Scenario

You are a junior data analyst working on the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company’s future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

About the company

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic’s marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members. Cyclistic’s finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a solid opportunity to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs. Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

A clear statement of the business task

The business task is to design marketing strategies aimed at converting casual riders into annual members. To achieve this task, there need to be understanding on the differences between casual riders and member, hence this project

2. A description of all data sources used

1. The trips data of the Cyclistic which was made available by Motivate International Inc. and were downloaded from <https://divvy-tripdata.s3.amazonaws.com/index.html>.

2. The downloaded data in the various months’ zip folders were unzipped into a folder named Cyclistic\_unzipped .15 current datasets from 2022 to 2024 were selected into another folder named cyclistic

3. Documentation of any cleaning or manipulation of data

# Loading Packages for work.  
> library(“tidyverse”)

── **Attaching core tidyverse packages** ────────────────────────────────────── tidyverse 2.0.0 ──

✔ dplyr 1.1.4 ✔ readr 2.1.5

✔ forcats 1.0.0 ✔ stringr 1.5.1

✔ ggplot2 3.5.1 ✔ tibble 3.2.1

✔ lubridate 1.9.3 ✔ tidyr 1.3.1

✔ purrr 1.0.2

── **Conflicts** ──────────────────────────────────────────────────────── tidyverse\_conflicts() ──

✖ purrr::%||%() masks base::%||%()

✖ dplyr::filter() masks stats::filter()

✖ dplyr::lag() masks stats::lag()

ℹ Use the conflicted package to force all conflicts to become errors

> library(dplyr)

> library(lubridate)

> library("hms")

Attaching package: ‘hms’

The following object is masked from ‘package:lubridate’:

Hms

# Importing Data and combining them

> setwd("C:/Users/Daniel/Desktop/Cyclistic") # setting work directory

> fnames <- list.files() # listing files

> print(fnames)

[1] "202201-divvy-tripdata.csv" "202202-divvy-tripdata.csv" "202203-divvy-tripdata.csv"

[4] "202204-divvy-tripdata.csv" "202206-divvy-tripdata.csv" "202307-divvy-tripdata.csv"

[7] "202308-divvy-tripdata.csv" "202309-divvy-tripdata.csv" "202310-divvy-tripdata.csv"

[10] "202311-divvy-tripdata.csv" "202312-divvy-tripdata.csv" "202401-divvy-tripdata.csv"

[13] "202402-divvy-tripdata.csv" "202403-divvy-tripdata.csv" "202404-divvy-tripdata.csv"

> csv <- lapply(fnames, read.csv) # reading files

> Cyclistic\_data <- do.call(rbind,csv) # Combining Files

> view(Cyclistic\_data)

# Data Wrangling

> Cyclistic\_data$started\_at <- as.POSIXct(Cyclistic\_data$started\_at, format = "%Y-%m-%d %H:%M:%S") # converting string started\_at to POSIXct format

> Cyclistic\_data$ended\_at <- as.POSIXct(Cyclistic\_data$ended\_at, format = "%Y-%m-%d %H:%M:%S") # converting string ended\_at to POSIXct format

> View(Cyclistic\_data)

> Cyclistic\_data$ride\_length <- difftime(Cyclistic\_data$ended\_at,Cyclistic\_data$started\_at) # finding the difference in between started\_at and ended\_at

> Cyclistic\_data$ride\_length <- as.numeric(Cyclistic\_data$ride\_length)

# storing the data$ride\_length as numeric

> Cyclistic\_data$ride\_length <- as\_hms(Cyclistic\_data$ride\_length) # storing the answer as hour minute and second

# Cleaning Data

> Cyclistic\_data <- Cyclistic\_data [,- c(5,6,7,8,9,10,11,12)] # remove unwanted columns by coulmn index

> Cyclistic\_data <- na.omit(Cyclistic\_data) # remove empty cells

> Cyclistic\_data <- distinct(Cyclistic\_data) # select distinct rows

> Cyclistic\_data <- subset(Cyclistic\_data,ride\_length > 0) # select only rows with ride\_length greater than zero

# adding new columns

# convert started\_at to date

# convert date to text

# retrive the day of the week as numeric

# convert date to month

# convert date to year

Cyclistic\_data <- Cyclistic\_data %>% mutate(date = as.Date(started\_at),

day\_of\_week = format( date, "%A"),

day = wday(date), month = format( date, "%m"),

year = format( date, "%Y"))

