Cyclistic-Bike Share Project

Babalola Oluwatosin

2024-01-13

# Overview of Final Capstone Project – Google Data Analytics Professional Certification

I am currently engaged in a comprehensive analysis as part of my final capstone project within the Google Data Analytics Professional Certification. This project centers around the Cyclistic bike-share analysis, stimulating real-world tasks of a junior data analyst. Following a structured approach aligning with the data analysis process which are: ASK, PREPARE, PROCESS, ANALYZE, SHARE, ACT. I am utilizing a public dataset spanning from December 2022 to November 2023. The analysis is conducted using the R programming language, leveraging its capabilities in data analysis and visualization. The primary objective of this project is to strategize the conversion of casual riders into annual members.

# Scenario

You are a junior data analyst working in 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, the director of marketting 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, the director of marketing believes there is a very good chance 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.

## ASK

This is the first phase of the analysis process which involves asking the right questions. The key stakeholders for this project are: Marketing department manager, The marketing analytics team and the Cyclistic’s Executive team. The right questions for this business task are:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

## PREPARE

1. Cyclistic’s historical data was made public by Motivate International Inc. The data can be accessed [here](https://divvy-tripdata.s3.amazonaws,com/index.html) under the [license](https://www.divvybikes.com/data.license-agreement).
2. I applied the ROCCC process to check for issues with bias or credibility in this data. Considering this process, the data is Reliable, Original, Comprehensive, Current and Cited.
3. I downloaded the data for a period of DECEMBER 2022 to NOVEMBER 2023, unzipped it, changed its format to .xls(excel file) from its original CVS file.

# Installing the required packages

install.packages(“tidyverse”) install.packages(“janitor”) install.packages(“lubridate”)

# Loading the required packages

library(tidyverse)

library(janitor)

library(lubridate)  
library(readxl)

# Creating data frame by importing excel files for each month

dec22 <- read\_excel("~/Custom Office Templates/202212-divvy-tripdataa.xltx")  
jan23 <- read\_excel("~/Custom Office Templates/202301-divvy-tripdataa.xltx")  
feb23 <- read\_excel("~/Custom Office Templates/202302-divvy-tripdata.xltx")  
mar23 <- read\_excel("~/Custom Office Templates/202303-divvy-tripdata.xltx")  
apr23 <- read\_excel("~/Custom Office Templates/202304-divvy-tripdata.xltx")  
may23 <- read\_excel("~/Custom Office Templates/202305-divvy-tripdata.xltx")  
jun23 <- read\_excel("~/Custom Office Templates/202306-divvy-tripdata.xltx")  
jul23 <- read\_excel("~/Custom Office Templates/202307-divvy-tripdata.xltx")  
aug23 <- read\_excel("~/Custom Office Templates/202308-divvy-tripdata.xltx")  
sep23 <- read\_excel("~/Custom Office Templates/202309-divvy-tripdata.xltx")  
oct23 <- read\_excel("~/Custom Office Templates/202310-divvy-tripdata.xltx")  
nov23 <- read\_excel("~/Custom Office Templates/202311-divvy-tripdata.xltx")

# Merging the 12 data frames into a single data frame

rides <- rbind(dec22, jan23, feb23, mar23, apr23, may23,   
 jun23, jul23, aug23, sep23, oct23, nov23)

## 

## PROCESS

This involves Cleaning the data to fix incomplete, incorrectly formatted and duplicate data. R programming is used to clean this large dataset.

# List column names of the data frame

colnames(rides)

## [1] "ride\_id" "rideable\_type" "started\_at"   
## [4] "ended\_at" "start\_station\_name" "start\_station\_id"   
## [7] "end\_station\_name" "end\_station\_id" "start\_lat"   
## [10] "start\_lng" "end\_lat" "end\_lng"   
## [13] "member\_casual"

# List the count of rows and columns

dim(rides)

## [1] 5677610 13

# List the column names and data types of the data frame

str(rides)

