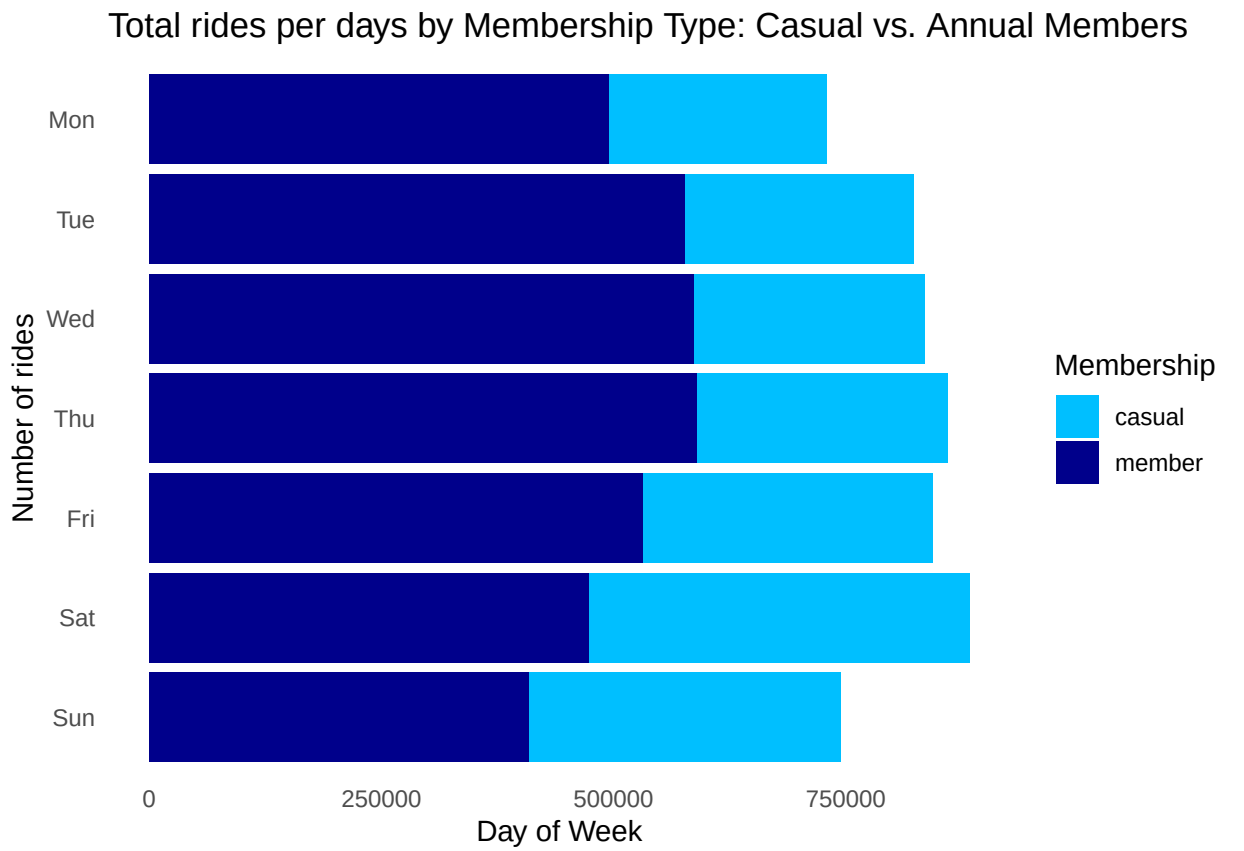

```
grpah_type_membership <- (graph3 + graph4) +
  (plot_layout(guides = 'collect') & theme(legend.position = "bottom"))
print(grpah_type_membership)
```



For annual members, there is no clear preferences for bike type, as the usage is evenly split between classic and electric bikes. However, for casual users, there is a slight preference for electric bikes over classic bikes, though the difference is not significant.

The next part of the analysis is the distrbution of total rides per day bu membership type.