Removing outliers below 3 stabdard deviatios and above 3 standard deviations of the mean

> Cyclistic\_data$seconds <- as.numeric(Cyclistic\_data$ride\_length)

> mean\_ride <- mean(Cyclistic\_data$seconds )

> print(mean\_ride)

[1] 1128.497

> std <- sd(Cyclistic\_data$seconds)

> Threshold <- 3 \* std

> outliers <-Cyclistic\_data$seconds[Cyclistic\_data$seconds >(mean\_ride+ Threshold)|Cyclistic\_data$seconds <(mean\_ride+ Threshold)]

> print(outliers)

Since ride durations do not follow a perfect normal distribution rule, it is not advisable to us the 3 std below and above the mean rule to remove outliers. The best method to use is th IQR

Not removing outliers will leave you with a maximum ride length of 1641:29:04

Cyclistic\_data$ride\_length\_numeric <- as.numeric(Cyclistic\_data$ride\_length)

# Calculate the first and third quartiles

Q1 <- quantile(Cyclistic\_data$ride\_length\_numeric, 0.25, na.rm = TRUE)

Q3 <- quantile(Cyclistic\_data$ride\_length\_numeric, 0.75, na.rm = TRUE)

# Calculate the IQR (Interquartile Range)

IQR = Q3 - Q1

# Define the lower and upper bounds to identify outliers

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Filter the DataFrame to remove outliers

Cyclistic\_data\_cleaned <- Cyclistic\_data[Cyclistic\_data$ride\_length\_numeric >= lower\_bound & Cyclistic\_data$ride\_length\_numeric <= upper\_bound, ]

THIS REDUCES ROWS FROM 6056186 TO 5607721

4. A summary of your analysis

Total Number of Cyclists

>Total\_number\_Users <- count(Cyclistic\_data)

n

1 6056186

Mode of Week

> library(DescTools)

> mode\_day\_week <- Mode(Cyclistic\_data$day\_of\_week)

> mode\_day\_week

[1] "Saturday"

attr(,"freq")

[1] 933677

Saturday being the mode of day means that it records the highest number of bike rides. Cyclistic can expand services, run promotions and organize community events all in the bid to capitalize on the number of riders and make profit

Total number of members and casual

> Total\_number\_Users\_pmt <- Cyclistic\_data %>% group\_by(member\_casual)%>% count()

> Total\_number\_Users\_pmt

# A tibble: 2 × 2

# Groups: member\_casual [2]

member\_casual n

*<chr>* *<int>*

1 casual 2141735

2 member 3914451

The highest number of rides came from members.

Mean of ride\_length

> mean\_ride\_lenght <- as\_hms(mean(Cyclistic\_data$ride\_length))

> mean\_ride\_lenght

00:18:30.402014

This shows that the typical ride duration of a user is 18 minutes, 30 seconds

Max of ride\_length

> max\_of\_ride\_lenght <- as\_hms(max(Cyclistic\_data$ride\_length))

> max\_of\_ride\_lenght

1641:29:04

# Maximum and Mean ride\_length by membership type

> Mm\_Mn\_of\_ride\_length\_mt <- Cyclistic\_data %>% group\_by(member\_casual) %>% summarise( mean = as\_hms(mean(ride\_length)), max = as\_hms(max(ride\_length)))

> Mm\_Mn\_of\_ride\_length\_mt

# A tibble: 2 × 3

member\_casual mean max

*<chr>* *<time>* *<time>*

1 casual 29'01.046615" 1641:29:04

2 member 12'45.353982" 25:59:56

# Average ride length of users by type of bike

Av\_ride\_length\_users\_tb <- Cyclistic\_data %>% group\_by(member\_casual,rideable\_type) %>% summarise( mean = as\_hms(mean(ride\_length)))

`summarise()` has grouped output by 'member\_casual'. You can override using the `.groups`

argument.

> Av\_ride\_length\_users\_tb

# A tibble: 5 × 3

# Groups: member\_casual [2]

member\_casual rideable\_type mean

*<chr>* *<chr>* *<time>*

1 casual classic\_bike 00:32:54.814899

2 casual docked\_bike 02:39:27.547238

3 casual electric\_bike 00:14:54.371702

4 member classic\_bike 00:14:11.689538

5 member electric\_bike 00:11:09.652798

#Average ride\_length for users by day of week

> Av\_ride\_length\_users\_dw <- Cyclistic\_data %>% group\_by(member\_casual,day\_of\_week) %>% summarise( mean = as\_hms(mean(ride\_length)))

`summarise()` has grouped output by 'member\_casual'. You can override using the `.groups`

argument.