## tibble [5,677,610 × 13] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:5677610] "65DBD2F447EC51C2" "0C201AA7EA0EA1AD" "E0B148CCB358A49D" "54C5775D2B7C9188" ...  
## $ rideable\_type : chr [1:5677610] "electric\_bike" "classic\_bike" "electric\_bike" "classic\_bike" ...  
## $ started\_at : POSIXct[1:5677610], format: "2022-12-05 10:47:18" "2022-12-18 06:42:33" ...  
## $ ended\_at : POSIXct[1:5677610], format: "2022-12-05 10:56:34" "2022-12-18 07:08:44" ...  
## $ start\_station\_name: chr [1:5677610] "Clifton Ave & Armitage Ave" "Broadway & Belmont Ave" "Sangamon St & Lake St" "Shields Ave & 31st St" ...  
## $ start\_station\_id : chr [1:5677610] "TA1307000163" "13277" "TA1306000015" "KA1503000038" ...  
## $ end\_station\_name : chr [1:5677610] "Sedgwick St & Webster Ave" "Sedgwick St & Webster Ave" "St. Clair St & Erie St" "Damen Ave & Madison St" ...  
## $ end\_station\_id : chr [1:5677610] "13191" "13191" "13016" "13134" ...  
## $ start\_lat : num [1:5677610] 41.9 41.9 41.9 41.8 41.9 ...  
## $ start\_lng : num [1:5677610] -87.7 -87.6 -87.7 -87.6 -87.7 ...  
## $ end\_lat : num [1:5677610] 41.9 41.9 41.9 41.9 41.9 ...  
## $ end\_lng : num [1:5677610] -87.6 -87.6 -87.6 -87.7 -87.7 ...  
## $ member\_casual : chr [1:5677610] "member" "casual" "member" "member" ...

# Statistical Summary of the data frame

summary(rides)

## ride\_id rideable\_type started\_at   
## Length:5677610 Length:5677610 Min. :2022-12-01 00:01:22.00   
## Class :character Class :character 1st Qu.:2023-05-10 19:59:37.50   
## Mode :character Mode :character Median :2023-07-13 06:25:06.50   
## Mean :2023-07-03 14:16:54.50   
## 3rd Qu.:2023-09-07 09:19:15.00   
## Max. :2023-11-30 23:59:14.00   
##   
## ended\_at start\_station\_name start\_station\_id   
## Min. :2022-12-01 00:03:41.00 Length:5677610 Length:5677610   
## 1st Qu.:2023-05-10 20:18:03.00 Class :character Class :character   
## Median :2023-07-13 06:40:33.50 Mode :character Mode :character   
## Mean :2023-07-03 14:35:06.98   
## 3rd Qu.:2023-09-07 09:34:18.00   
## Max. :2023-12-01 20:42:31.00   
##   
## end\_station\_name end\_station\_id start\_lat start\_lng   
## Length:5677610 Length:5677610 Min. :41.63 Min. :-87.94   
## Class :character Class :character 1st Qu.:41.88 1st Qu.:-87.66   
## Mode :character Mode :character Median :41.90 Median :-87.64   
## Mean :41.90 Mean :-87.65   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.07 Max. :-87.46   
##   
## end\_lat end\_lng member\_casual   
## Min. : 0.00 Min. :-88.16 Length:5677610   
## 1st Qu.:41.88 1st Qu.:-87.66 Class :character   
## Median :41.90 Median :-87.64 Mode :character   
## Mean :41.90 Mean :-87.65   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.18 Max. : 0.00   
## NA's :6879 NA's :6879

# Remove NA data from data frame (rides)

rides1 <- drop\_na(rides)  
dim(rides1)

## [1] 4299967 13

summary(rides1)

## ride\_id rideable\_type started\_at   
## Length:4299967 Length:4299967 Min. :2022-12-01 00:01:22.0   
## Class :character Class :character 1st Qu.:2023-05-10 07:21:06.5   
## Mode :character Mode :character Median :2023-07-12 16:52:18.0   
## Mean :2023-07-03 03:47:14.9   
## 3rd Qu.:2023-09-07 07:39:31.5   
## Max. :2023-11-30 23:58:38.0   
## ended\_at start\_station\_name start\_station\_id   
## Min. :2022-12-01 00:03:41.00 Length:4299967 Length:4299967   
## 1st Qu.:2023-05-10 07:31:49.00 Class :character Class :character   
## Median :2023-07-12 17:04:54.00 Mode :character Mode :character   
## Mean :2023-07-03 04:03:12.25   
## 3rd Qu.:2023-09-07 07:51:02.50   
## Max. :2023-12-01 09:47:35.00   
## end\_station\_name end\_station\_id start\_lat start\_lng   
## Length:4299967 Length:4299967 Min. :41.65 Min. :-87.84   
## Class :character Class :character 1st Qu.:41.88 1st Qu.:-87.66   
## Mode :character Mode :character Median :41.90 Median :-87.64   
## Mean :41.90 Mean :-87.64   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.06 Max. :-87.53   
## end\_lat end\_lng member\_casual   
## Min. : 0.00 Min. :-87.84 Length:4299967   
## 1st Qu.:41.88 1st Qu.:-87.66 Class :character   
## Median :41.90 Median :-87.64 Mode :character   
## Mean :41.90 Mean :-87.64   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.06 Max. : 0.00

