Bike-Share Navigate Speedy Success?

Jason D.

2023-01-15

Contents

1. Introduction

This is a capstone project from Google's Data Analyst curriculum program. A bike share program in Chicago would like to increase its future success which depends on maximizing the number of annual memberships in its program. The company has provided data to be analyzed to aid with the marketing team in finding strategies to convert casual riders to annual members.

2. Ask Guided Questions

2.1 Characters and teams:

- Cyclistic: A bike-share program that features more than 5,800 bicycles and 600 docking stations.
- Lily Moreno: The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.
- Cyclistic marketing analytics team: A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy.

- Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

2.2 Business task:

- Lily Mareno the director of marketing has assigned the question, how do annual members and casual riders use Cyclistic bikes differently?

2.3 Data:

- The data to be analyzed has been sourced from: Index of bucket "divvy-tripdata": Data set. The data is available by Motivate International Inc. under this license. This is public data for the purpose of exploring how different customer types are using Cyclistic bikes. Regarding data-privacy issues the historical purchases using personal payment information and fare not provided.

3. Preparing Data for Exploration

The last 12 months of data (November 2021- October 2022) will be analyzed for the purpose of this project. Data comes in 12 cvs files for corresponding to the last 12 months. All 12 files take a lot of memory to handel so R programming will be use to complete this project. Regarding the quality of this data, it is considered data that has been generated by Lyft Bikes and Scooters, LLC ("Bikeshare") that operates in the City of Chicago's ("City") Divvy bicycle sharing service.

4. Process Data from Dirty to Clean

Please refer to the Appendix regarding the R programing code used to accomplish the following task:

4.1 Download the Librarys used to work with this data.

4.2 Download the raw data Data set and combine into one table.

Cleaning Data

4.3 Clean up column names to proper formatting

4.4 To aide in future analysis create new columns for year, month , day , week , y/m/d , and h:m:s by adding extra columns to the data set. The codes used was created with the help of Isabel Peels project as referance .

4.5 Check to make sure all numbers are formatted correctly.

4.6 Find the ride length and create a new column in the data set. Delete any rows with ride_length less then zero because anything else does not make sense.

4.7 Delete any rows with NA and "tes"t" data.

5. Analyses

Based on the statistical summery, members totaled 2,465,636 rides compared to casual with 1,652,826 total rides in the last 12 months from November 2021 to October 2022. In contrast casual riders have a longer ride lengths than member riders.

Cyclistic bike use is popular during the summer months and decrease during fall and winter months for both casual and annual members. Casual members use Cyclistic bikes predominantly during the summer months peaking during the last week of May from 11:00:00 to 16:00:00. Compared to casual riders, member riders has a relatively steady amount of rider usage throughout the summer months during May through August.

5.1 Heat Map

Annual members tend to be most active during the beginning of the week peaking on Wednesday totaling at 388,275 trips. Inversely casual riders are most active during Saturday at 350,037 rides.

5.2 Bar Graph

Member riders use Cyclistic bike more than casual riders. Inversely casual rider's ride length is higher then member riders. This may be due to causal riders using Cyclistic bike for leisurely activity for longer period of time compared to members using the service for short commutes or running errands. Ride length is longer for the classic bike followed by electric and docked bikes with casual riders . Members riders do not use dock bikes at all and prefer classic bike or electric bikes.

5.3 Bar Graph Ride Length

Cyclistic bike use is very popular for casual and member riders around the "Loop" area of the city of Chicago, a prime business and tourist location.

5.4 Geographic Map

Popular location for Memeber bike rides:

Popular location for Casual bike Rides:

5.4.1 The Top 100 bike stations

5.4.2 The Top 10 bike stations for both casual and member riders

Casual riders out number member riders around the city center area due to the high tourist attraction and parks, reflecting bike use for leisurely activity. In contrast, member riders usage data reflect the act of commuting for work or business activity. The second popular area for Cyclistic bike use is north of the Loop around Wrigley Field.

6. Conclusion

How could your team and business apply your insights?

• Any promotions should be done on Saturday during peak hours of 1200 to 1600 hr. Specifically summer months late May through August to maximize the promotional message to casual bike share riders.