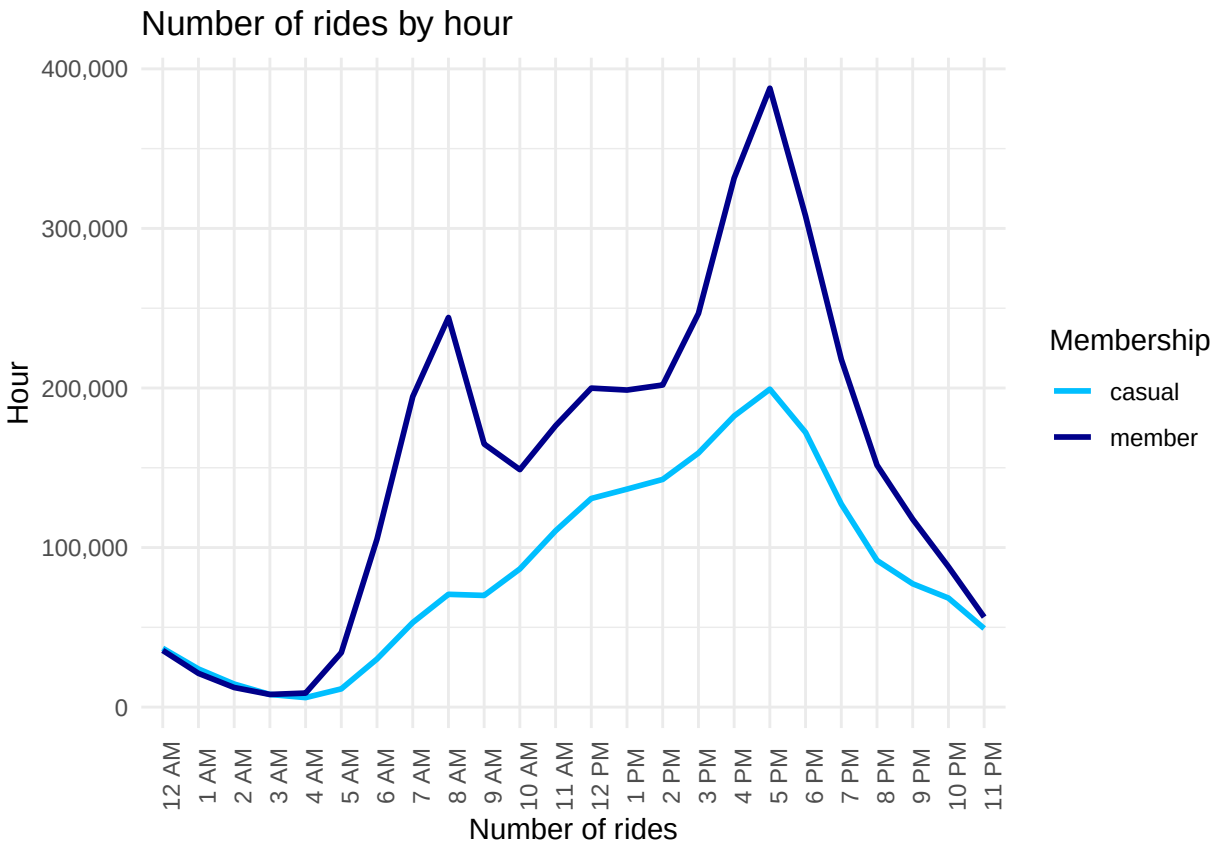
```
colors <- c("classic_bike" = "cyan3", "docked_bike" = "darkslategray1",
            "electric_bike"="deepskyblue2", "member" = "darkblue", "casual"="deepskyblue")
labels_biketype <- c("classic_bike" = "classic bike", "docked_bike" = "docked bike",
                     "electric_bike"="electric bike", "1" = "casual", "2" = "member")

ggplot(count_data, aes(fill=member_casual, y=count , x=day_of_week )) +
  geom_bar(position="stack", stat="identity") +
  scale_fill_manual(values = colors) +
  labs(y = "Day of Week", x = "Number of rides",
       title = "Total rides per days by Membership Type: Casual vs. Annual Members", fill = "Membership
  theme_minimal() +
  coord_flip() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