> Av\_ride\_length\_users\_dw

# A tibble: 14 × 3

# Groups: member\_casual [2]

member\_casual day\_of\_week mean

*<chr>* *<chr>* *<time>*

1 casual Friday 27'42.461208"

2 casual Monday 28'34.698235"

3 casual Saturday 32'00.074501"

4 casual Sunday 33'50.792240"

5 casual Thursday 26'05.511689"

6 casual Tuesday 26'18.029221"

7 casual Wednesday 25'05.929885"

8 member Friday 12'29.722483"

9 member Monday 12'16.505640"

10 member Saturday 14'00.685026"

11 member Sunday 14'16.554497"

12 member Thursday 12'13.825664"

13 member Tuesday 12'21.681553"

14 member Wednesday 12'15.515211"

#Number of rides for users by day of the week

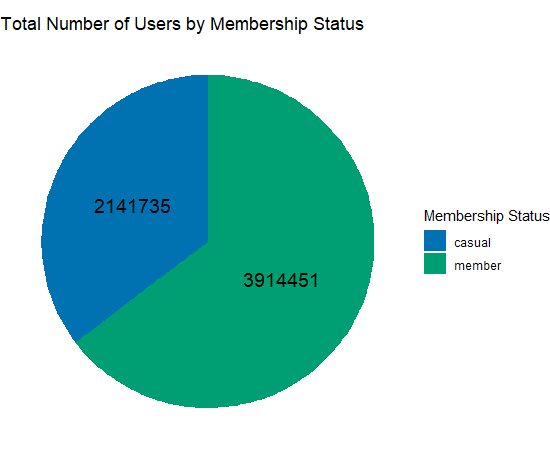
|  |
| --- |
| > No\_rides\_users\_dweek <- Cyclistic\_data %>% group\_by(member\_casual,day\_of\_week) %>% count()  > No\_rides\_users\_dweek  # A tibble: 14 × 3  # Groups: member\_casual, day\_of\_week [14]  member\_casual day\_of\_week n  *<chr>* *<chr>* *<int>*  1 casual Friday 305177  2 casual Monday 251825  3 casual Saturday 431619  4 casual Sunday 360272  5 casual Thursday 281415  6 casual Tuesday 258002  7 casual Wednesday 253425  8 member Friday 540139  9 member Monday 556010  10 member Saturday 502058  11 member Sunday 437678  12 member Thursday 631441  13 member Tuesday 627128  14 member Wednesday 619997 |
|  |
| |  | | --- | |  | |

5. Supporting visualizations and key findings

Visualixation 1

Total Number of Users by Membership Status

> ggplot(Total\_number\_Users\_pmt, aes(x = "", y = n, fill = member\_casual)) + geom\_bar(stat = "identity") + geom\_text(aes(label = n), position = position\_stack(vjust = 0.5), color = "black", size = 5) + labs(title = "Total Number of Users by Membership Status", x = NULL, y = NULL, fill = "Membership Status") + scale\_fill\_manual(values = c("#0072B2", "#009E73")) + coord\_polar("y", start = 0) + theme\_void()



Mean Ride Length by Membership Status

ggplot(Mm\_Mn\_of\_ride\_length\_mt, aes(x = member\_casual,fill = member\_casual, y = mean)) +

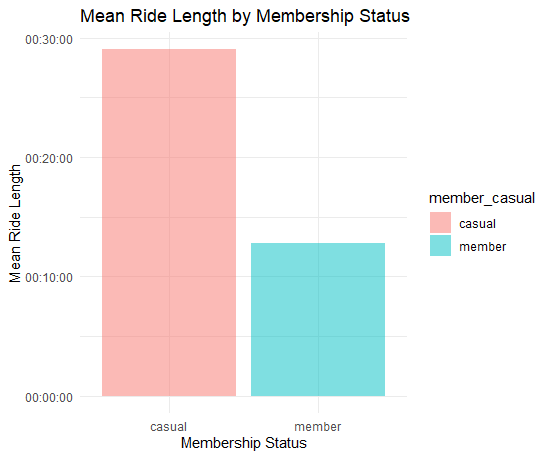
geom\_bar(stat = "identity", alpha = 0.5) +

labs(title = "Mean Ride Length by Membership Status",

x = "Membership Status",

y = "Mean Ride Length") +

theme\_minimal()