What next steps would you or your stakeholders take based on your findings?

Promotional campaign focused on casual riders. Communicate to casual riders that there could be incentives to becoming from member rider. • 1 month free or discounted price for 12 months subscription * Communicate to casual member who reach a ride length thrush hold that it may be beneficial economically to convert to annual

In summary, Cyclistic's data demonstrates that casual riders predominantly use bikes for leisure, especially on summer weekends, while annual members' usage suggests regular commuting, peaking midweek. Marketing strategies should thus target casual riders with summer promotions and consider membership incentives based on ride length. For annual members, ensuring bike availability during peak commuting times is crucial. One thing to note, although the analysis provides actionable insights, the absence of cost data presents a limitation and an opportunity for deeper understanding to refine promotional strategies further.

7. Appendix

```
knitr::opts_chunk$set(echo = FALSE)

# Library used:
library("tidyverse")
library("janitor")
library("skimr")
library("lubridate")
library("ggpubr")
library("data.table")
library("viridis")
library("leaflet")
library("htmlwidgets")
library("htmltools")


# 4.2 download Raw data
url <- "https://divvy-tripdata.s3.amazonaws.com/202111-divvy-tripdata.zip"
download.file(url, "temp_data.zip")
unzip("temp_data.zip")
df1 <- read.csv("202111-divvy-tripdata.csv")

month2_url <- "https://divvy-tripdata.s3.amazonaws.com/202112-divvy-tripdata.zip"
download.file(month2_url, "temp_data.zip")
unzip("temp_data.zip")
df2 <- read.csv("202112-divvy-tripdata.csv")

month3_url <- "https://divvy-tripdata.s3.amazonaws.com/202201-divvy-tripdata.zip.zip"
download.file(month3_url, "temp_data.zip")
unzip("temp_data.zip")
df3 <- read.csv("202201-divvy-tripdata.csv")

month4_url <- "https://divvy-tripdata.s3.amazonaws.com/202202-divvy-tripdata.zip.zip"
download.file(month4_url, "temp_data.zip")
unzip("temp_data.zip")
df4 <- read.csv("202202-divvy-tripdata.csv")

month5_url <- "https://divvy-tripdata.s3.amazonaws.com/202203-divvy-tripdata.zip"
download.file(month5_url, "temp_data.zip")
unzip("temp_data.zip")
df5 <- read.csv("202203-divvy-tripdata.csv")

month6_url <- "https://divvy-tripdata.s3.amazonaws.com/202204-divvy-tripdata.zip"
download.file(month6_url, "temp_data.zip")
unzip("temp_data.zip")
df6 <- read.csv("202204-divvy-tripdata.csv")

month7_url <- "https://divvy-tripdata.s3.amazonaws.com/202205-divvy-tripdata.zip"
download.file(month7_url, "temp_data.zip")
unzip("temp_data.zip")
df7 <- read.csv("202205-divvy-tripdata.csv")
```

```r
month8_url <- "https://divvy-tripdata.s3.amazonaws.com/202206-divvy-tripdata.zip"
download.file(month8_url, "temp_data.zip")
unzip("temp_data.zip")
df8<- read.csv("202206-divvy-tripdata.csv")

month9_url <- "https://divvy-tripdata.s3.amazonaws.com/202207-divvy-tripdata.zip"
download.file(month9_url, "temp_data.zip")
unzip("temp_data.zip")
df9 <- read.csv("202207-divvy-tripdata.csv")

month10_url <- "https://divvy-tripdata.s3.amazonaws.com/202208-divvy-tripdata.zip"
download.file(month10_url, "temp_data.zip")
unzip("temp_data.zip")
df10 <- read.csv("202208-divvy-tripdata.csv")

month11_url <- "https://divvy-tripdata.s3.amazonaws.com/202209-divvy-tripdata.zip"
download.file(month11_url, "temp_data.zip")
unzip("temp_data.zip")
df11 <- read.csv("202209-divvy-tripdata.csv")

month12_url <- "https://divvy-tripdata.s3.amazonaws.com/202210-divvy-tripdata.zip"
download.file(month12_url, "temp_data.zip")
unzip("temp_data.zip")