In general, the highest number of rides occurs on Saturday. However, members tend to use the bikes more frequently from Tuesday to Thursday while casual users have a higher number of rides on weekends. The lowest number of rides overall is on Monday. However, for members it is on Sunday, while for casual users it is also on Monday.

The line graph below illustraties the distribution of rides throughout the day for different types of members.

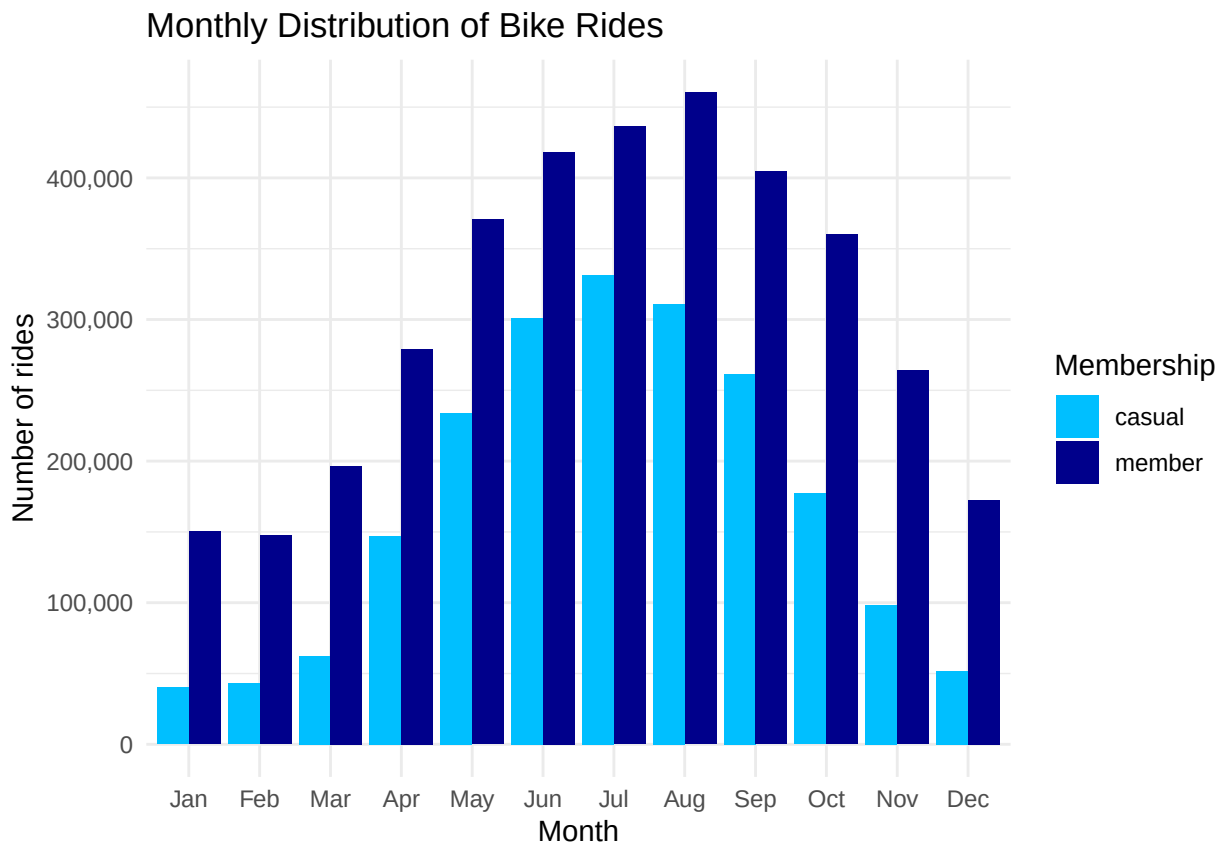
```
ggplot(hour_data, aes(x=hour, y=count, color=member_casual, group = member_casual )) +
  geom_line(size=1) +
  scale_color_manual(values = colors) +
  scale_y_continuous(labels = comma) +
  labs(color = "Membership", title="Number of rides by hour", x="Number of rides", y="Hour") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```