Max Ride Length by Membership Status

> ggplot(Mm\_Mn\_of\_ride\_length\_mt, aes(x = member\_casual,fill = member\_casual, y = as\_hms(max))) +

geom\_bar(stat = "identity", alpha = 0.5) +

geom\_text(aes(label = max), vjust = -0.5, color = "black", size = 4) +

labs(title = "Max Ride Length by Membership Status",

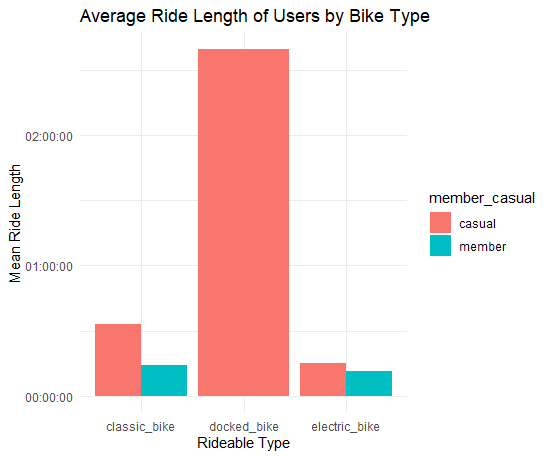
x = "Membership Status",

y = "Max Ride Length") +

theme\_minimal()

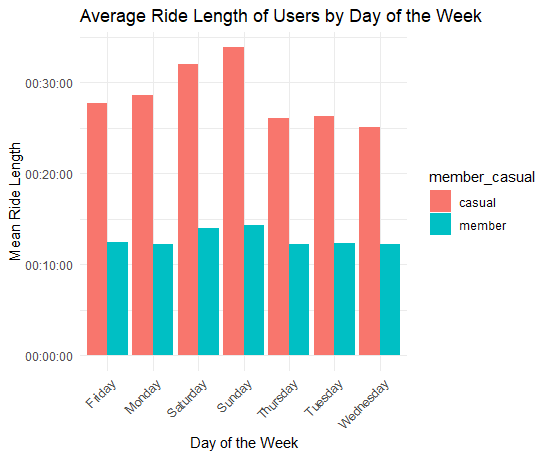
Average Ride Length of Users by Bike Type

ggplot(Av\_ride\_length\_users\_tb, aes(x = rideable\_type, fill = member\_casual, y = mean)) + geom\_col(position = "dodge") + labs(title = "Average Ride Length of Users by Bike Type", x = "Rideable Type", y = "Mean Ride Length") + theme\_minimal()

Average

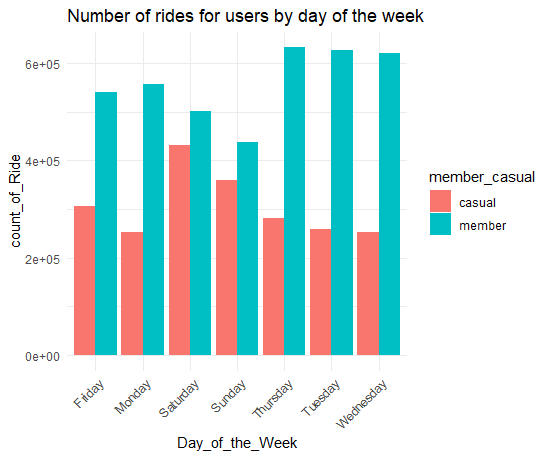
Ride Length of Users by Day of the Week

ggplot(Av\_ride\_length\_users\_dw, aes(x = day\_of\_week, fill = member\_casual, y = mean)) + geom\_col(position = "dodge") + labs(title = "Average Ride Length of Users by Day of the Week", x = "Day of the Week", y = "Mean Ride Length") + theme\_minimal() + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Adjust angle and justification of x-axis labels



Number of rides for users by day of the week

ggplot(No\_rides\_users\_dweek, aes(x = day\_of\_week, fill = member\_casual, y = n)) + geom\_col(position = "dodge") + labs(title = "Number of rides for users by day of the week", x = "Day\_of\_the\_Week", y = "count\_of\_Ride") + theme\_minimal() + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Adjust angle and justification of x-axis labels