df12 <- read.csv("202210-divvy-tripdata.csv")

#We start working with the data by first combining the last 12 months of data into one file. Then check


# Combine all 12 data file into one data set
bike_rides <- rbind(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)

# produce an overview of the data
str(bike_rides)

skim_without_charts(bike_rides)

glimpse(bike_rides)


# 4.3 make all column names text with proper formatting
clean_names(bike_rides)


# check if columns with dates are formatted as dates
sapply(bike_rides, is.po)


# results shows that all columns formatted as dates are true



# 4.4 Create column data for year, month , day , week ,  y/m/d , & h:m:s  .
bike_rides$year <- format(
```

```
  bike_rides$started_at,
  "%Y"
)

# Month
bike_rides$month <- format(
  bike_rides$started_at,
  "%m"
)


  # Week
bike_rides$week <- format(
  bike_rides$started_at,
  "%W"
)

# Day
bike_rides$day <- format(
  bike_rides$started_at,
  "%d"
)

# Day of week
bike_rides$day_of_week <- format(
  bike_rides$started_at,
  "%A"
)

# Date, YYYY-MM-DD
bike_rides$YMD <- format(
  bike_rides$started_at,
  "%Y-%m-%d"
)

# Time of Day, HH:MM:SS
bike_rides$ToD <- format(
  bike_rides$started_at,
  "%H:%M:%S"
)

# 4.5 the following checks all columns if numeric:
sapply(bike_rides ,is.numeric)

# Order by date
bike_rides <- bike_rides %>%
  arrange(started_at)

# 4.6 Find the ride length and create a new column in the data frame.Delete any rows with ride_length l
bike_rides$ride_length <- difftime(
  bike_rides$ended_at,
  bike_rides$started_at,
  units = "secs"
```

```
)

glimpse(bike_rides)
# By checking the data with glimpse, it looks like the ride length is time base, change to numeric for
#ride length is less then zero, which is cause for deletion
bike_rides$ride_length <- as.numeric(
  as.character(bike_rides$ride_length))

# to keep original data, we create a new cleaned data set named "bike_rides_cleaned".
bike_rides_cleaned <- bike_rides %>%
  filter(!(ride_length < 0))

check1 <- bike_rides_cleaned %>%
  arrange(ride_length)

# 4.7 Delete any rows with NA
bike_rides_cleaned <- na.omit(bike_rides_cleaned)


#There are some station names which are capitalized. Identify the station and delete that particular bi

capitalized_station_name_check <- bike_rides_cleaned %>%

  filter(
    str_detect(start_station_name, "[:upper:]")
    & !str_detect(start_station_name,"[:lower:]")
  ) %>%

  group_by(
    start_station_name
  ) %>%

  count(
    start_station_name
  )

# Remove capitalized station name
bike_rides_cleaned <- bike_rides_cleaned %>%
  filter(
    !(str_detect(start_station_name, "[:upper:]")
      & !str_detect(start_station_name, "[:lower:]"))
  )


# Statistical Summery

SELECT
    member_casual,
    COUNT(*) as total_trips,
    AVG(ride_length_min) as avg_ride_length_mins,
    SUM(ride_length_min) / 60 as total_bike_ride_length_hr,
    SUM(CASE WHEN rideable_type = 'electric_bike' THEN ride_length_min ELSE 0 END) / 60 as total_electri
    SUM(CASE WHEN rideable_type = 'docked_bike' THEN ride_length_min ELSE 0 END) / 60 as total_docked_bi
```

```
    SUM(CASE WHEN rideable_type = 'classic_bike' THEN ride_length_min ELSE 0 END) / 60 as total_classic_
FROM bike_ride_clean
WHERE start_station_name IS NOT NULL AND end_station_name IS NOT NULL -- Exclude rows with null values
GROUP BY member_casual;
#5.1 Heat Map
# Create data frame that summarize the number of trips by date to
   #find when is the most popular time bike share is used
heat_map_data1 <- bike_rides_cleaned %>%

  select(
    YMD,
    started_at,
    week,
    year,