# Remove duplicate ride\_id

rides2 <- rides1[!duplicated(rides1$ride\_id), ]  
dim(rides2)

## [1] 4299967 13

# Remove empty rows and columns

rides1 <- janitor::remove\_empty(dat=rides1,which=c("cols"))  
rides1 <- janitor::remove\_empty(dat=rides1,which=c("rows"))  
cat("Dimensions of rides1:", nrow(rides1), "rows and", ncol(rides1), "columns\n")

## Dimensions of rides1: 4299967 rows and 13 columns

# Changing date and time from char to date format

rides1$started\_at = strptime(rides1$started\_at, "%Y-%m-%d %H:%M:%S")  
rides1$ended\_at = strptime(rides1$ended\_at, "%Y-%m-%d %H:%M:%S")  
class(rides1$started\_at)

## [1] "POSIXlt" "POSIXt"

class(rides1$ended\_at)

## [1] "POSIXlt" "POSIXt"

# Adding two columns for month and day respectively

rides1$month <- format(as.Date(rides1$started\_at), "%m")  
rides1$day\_of\_week <- format(as.Date(rides1$started\_at), "%A")

# Check the new columns

head(rides1[, c("month", "day\_of\_week")])

## # A tibble: 6 × 2  
## month day\_of\_week  
## <chr> <chr>   
## 1 12 Monday   
## 2 12 Sunday   
## 3 12 Tuesday   
## 4 12 Tuesday   
## 5 12 Wednesday   
## 6 12 Friday

str(rides1[, c("month", "day\_of\_week")])

## tibble [4,299,967 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ month : chr [1:4299967] "12" "12" "12" "12" ...  
## $ day\_of\_week: chr [1:4299967] "Monday" "Sunday" "Tuesday" "Tuesday" ...

unique(rides1$month)

## [1] "12" NA "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11"

unique(rides1$day\_of\_week)

## [1] "Monday" "Sunday" "Tuesday" "Wednesday" "Friday" "Thursday"   
## [7] "Saturday" NA

# Adding trip duration in seconds and minutes

rides1$ride\_length <- difftime(rides1$ended\_at,rides1$started\_at)  
rides1$ride\_length\_m <- (difftime(rides1$ended\_at,rides1$started\_at))/60

# Check the new columns

head(rides1[, c("ride\_length", "ride\_length\_m")])

## # A tibble: 6 × 2  
## ride\_length ride\_length\_m   
## <drtn> <drtn>   
## 1 556 secs 9.266667 min  
## 2 1571 secs 26.183333 min  
## 3 726 secs 12.100000 min  
## 4 1741 secs 29.016667 min  
## 5 851 secs 14.183333 min  
## 6 567 secs 9.450000 min

str(rides1[, c("ride\_length", "ride\_length\_m")])

## tibble [4,299,967 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_length : 'difftime' num [1:4299967] 556 1571 726 1741 ...  
## ..- attr(\*, "units")= chr "secs"  
## $ ride\_length\_m: 'difftime' num [1:4299967] 9.26666666666667 26.1833333333333 12.1 29.0166666666667 ...  
## ..- attr(\*, "units")= chr "min"

# Filter trip duration greater than zero in seconds and minutes

rides1 <- filter(rides1, rides1$ride\_length>0)  
rides1 <- filter(rides1, rides1$ride\_length\_m>0)  
head(rides1)