The graph shows that bike usage reached its lowest point between 3 and 4 AM. It then gradually increases, peaking at 8 am for members and at 5 PM for both members and casual users. After this peak, usage sharply declines.

Here, the monthly distribution of bike rides for both annual members and casual users is shown:

```
ggplot(monthly, aes(fill=member_casual, y=count, x=month)) +
  geom_bar(position="dodge", stat="identity") +
  scale_fill_manual(values = colors) +
  scale_y_continuous(labels = comma) +
  labs(y = "Number of rides", x = "Month", title = "Monthly Distribution of Bike Rides",
       fill = "Membership") +
  theme_minimal()
```

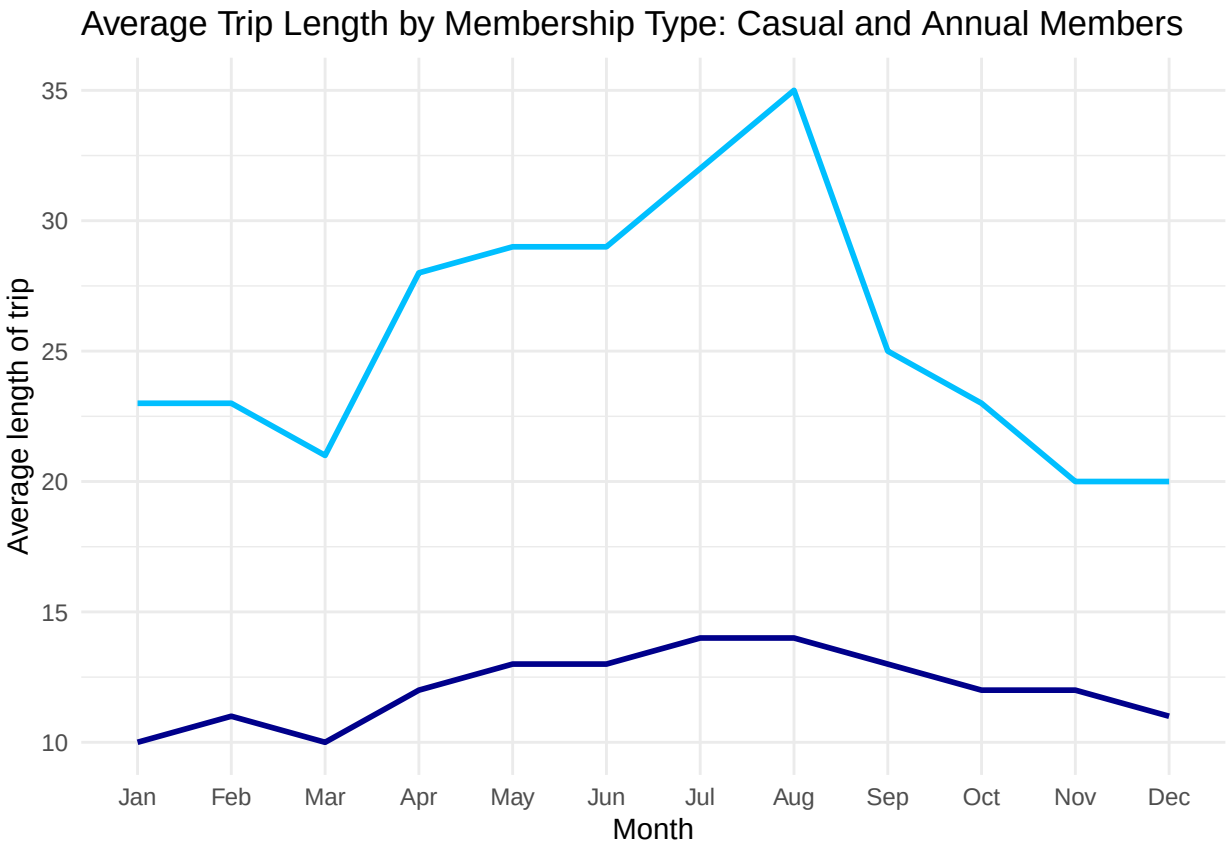


From the graph, it is evident that the most popular month for bike usage is August for annual members and July for casual users, while the least popular months are February for annual members and January for casual users.

The next graph shows the distribution of the average trip length for casual usera and annual members throughout the year:

```
ggplot() +
  geom_line(data = member_average, aes(x = month, y = mean_ride_length, group=2),
            color="darkblue", size=1) +
  geom_line(data = casual_average, aes(x = month, y = mean_ride_length, group = 3),
            color="deepskyblue", size = 1) +
  labs(y = "Average length of trip", x = "Month",
       title = "Average Trip Length by Membership Type: Casual and Annual Members",
       fill = "Membership") +
  theme_minimal()
```

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



From the graph, it can be seen that casual users have the shortest average trip length between November and December with a peak in August following some fluctuation, For annual members there are slight dips in January and March, but overall the average trip length is more stable. Casual users have a higher average trip length than members with the highest average length being about 35 minutes for casual users and 13 minutes for members.