    member_casual
  ) %>%
  filter(member_casual=="member")

#create a column for hour
heat_map_data1$hour <- format(
  heat_map_data1$started_at,
  "%H"
)

#create a column for week
heat_map_data1$week <-
  as.numeric(heat_map_data1$week)


heat_map_data1 <-heat_map_data1 %>%
  group_by(
    YMD,hour
  ) %>%

  mutate(
    numtrips = n()
  ) %>%

  distinct(
    YMD,
    .keep_all = TRUE
  )



# Arrange hours in order for y-axis
heat_map_data1$hour<- ordered(
  heat_map_data1$hour,
  levels = c(
    "00","01", "02", "03", "04",
    "05", "06", "07","08","09","10","11","12","13","14","15","16",
    "17","18","19","20","21","22","23", "24"
  )
```

```
)

# Create a heat map to show most popular time of year
hhp3 <- ggplot(
  heat_map_data1,
  aes(
    x = week,
    y = hour,
    fill = numtrips
  )
) +

  # Use the viridis colour scheme to show the popularity of each day
  scale_fill_viridis(
    option = "D",
    direction = 1,
    name = "Number of Trips"
  ) +

  # Create a rectangular heat map
  geom_tile(
    colour = "white",
    na.rm = FALSE
  ) +

  # Separate the heat maps by year
  facet_wrap(
    "year",
    ncol = 1
  ) +

  # Reverse the y-axis so that the hour days read vertically Monday to Sunday
  scale_y_discrete(
    limits = rev
  ) +

  # Add x-axis labels to show the months of the year
  scale_x_continuous(
    expand = c(0, 0),
    breaks = seq(1, 52, length = 12),
    labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
               "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
  ) +

  # Set the light theme
  theme_light() +

# Add a title
labs(title = "Member Riders" , y= "Hour of the Day" ,x="Months")



# Popular time for casual members to use bike share
```

```r
# Create data frame that summaries the number of trips by date
heat_map_data2 <- bike_rides_cleaned %>%

  select(
    YMD,
    started_at,
    week,
    year,
    member_casual
  ) %>%
  filter(member_casual=="casual")

#create a column for hour
heat_map_data2$hour <- format(
  heat_map_data2$started_at,
  "%H"
)

#create a column for week
heat_map_data2$week <-
  as.numeric(heat_map_data2$week)


# Organize the heat map and count for each bike ride
heat_map_data2 <-heat_map_data2 %>%
  group_by(
    YMD, hour
  ) %>%

  mutate(
    numtrips = n()
  ) %>%

  distinct(
    YMD,
    .keep_all = TRUE
  )



# Arrange hours in order for y-axis
heat_map_data2$hour<- ordered(
  heat_map_data2$hour,
  levels = c(
    "00","01", "02", "03", "04",
    "05", "06", "07","08","09","10","11","12","13","14","15","16",
    "17","18","19","20","21","22","23", "24"
  )
)

# Create a heat map to show most popular time of year
hhp4 <- ggplot(
  heat_map_data2,
```

```
  aes(
    x = week,
    y = hour,
    fill = numtrips
  )
) +

  # Use the viridis colour scheme to show the popularity of each day
  scale_fill_viridis(
    option = "D",
    direction = 1,
    name = "Number of Trips"
  ) +

  # Create a rectangular heat map
  geom_tile(
    colour = "white",
    na.rm = FALSE
  ) +

  # Separate the heat maps by year
  facet_wrap(
    "year",
    ncol = 1
  ) +

  # Reverse the y-axis so that the hourdays read vertically Monday to Sunday
  scale_y_discrete(
    limits = rev
  ) +

  # Add x-axis labels to show the months of the year
  scale_x_continuous(
    expand = c(0, 0),
    breaks = seq(1, 52, length = 12),
    labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
               "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
  ) +

  # Set the light theme
  theme_light() +

  # Add a title
  labs(title = "Casual Riders" , y= "Hour of the Day" ,x="Months")