Stacked graph Number of rides for users by day of the week

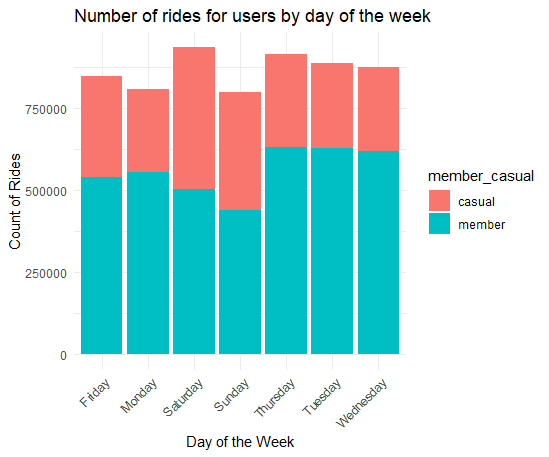
ggplot(No\_rides\_users\_dweek, aes(x = day\_of\_week, fill = member\_casual, y = n)) +

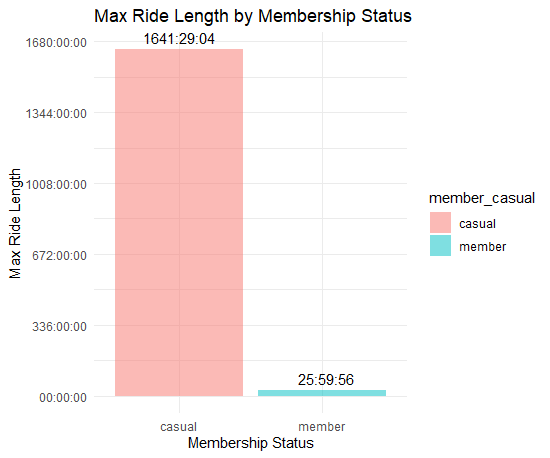
geom\_col(position = "stack") +

labs(title = "Number of rides for users by day of the week", x = "Day of the Week", y = "Count of Rides") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))





6. Your top three recommendations based on your analysis

WITHOUT OUTLIERS

> Total\_number\_Users <- count(Cyclistic\_data\_cleaned)

> Total\_number\_Users

# A tibble: 1 × 1

n

*<int>*

1 5607721

mode\_day\_week <- Mode(Cyclistic\_data\_cleaned$day\_of\_week)

> mode\_day\_week

[1] "Thursday"

attr(,"freq")

[1] 863432

Total number of members and casual

Total\_number\_Users\_pmt <- Cyclistic\_data\_cleaned %>% group\_by(member\_casual)%>% count()

> Total\_number\_Users\_pmt

# A tibble: 2 × 2

# Groups: member\_casual [2]

member\_casual n

*<chr>* *<int>*

1 casual 1841557

2 member 3766164

The highest number of rides came from members.

Mean of ride\_length

mean\_ride\_lenght <- as\_hms(mean(Cyclistic\_data\_cleaned$ride\_length))

> mean\_ride\_lenght

00:10:56.325638

Max of ride\_length

max\_of\_ride\_lenght <- as\_hms(max(Cyclistic\_data\_cleaned$ride\_length))

> max\_of\_ride\_lenght

00:34:46

Maximum and Mean ride\_length by membership type

Mm\_Mn\_of\_ride\_length\_mt <- Cyclistic\_data\_cleaned %>% group\_by(member\_casual) %>% summarise( mean = as\_hms(mean(ride\_length)), max = as\_hms(max(ride\_length)))

> Mm\_Mn\_of\_ride\_length\_mt

# A tibble: 2 × 3

member\_casual mean max

*<chr>* *<time>* *<time>*

1 casual 12'29.535607" 34'46"

2 member 10'10.748369" 34'46"

# Average ride length of users by type of bike

> Av\_ride\_length\_users\_tb <- Cyclistic\_data\_cleaned %>% group\_by(member\_casual,rideable\_type) %>% summarise( mean = as\_hms(mean(ride\_length)))

`summarise()` has grouped output by 'member\_casual'. You can override using the `.groups`

argument.