The finally graph presents a map of the top stations in Chicago:

```
library(leaflet)
library(sf)
library(viridis)

stations_sf <- st_as_sf(stations_top, coords = c("longitude", "latitude"), crs = 4326)
pal_cont <- colorFactor("viridis", levels = stations_sf$count)

pal <- colorNumeric(palette = viridis(20), domain = stations_top$count)

legend_entries <- paste0(
  '<div style="background-color: ', pal_cont(stations_top$count), '; height: 20px; width: 20px;
  display: inline-block;"></div> ',
  stations_top$station, '<br>'
)
legend_html <- paste0(
  '<div style="background-color: white; padding: 10px; border: 1px solid #ccc; width: 300px;">',
  '<b>Stations Legend</b><br>',
  paste(legend_entries, collapse = ''),
  '</div>'
```

```
)

stations_top$latitude <- as.numeric(stations_top$latitude)
stations_top$longitude <- as.numeric(stations_top$longitude)

leaflet_map <- leaflet() %>%
  addTiles() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  setView(lng = -87.63, lat = 41.87, zoom = 12) %>%
  addCircleMarkers(data = stations_top, color = ~pal_cont(count), radius = ~sqrt(count) * 0.01, fillOpa
  addControl(
    html = legend_html,
    position = "topright"
  )
```

```
## Assuming "longitude" and "latitude" are longitude and latitude, respectively
```

Analysis and Insights

Here is the detailed analysis based on the trip data for the year 2023 ## Overall trends

Overall, **casual users have a higher average trip duration of 28 minutes**, with a **maximum recorded duration of 68 days**—likely an error due to malfunctioning stations, as many users have reported issues and incorrect charges despite returning bicycles on time. **Annual subscribers** have a maximum trip duration of just over one day, **with an average trip length of 13 minutes**.

Annual members make up the majority, **approximately 63% of users**. Casual users have a trip duration about 1.86 times longer than annual subscribers. Among bicycle types, **electric bikes are slightly preferred overall**, while classic bikes accumulate more total usage hours. Annual subscribers do not show a strong preference for bike type, whereas **casual users prefer more electric bikes**.

Monthly and Seasonal Trends

Trips peak in August, slightly surpassing July, with a notable increase from March through August. Winter months, particularly January and February, show the lowest usage levels, with trips **rising by about 75.6% in summer compared to winter**. This pattern is likely influenced by weather conditions such as snow, ice, and rain, which lead to a decline in trips during colder months.

Trip duration peaks in July and August, roughly doubling compared to autumn.**The average trip length drops to about half of its summer duration by December** and declines further in March. While weather conditions likely contribute to this decline, the sharp reduction in March, despite stable trip numbers, suggests that changes in pricing policies could also be a factor.

Membership trends

Annual subscribers use bicycles more consistently, with an **average trip duration of 13 minutes, about half of casual users' average duration**. During summer, their trip duration extends by approximately 7.7% to 14 minutes, while it drops by around 23.1% to 10 minutes in winter. Both user types see a decline in trip duration in March. For casual users, trip duration starts at about 21.5 minutes in March, increasing by 62% to 35 minutes from March to August, then falling to 20 minutes by November.

Annual subscribers also make about 3.75 times more trips during winter months compared to casual users. Casual users experience a significant drop in bike usage in January and February, with

usage rising in April and peaking in July. Even at their peak, casual users make about 1.3 times fewer trips than yearly subscribers. **Annual subscribers reach their peak in August** with over 450,000 trips, approximately **three times the number of winter trips**, before declining again in September.

Usage by Time of Day

Bicycle usage is minimal during the night throughout the workweek, with peak activity occurring at 8 AM and 5 PM. **The highest usage is observed between Tuesday and Thursday at 5 PM**, coinciding with the typical commute times for work and school, and then gradually declines, reaching a minimum around midnight.

On weekends, particularly Saturdays, the pattern differs. Usage starts to rise from 8 AM, peaks between 12 PM and 4 PM, and then declines more slowly than on weekdays. This indicates that bicycles **are used more consistently on Saturdays**, with a stable pattern throughout the day. Sundays also see significant usage during the daytime.

Overall, bicycles are predominantly used for commuting to work or school during weekdays, while on weekends, they are more commonly used for leisure activities, such as visiting parks.