#Combine Casual and Annual Member heat map
hhp2a <- ggarrange(
  hhp3,
  hhp4,
  ncol = 1,
  nrow = 2,
```

```
  common.legend = TRUE,
  legend = "right"
)
#5.2 Bar Graph
#Finding the most popular day of the week to ride in the last 12 months.
week_day1 <- bike_rides_cleaned %>%

  select(
    day_of_week
  ) %>%
  group_by(day_of_week) %>%
  mutate(
    numtrips=n()
  ) %>%
  distinct(
    day_of_week,
    .keep_all = TRUE
  ) %>%
  arrange(
    day_of_week
  )

# Adding x- axis labels
week_day1$day_of_week<- ordered(
  week_day1$day_of_week,
  levels = c(
    "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"
  )
)

#Plotting the bar graph
Weekbar<- week_day1 %>%
  ggplot(aes(x = day_of_week, y = numtrips)) +
  geom_bar(stat = "identity",
           position = "dodge",
           aes(fill=numtrips))+
  labs(title = " Day of Week most Popular ", y="Total Bike Rides per day", x= "")


# Members:
member_weekday1a <- bike_rides_cleaned%>%
  filter(member_casual=="member") %>%
  select(
  day_of_week
) %>%
  group_by(day_of_week) %>%
  mutate(
    numtrips=n()
  ) %>%
  distinct(
    day_of_week,
    .keep_all = TRUE
  ) %>%
```

```
  arrange(
    day_of_week
  )

# Adding x- axis labels
member_weekday1a$day_of_week<- ordered(
  member_weekday1a$day_of_week,
  levels = c(
    "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"
  )
)

Weekbar1<- member_weekday1a %>%
  ggplot(aes(x = day_of_week, y = numtrips)) +
  geom_bar(stat = "identity",
           position = "dodge",
           aes(fill=numtrips))+
  labs(title = " Member's most Popular Day for Riding ", y="Total Bike Rides per day", x= "")


# Casual:
member_weekday1b <- bike_rides_cleaned%>%
  filter(member_casual=="casual") %>%
  select(
    day_of_week
  ) %>%
  group_by(day_of_week) %>%
  mutate(
    numtrips=n()
  ) %>%
  distinct(
    day_of_week,
    .keep_all = TRUE
  ) %>%
  arrange(
    day_of_week
  )

# Adding x- axis labels
member_weekday1b$day_of_week<- ordered(
  member_weekday1b$day_of_week,
  levels = c(
    "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"
  )
)

Weekbar2<- member_weekday1b %>%
  ggplot(aes(x = day_of_week, y = numtrips)) +
  geom_bar(stat = "identity",
           position = "dodge",
           aes(fill=numtrips))+
  labs(title = " Casual's most Popular Day for Riding ", y="Total Bike Rides per day", x= "")
```

```r
# combine


Weekday2 <- ggarrange(    #code to combine both graph
  Weekbar1,         #first graph
  Weekbar2,       #2nd graph
  ncol = 1,
  nrow = 2,
  common.legend = FALSE
  #legend = "right"
)

#5.3 Bar Graph Ride Length
# Member bike ride length and riding frequency throughout the year

#create data frame for members using classic bikes
bar1a <- bike_rides_cleaned %>%
  select(month,year, member_casual,ride_length_hrs,rideable_type) %>%
  filter(member_casual=="member") %>%
  filter(rideable_type == "classic_bike") %>%
  group_by(month,year)%>%
  summarise(Classic_bike=sum(ride_length_hrs),.groups = 'drop')

bar1a<- bar1a %>%
  rename(classic_bike=Classic_bike)

#create data frame for members using docked bikes
bar2a<- bike_rides_cleaned %>%
  select(month, year,  member_casual,ride_length_hrs,rideable_type) %>%
  filter(member_casual=="member") %>%
  filter(rideable_type == "docked_bike") %>%
  group_by(month,year)%>%
  summarise(dock_bike=sum(ride_length_hrs),.groups = 'drop')

#create data frame for members using electric bikes

bar3a <- bike_rides_cleaned %>%
  select(month, year, member_casual,ride_length_hrs,rideable_type) %>%
  filter(member_casual=="member") %>%
  filter(rideable_type == "electric_bike") %>%