> Av\_ride\_length\_users\_tb

# A tibble: 5 × 3

# Groups: member\_casual [2]

member\_casual rideable\_type mean

*<chr>* *<chr>* *<time>*

1 casual classic\_bike 13'39.322306"

2 casual docked\_bike 18'33.778737"

3 casual electric\_bike 11'14.969103"

4 member classic\_bike 10'37.141094"

5 member electric\_bike 09'42.129297"

#Average ride\_length for users by day of week

> Av\_ride\_length\_users\_dw <- Cyclistic\_data\_cleaned %>% group\_by(member\_casual,day\_of\_week) %>% summarise( mean = as\_hms(mean(ride\_length)))

`summarise()` has grouped output by 'member\_casual'. You can override using the `.groups`

argument.

> Av\_ride\_length\_users\_dw

# A tibble: 14 × 3

# Groups: member\_casual [2]

member\_casual day\_of\_week mean

*<chr>* *<chr>* *<time>*

1 casual Friday 12'16.455523"

2 casual Monday 12'08.415864"

3 casual Saturday 13'28.623664"

4 casual Sunday 13'37.211941"

5 casual Thursday 11'43.617981"

6 casual Tuesday 11'48.624868"

7 casual Wednesday 11'36.942078"

8 member Friday 10'00.622227"

9 member Monday 09'54.119648"

10 member Saturday 10'49.822968"

11 member Sunday 10'48.887464"

12 member Thursday 09'59.470687"

13 member Tuesday 10'01.928370"

14 member Wednesday 09'57.624746"

#Number of rides for users by day of the week

No\_rides\_users\_dweek <- Cyclistic\_data\_cleaned %>% group\_by(member\_casual,day\_of\_week) %>% count()

> No\_rides\_users\_dweek

# A tibble: 14 × 3

# Groups: member\_casual, day\_of\_week [14]

member\_casual day\_of\_week n

*<chr>* *<chr>* *<int>*

1 casual Friday 267093

2 casual Monday 215900

3 casual Saturday 356995

4 casual Sunday 293369

5 casual Thursday 251760

6 casual Tuesday 228634

7 casual Wednesday 227806

8 member Friday 520707

9 member Monday 536925

10 member Saturday 476094

11 member Sunday 412687

12 member Thursday 611672

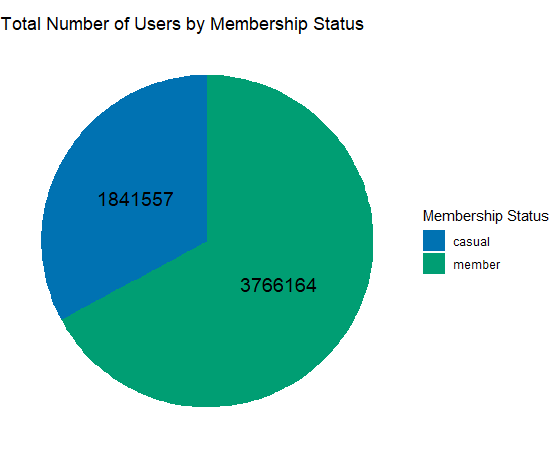
13 member Tuesday 606711

14 member Wednesday 601368

5. Supporting visualizations and key findings

Total Number of Users by Membership Status

> ggplot(Total\_number\_Users\_pmt, aes(x = "", y = n, fill = member\_casual)) + geom\_bar(stat = "identity") + geom\_text(aes(label = n), position = position\_stack(vjust = 0.5), color = "black", size = 5) + labs(title = "Total Number of Users by Membership Status", x = NULL, y = NULL, fill = "Membership Status") + scale\_fill\_manual(values = c("#0072B2", "#009E73")) + coord\_polar("y", start = 0) + theme\_void()



Mean Ride Length by Membership Status

ggplot(Mm\_Mn\_of\_ride\_length\_mt, aes(x = member\_casual,fill = member\_casual, y = mean)) +

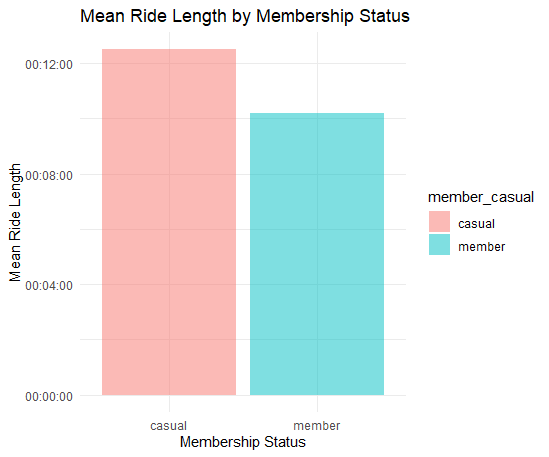
geom\_bar(stat = "identity", alpha = 0.5) +