## # A tibble: 6 × 17  
## ride\_id rideable\_type started\_at ended\_at   
## <chr> <chr> <dttm> <dttm>   
## 1 65DBD2F447EC51C2 electric\_bike 2022-12-05 10:47:18 2022-12-05 10:56:34  
## 2 0C201AA7EA0EA1AD classic\_bike 2022-12-18 06:42:33 2022-12-18 07:08:44  
## 3 E0B148CCB358A49D electric\_bike 2022-12-13 08:47:45 2022-12-13 08:59:51  
## 4 54C5775D2B7C9188 classic\_bike 2022-12-13 18:50:47 2022-12-13 19:19:48  
## 5 A4891F78776D35DF classic\_bike 2022-12-14 16:13:39 2022-12-14 16:27:50  
## 6 DB91D9B8DFACA07A electric\_bike 2022-12-02 15:24:47 2022-12-02 15:34:14  
## # ℹ 13 more variables: start\_station\_name <chr>, start\_station\_id <chr>,  
## # end\_station\_name <chr>, end\_station\_id <chr>, start\_lat <dbl>,  
## # start\_lng <dbl>, end\_lat <dbl>, end\_lng <dbl>, member\_casual <chr>,  
## # month <chr>, day\_of\_week <chr>, ride\_length <drtn>, ride\_length\_m <drtn>

## ANALYZE

This section shows the descriptive analysis on trip duration for casuals and members users

rides1 %>%  
 group\_by(member\_casual) %>%  
 summarize(mean = mean(ride\_length),   
 median = median(ride\_length),   
 max = max(ride\_length),   
 min = min(ride\_length), )

## # A tibble: 2 × 5  
## member\_casual mean median max min   
## <chr> <drtn> <drtn> <drtn> <drtn>  
## 1 casual 1375.6798 secs 765 secs 728178 secs 1 secs  
## 2 member 727.3825 secs 517 secs 89872 secs 1 secs

# Average trip duration by each day of week for members vs casual users

aggregate(rides1$ride\_length ~ rides1$member\_casual + rides1$day\_of\_week, FUN = mean)

## rides1$member\_casual rides1$day\_of\_week rides1$ride\_length  
## 1 casual Friday 1342.3552 secs  
## 2 member Friday 722.1001 secs  
## 3 casual Monday 1350.3308 secs  
## 4 member Monday 691.5729 secs  
## 5 casual Saturday 1555.2106 secs  
## 6 member Saturday 815.9539 secs  
## 7 casual Sunday 1593.3076 secs  
## 8 member Sunday 818.1290 secs  
## 9 casual Thursday 1199.4113 secs  
## 10 member Thursday 693.4649 secs  
## 11 casual Tuesday 1227.7919 secs  
## 12 member Tuesday 698.4422 secs  
## 13 casual Wednesday 1175.1200 secs  
## 14 member Wednesday 696.0243 secs

colnames(rides1)

## [1] "ride\_id" "rideable\_type" "started\_at"   
## [4] "ended\_at" "start\_station\_name" "start\_station\_id"   
## [7] "end\_station\_name" "end\_station\_id" "start\_lat"   
## [10] "start\_lng" "end\_lat" "end\_lng"   
## [13] "member\_casual" "month" "day\_of\_week"   
## [16] "ride\_length" "ride\_length\_m"

# Re-ordering the day of week

rides1$day\_of\_week <- ordered(rides1$day\_of\_week, levels = c("Sunday", "Monday", "Tuesday", "Wednesday",   
 "Thursday", "Friday", "Saturday"))  
aggregate(rides1$ride\_length ~ rides1$member\_casual + rides1$day\_of\_week, FUN = mean)

## rides1$member\_casual rides1$day\_of\_week rides1$ride\_length  
## 1 casual Sunday 1593.3076 secs  
## 2 member Sunday 818.1290 secs  
## 3 casual Monday 1350.3308 secs  
## 4 member Monday 691.5729 secs  
## 5 casual Tuesday 1227.7919 secs  
## 6 member Tuesday 698.4422 secs  
## 7 casual Wednesday 1175.1200 secs  
## 8 member Wednesday 696.0243 secs  
## 9 casual Thursday 1199.4113 secs  
## 10 member Thursday 693.4649 secs  
## 11 casual Friday 1342.3552 secs  
## 12 member Friday 722.1001 secs  
## 13 casual Saturday 1555.2106 secs  
## 14 member Saturday 815.9539 secs

# Average Ride Length and number of rides by day of the week

rides1 %>%  
 group\_by(member\_casual, day\_of\_week) %>%  
 summarise(number\_of\_rides = n(), average\_ride\_length = mean(ride\_length)) %>%  
 arrange(member\_casual, day\_of\_week)