  group_by(month,year)%>%
  summarise(electric_bike=sum(ride_length_hrs),.groups = 'drop')

# Join all data frames together. There are only 2 data frames since members do not use docked bikes
barjoined2<- left_join(bar1a, bar3a, by=c("month"="month","year"="year"))

#for plotting purpose, change year column to numeric
barjoined2$year <-as.numeric(barjoined2$year)

# To produce a correct bar graph transform data frame wide to long
barj2<-pivot_longer(barjoined2, 3:4)

# plot the data frame
```

```r
bar_final2<-barj2 %>%
  #pivot_longer(cols = -month) %>%
  ggplot(aes(x = month, y = value)) +
  geom_bar(stat = "identity",
           position = "dodge",
           aes(fill=name))+
  labs(title = " Members Bike Type Used ", y="Total Bike Ride Length (hr)")+
  coord_cartesian(ylim = c(0, 75000))+
  guides(fill=guide_legend("Bike Type"))+
  facet_wrap(~year)


# Combine both the member and casual bar graphs into one.

annual_casual_comp <- ggarrange(    #code to combine both graph
  bar_final1,        #first graph
  bar_final2,      #2nd graph
  ncol = 1,
  nrow = 2,
  common.legend = FALSE
  #legend = "right"
  )
# 5.4
# Create a data frame which groups number of trips by station name and includes latitude and longitude
map_data_mem <- bike_rides_cleaned %>%

  select(
    start_station_name,
    start_lat,
    start_lng,
    member_casual
  ) %>%
  filter(member_casual== "casual") %>%

  group_by(
    start_station_name
  ) %>%

  mutate(
    numtrips = n()
  ) %>%

  distinct(
    start_station_name,
    .keep_all = TRUE
  )

#data frame member:
map_data_mem2 <- bike_rides_cleaned %>%

  select(
    start_station_name,
    start_lat,
```

```
    start_lng,
    member_casual
  ) %>%
  filter(member_casual== "member") %>%

  group_by(
    start_station_name
  ) %>%

  mutate(
    numtrips = n()
  ) %>%

  distinct(
    start_station_name,
    .keep_all = TRUE
  )
#arrange by number of trips  in descending order
map_data_mem <- map_data_mem %>%
  arrange(desc(numtrips))

map_data_mem2 <- map_data_mem2 %>%
  arrange(desc(numtrips))

#merge map data together
map_data2 <- rbind(map_data_mem,map_data_mem2)

#Legend labels
groups_all2<- c( "member", "casual")  # this is the text that will be in the legend

# Create a sequence of values which will act as the key shown on the leaflet map to group stations whic
mybins2 <- seq(0, 70000, by = 5000)


## viridis colour palette to visually show how popular a station is
mypalette2 <- colorBin(
  palette ="viridis",
  domain = map_data2$numtrips,
  na.color = "transparent",
  bins = mybins2
)


# Prepare text to be used in a tooltip so that users can interact with the coloured markers on the map
mytext2 <- paste(
  "Station name: ", map_data2$start_station_name, "<br/>",
  "Number of trips: ", map_data2$numtrips, sep = ""
) %>%
  lapply(htmltools::HTML)

#plot the data
foo <-leaflet(map_data2) %>%
  addTiles() %>%
```

```
  # Set coordinates over the city of Chicago
  setView(
    lng = -87.6298, lat = 41.8781, zoom = 11.5
  ) %>%
  addProviderTiles("Esri.WorldGrayCanvas") %>%
  addCircleMarkers(
    lng = ~start_lng,
    lat = ~start_lat,
    label = mytext2,
    fillColor = ~ mypalette2(numtrips),
    fillOpacity = 0.7,
    color = "white",
    radius = 5,
    stroke= FALSE,
    group = ~paste(member_casual),
    labelOptions = labelOptions(
      style = list(
        "font-weight" = "normal",
        padding = "3px 8px"
      ),
      textsize = "13px",
      direction = "auto"
    )

  )%>%
  #this is the layer where user can control the categorizes that can be toggled on or off
  addLayersControl(
