**Abstract:**

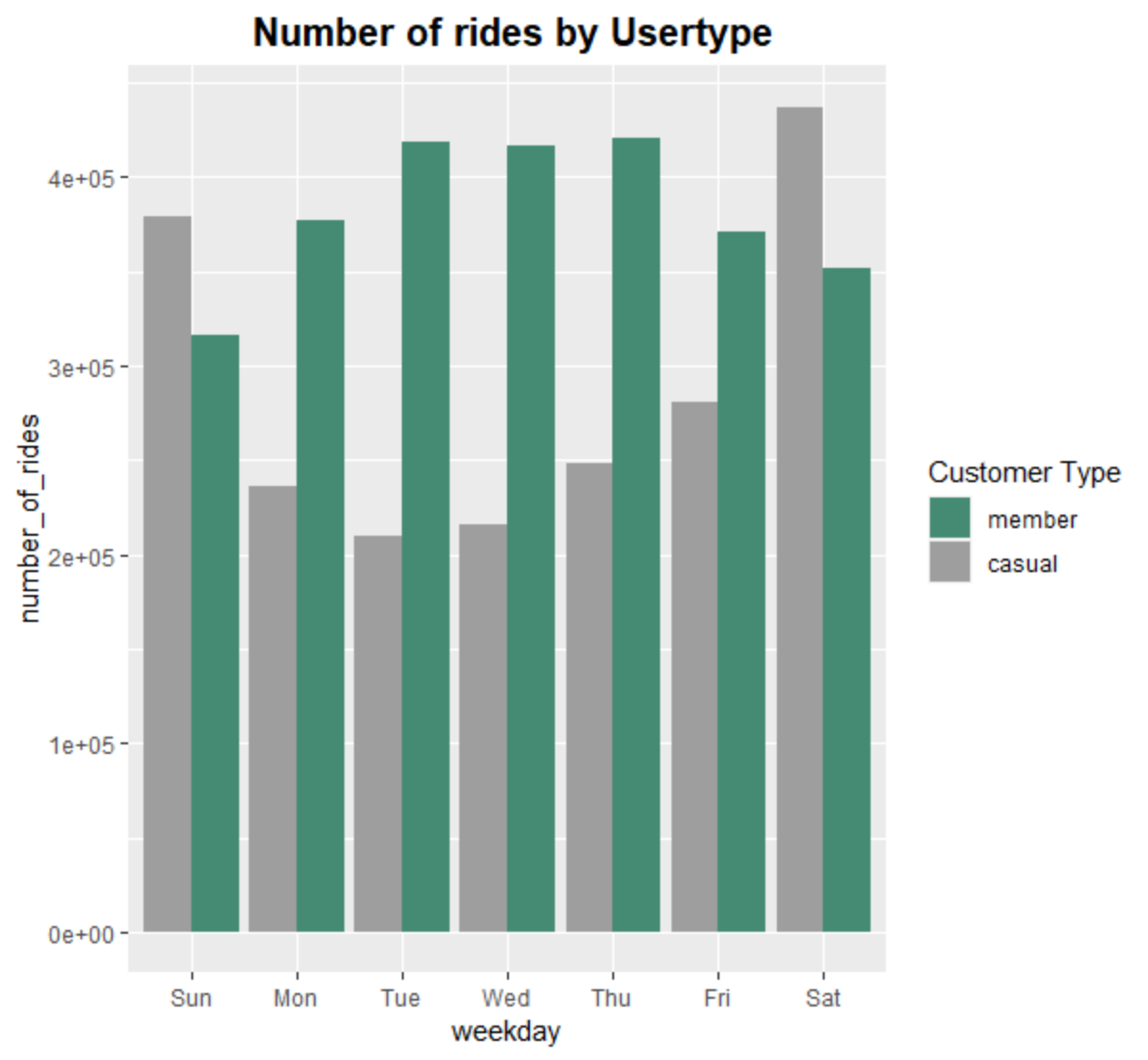
In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. 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 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. The main purpose of is to design marketing strategies aimed at converting casual riders into annual members.

**Goal:**

How do annual members and casual riders use Cyclistic bikes differently?