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

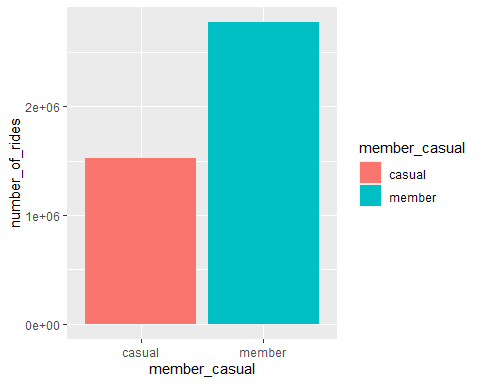
## # A tibble: 14 × 4  
## # Groups: member\_casual [2]  
## member\_casual day\_of\_week number\_of\_rides average\_ride\_length  
## <chr> <ord> <int> <drtn>   
## 1 casual Sunday 253149 1593.3076 secs   
## 2 casual Monday 175281 1350.3308 secs   
## 3 casual Tuesday 181850 1227.7919 secs   
## 4 casual Wednesday 181970 1175.1200 secs   
## 5 casual Thursday 199307 1199.4113 secs   
## 6 casual Friday 226152 1342.3552 secs   
## 7 casual Saturday 308628 1555.2106 secs   
## 8 member Sunday 303210 818.1290 secs   
## 9 member Monday 384386 691.5729 secs   
## 10 member Tuesday 448020 698.4422 secs   
## 11 member Wednesday 446690 696.0243 secs   
## 12 member Thursday 451694 693.4649 secs   
## 13 member Friday 393325 722.1001 secs   
## 14 member Saturday 345719 815.9539 secs

## SHARE

The tool used for the visualization of the analyzed data is Rstudio.

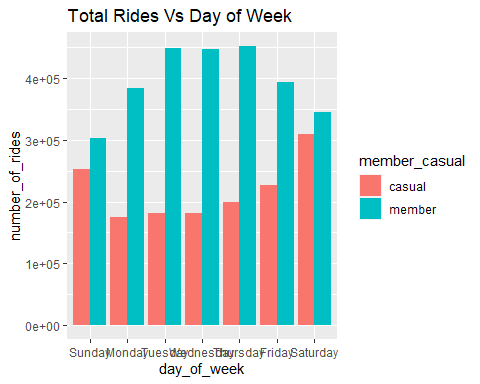
# Visualization of the total number of rides for member and casual riders

rides1 %>%  
 group\_by(member\_casual) %>%  
 summarize(number\_of\_rides = n()) %>%  
 arrange(member\_casual) %>%  
 ggplot(aes(x=member\_casual, y=number\_of\_rides, fill=member\_casual)) +  
 geom\_col(position = "dodge")



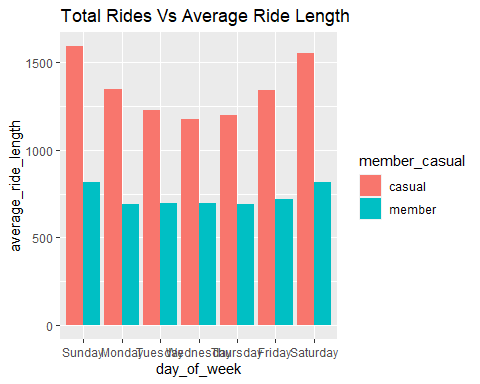
# Visualization of the number of rides by day of the week with rider type

rides1 %>%  
 group\_by(member\_casual, day\_of\_week) %>%  
 summarize(number\_of\_rides = n(), average\_ride\_length = mean(ride\_length)) %>%  
 arrange(member\_casual, day\_of\_week) %>%  
 ggplot(aes(x=day\_of\_week, y=number\_of\_rides, fill=member\_casual)) +  
 geom\_col(position = "dodge") +   
 labs(title="Total Rides Vs Day of Week")