    overlayGroups = groups_all2,
    options = layersControlOptions(collapsed = FALSE)
  ) %>%

  # Add a legend
  addLegend(
    pal = mypalette2,
    values = ~ numtrips,
    opacity = 0.9,
    title = "Number of trips",
    position = "bottomright"
  )%>%
  hideGroup("casual")

# 5.4.1 The Top 100 bike stations for both casual and member riders
# Create a data frame which groups number of trips by station name and includes latitude and longitude
map_data <- bike_rides_cleaned %>%

  select(
    start_station_name,
    start_lat,
    start_lng
  ) %>%

  group_by(
    start_station_name
```

```
  ) %>%

  # counts how many a bike share  is used in a group
  mutate(
    numtrips = n()
  ) %>%

  distinct(
    start_station_name,
    .keep_all = TRUE
  )



# Create a sequence of values which will act as the key shown on the leaflet map to group stations whic
mybins <- seq(0, 70000, by = 10000)

# Assign the viridis colour palette to visually show how popular a station is
mypalette <- colorBin(
  palette ="viridis",
  domain = map_data$numtrips,
  na.color = "transparent",
  bins = mybins
)

# Prepare text to be used in a tooltip so that users can interact with the coloured markers on the map
mytext <- paste(
  "Station name: ", map_data$start_station_name, "<br/>",
  "Number of trips: ", map_data$numtrips, sep = ""
) %>%
  lapply(htmltools::HTML)

# top 100 stations used in a year:
map_data_100 <- map_data %>%
  arrange((desc(numtrips)))

#Show only top 100 of data
  map_data_100<- head(map_data_100,100)


#Plotting the top 100 bike stations
p2<- leaflet(map_data_100) %>%
  addTiles() %>%

  # Set coordinates over the city of Chicago
  setView(
    lng = -87.6298, lat = 41.8781, zoom = 11.5
  ) %>%

  # Set map style
  addProviderTiles("Esri.WorldGrayCanvas") %>%

  # Add circle markers to represent each station
```

```
  # & add a fill colour to show the popularity of each station
  # & add an interactive tooltip for detail
  addCircleMarkers(
    ~ start_lng, ~ start_lat,
    fillColor = ~ mypalette(numtrips),
    fillOpacity = 0.7,
    color = "white",
    radius = 8,
    stroke = FALSE,
    label = mytext,
    labelOptions = labelOptions(
      style = list(
        "font-weight" = "normal",
        padding = "3px 8px"
      ),
      textsize = "13px",
      direction = "auto"
    )
  ) %>%

  # Add a legend
  addLegend(
    pal = mypalette,
    values = ~ numtrips,
    opacity = 0.9,
    title = "Number of trips",
    position = "bottomright"
  )

#5.4.2 The Top 10 bike stations for both casual and member riders
map_data_10<- head(map_data_100,10)
# top 10 stations used in a year:
map_data_100 <- map_data %>%
  arrange((desc(numtrips)))

map_data_100<- head(map_data_100,100)

p3<- leaflet(map_data_10) %>%
  addTiles() %>%

  # Set coordinates over the city of Chicago
  setView(
    lng = -87.6298, lat = 41.8781, zoom = 11.5
  ) %>%

  # Set map style
  addProviderTiles("Esri.WorldGrayCanvas") %>%

  # Add circle markers to represent each station
  # & add a fill colour to show the popularity of each station
  # & add an interactive tooltip for detail
  addCircleMarkers(
    ~ start_lng, ~ start_lat,
```

```
  fillColor = ~ mypalette(numtrips),
  fillOpacity = 0.7,
  color = "white",
  radius = 8,
  stroke = FALSE,
  label = mytext,
  labelOptions = labelOptions(
    style = list(
      "font-weight" = "normal",
      padding = "3px 8px"
    ),
    textsize = "13px",
    direction = "auto"
  )
) %>%

# Add a legend
addLegend(
  pal = mypalette,
  values = ~ numtrips,
  opacity = 0.9,
  title = "Number of trips",
  position = "bottomright"
)
```