labs(title = "Mean Ride Length by Membership Status",

x = "Membership Status",

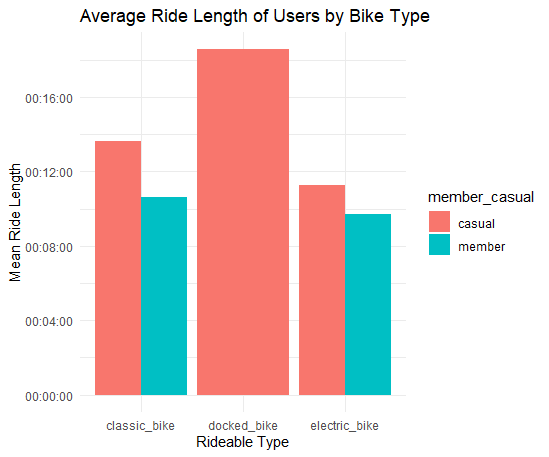
y = "Mean Ride Length") +

theme\_minimal()



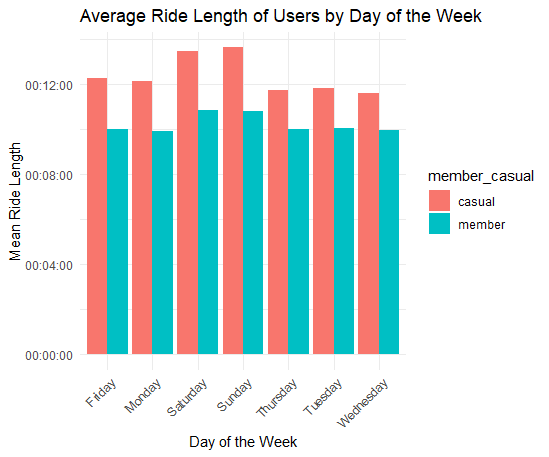
Average Ride Length of Users by Bike Type

ggplot(Av\_ride\_length\_users\_tb, aes(x = rideable\_type, fill = member\_casual, y = mean)) + geom\_col(position = "dodge") + labs(title = "Average Ride Length of Users by Bike Type", x = "Rideable Type", y = "Mean Ride Length") + theme\_minimal()



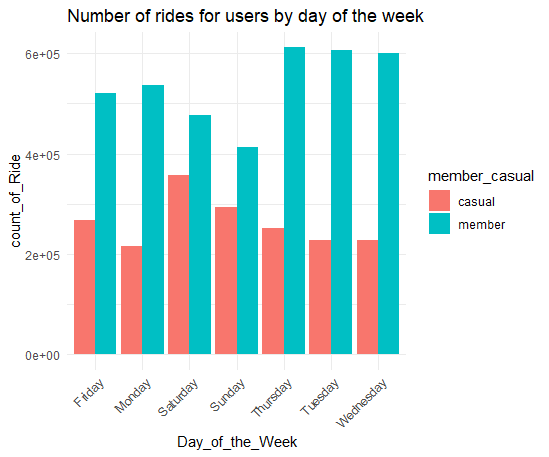
Ride Length of Users by Day of the Week

ggplot(Av\_ride\_length\_users\_dw, aes(x = day\_of\_week, fill = member\_casual, y = mean)) + geom\_col(position = "dodge") + labs(title = "Average Ride Length of Users by Day of the Week", x = "Day of the Week", y = "Mean Ride Length") + theme\_minimal() + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Adjust angle and justification of x-axis labels



Number of rides for users by day of the week

ggplot(No\_rides\_users\_dweek, aes(x = day\_of\_week, fill = member\_casual, y = n)) + geom\_col(position = "dodge") + labs(title = "Number of rides for users by day of the week", x = "Day\_of\_the\_Week", y = "count\_of\_Ride") + theme\_minimal() + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Adjust angle and justification of x-axis labels



Stacked graph Number of rides for users by day of the week

ggplot(No\_rides\_users\_dweek, aes(x = day\_of\_week, fill = member\_casual, y = n)) +

geom\_col(position = "stack") +

labs(title = "Number of rides for users by day of the week", x = "Day of the Week", y = "Count of Rides") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