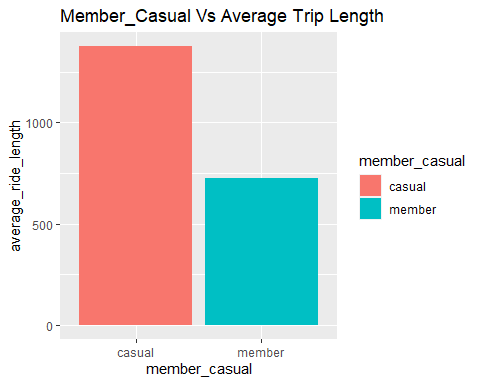
# Visualization of the average ride length with number of rides

rides1 %>%  
 group\_by(member\_casual, day\_of\_week) %>%  
 summarize(average\_ride\_length = mean(ride\_length)) %>%  
 arrange(member\_casual, day\_of\_week) %>%  
 ggplot(aes(x=day\_of\_week, y=average\_ride\_length, fill=member\_casual)) +  
 geom\_col(position = "dodge") +   
 labs(title="Total Rides Vs Average Ride Length")



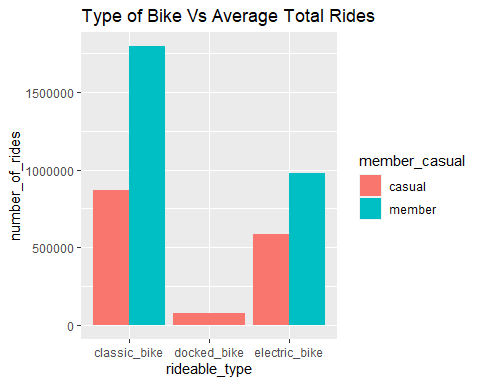
# Visualization of the average length of ride with member type

rides1 %>%  
 group\_by(member\_casual) %>%  
 summarize(average\_ride\_length = mean(ride\_length)) %>%  
 arrange(member\_casual) %>%  
 ggplot(aes(x=member\_casual, y=average\_ride\_length, fill=member\_casual)) +  
 geom\_col(position = "dodge") +   
 labs(title="Member\_Casual Vs Average Trip Length")



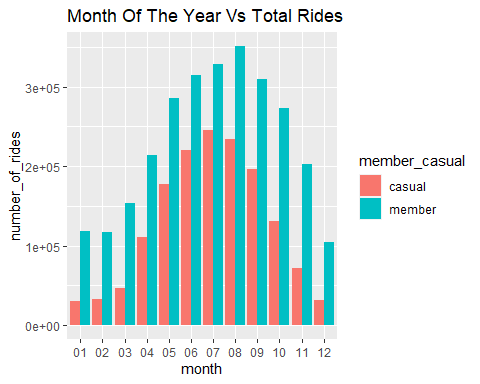
# Vizualization of the different types of bikes with the total number of rides

rides1 %>%  
 group\_by(member\_casual, rideable\_type) %>%  
 summarize(number\_of\_rides = n()) %>%  
 arrange(member\_casual, rideable\_type) %>%  
 ggplot(aes(x=rideable\_type, y=number\_of\_rides, fill=member\_casual)) +  
 geom\_col(position = "dodge") +   
 labs(title="Type of Bike Vs Average Total Rides")



# Vizualization of the months of the year with the total number of rides

rides1 %>%  
 group\_by(member\_casual, month) %>%  
 summarize(number\_of\_rides = n()) %>%  
 arrange(member\_casual, month) %>%  
 ggplot(aes(x=month, y=number\_of\_rides, fill=member\_casual)) +  
 geom\_col(position="dodge") +   
 labs(title="Month Of The Year Vs Total Rides")



## ACT

This section involves the conclusion and Recommendation to Stakeholders.

# Identifying Trends and Relationships.

Both User Types: 1. They barely ride the Cyclistic bikes during the colder months. 2. They use the Classic bike type of rideable bikes more.

# Casuals

1. They have the Largest Average Ride length done mostly on weekends (Leisure).
2. The highest rides occur in July (Summer-time).
3. They have a negligible presence during winter months.

# Members

1. They have the highest number of rides during the week days.
2. The highest rides occur in August (Pleasantly warm temperature with a cool breeze).
3. They barely use the docked bike compared to electric and classic bike.
4. They have a consistent Average Ride Length during weekdays.

# Conclusion

1. Membership riders have taken the maximum number of rides compared to the casual riders.
2. Membership riders have a greater number of rides during week days while Casual riders has a greater number of trips during weekends(leisure).

# Recommendations

1. We can provide the casual riders the option to select a weekend membership program.
2. We can offer promotions and discounts to casual riders after subscription to membership program.
3. Arrangement of Membership rider’s competition with prizes and awards can be done.
4. Cyclistic can influence casual riders to become members by using socia; media platforms on weekends in their stations.