By user type, notable trends emerge: **annual subscribers exhibit two** distinct **peaks** in bicycle usage, **at 8 AM and 5 PM**. In contrast, **casual users** experience a significant increase starting at 9 AM, **reaching a peak at 5 PM**. This suggests that annual members primarily use bicycles for go to and from work or school, whereas casual users are more likely to use bicycles for leisure activities or for traveling home in the evening. Casual users may also rely more on other modes of transportation in the mornings.

Weekly Trends

As previously mentioned, **the highest number of trips occurs on Saturdays**, while the fewest trips are on Mondays. Annual subscribers use bicycles the most between Tuesday and Thursday, with the lowest usage on Sundays. In contrast, casual users bike the most on weekends and the least during weekdays. This indicates that **annual subscribers primarily use bicycles to get to work or school, while casual users use them more for leisure activities.**

Top Station

The most popular station, **Streeter Dr & Grand Ave**, some stations were identified using KNN due to missing station name data. In addition, **DuSable Lake & Shore Dr and Monroe St,DuSable Lake & Shore Dr and North Blvd, Michigan Ave & Oak St are** also too popular stations and are located close to each other within the same area.

Recomendation

In my opinion, to make bike-share more attractive and user-friendly, it is helpful to consider the practices of European cities. European cities that successfully promote bike use often offer short-term and not only long-term subscriptions. Analysis shows that occasional users are put off by the idea of a year-long plan. Since the service currently only offers single rides, one-day passes and annual subscriptions, introducing new type of subscriptions could attract more users and boost profits without significant extra cost:

1. Membership: - Short-Term Memberships: Introduce flexible options like a 3-day pass, 10-ride pass, monthly pass, or a three-month pass with a limit on the length of rides. This tiered approach caters to users who may not want to commit to an annual membership but are more likely to use the service regularly. It can serve as a bridge to long-term membership by offering them a taste of the benefits, encouraging them to eventually upgrade to an annual plan.

- **Trial Memberships:** Offer a trial period for annual memberships at a reduced rate, allowing casual users to experience the benefits without committing fully at first.
- **Corporate Partnerships:** Collaborate with companies to offer discounted annual memberships as part of corporate programs.

-**Incentives for Seasonal Peaks:** Offer limited-time discounts or promotions during the peak months (April to August), when casual users are most active. For instance, provide discounts on annual memberships during these periods.

2. Implement a Reward System:

-**Loyalty Programs:**Introduce a points-based loyalty program where casual users earn points for each ride, which can later be redeemed for discounts on an annual subscription. Also, it necessary to encourage more rides during off-peak hours or providing discounts during periods of low usage

-**Targeted Discount Program:**To attract a broader range of users and enhance the accessibility of our bike share program, I recommend implementing discounts for specific groups, including students, individuals under 18, seniors over 55, and unemployed individuals. These targeted discounts will make the service more appealing and affordable for these key demographics, potentially increasing overall user engagement. Additionally, it may be beneficial to consider introducing a discount on public holidays to further incentive usage.

3. Also is necessary **resolve issues highlighted by current users such as:**

- **Docked stations:**resolve and prevent problems with docking stations to ensure they function correctly and do not cause issues with bike returns. -**Customer Service:**improve customer service by providing high-quality support to effectively resolve user issues.
- **Mobile App Improvements:** Simplify the bike rental process through the app, making it more user-friendly. Introduce features like quick rebooking, easier route planning, and real-time bike availability updates

4. Targeted Advertising: Focus advertising efforts on the top 20 most-used stations to maximize reach and impact.

Conclustions

In conclusion, bike usage peaks on weekends and during the spring and summer months. Most users are annual subscribers, but occasional users tend to use bikes for longer periods. It is essential to address current issues with docking stations and customer service, as these problems have led to user dissatisfaction and loss.

Introducing new subscription options, such as three-day passes, ten-ride packages, and monthly subscriptions, would likely increase revenue. These options are more cost-effective for the company compared to single rides or daily passes. Additionally, short-term subscriptions could attract occasional users, providing them with a better experience and potentially encouraging some to switch to annual subscriptions if their experience is positive and issues are resolved.

License

References

- Tripadvisor Divvy Bikes (https://www.tripadvisor.com/Attraction_Review-g35805-d5074715-Reviews-or10-Divvy_Bikes-Chicago_Illinois.html)
- Valenbici (https://www.valenbisi.es/en/offers/groups)
- Donkey Republic (https://www.donkey.bike/blog-how-much-does-it-cost-to-rent-a-donkey-republic-bike/)