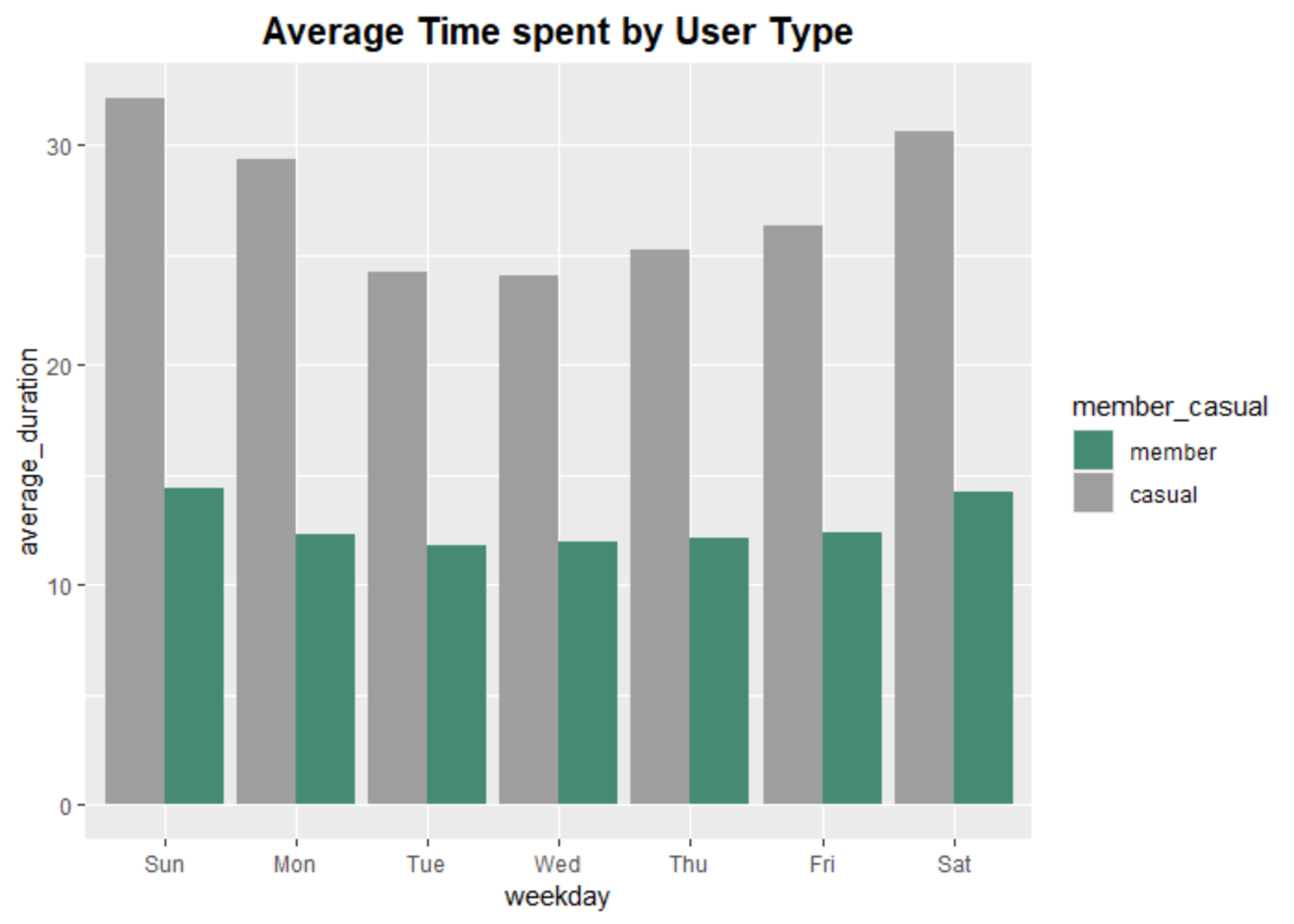
**Steps**

To tackle this problem, we have divided this question to subdivisions of data life cycle, Ask, Prepare, Process, Analyze, Share, Act.

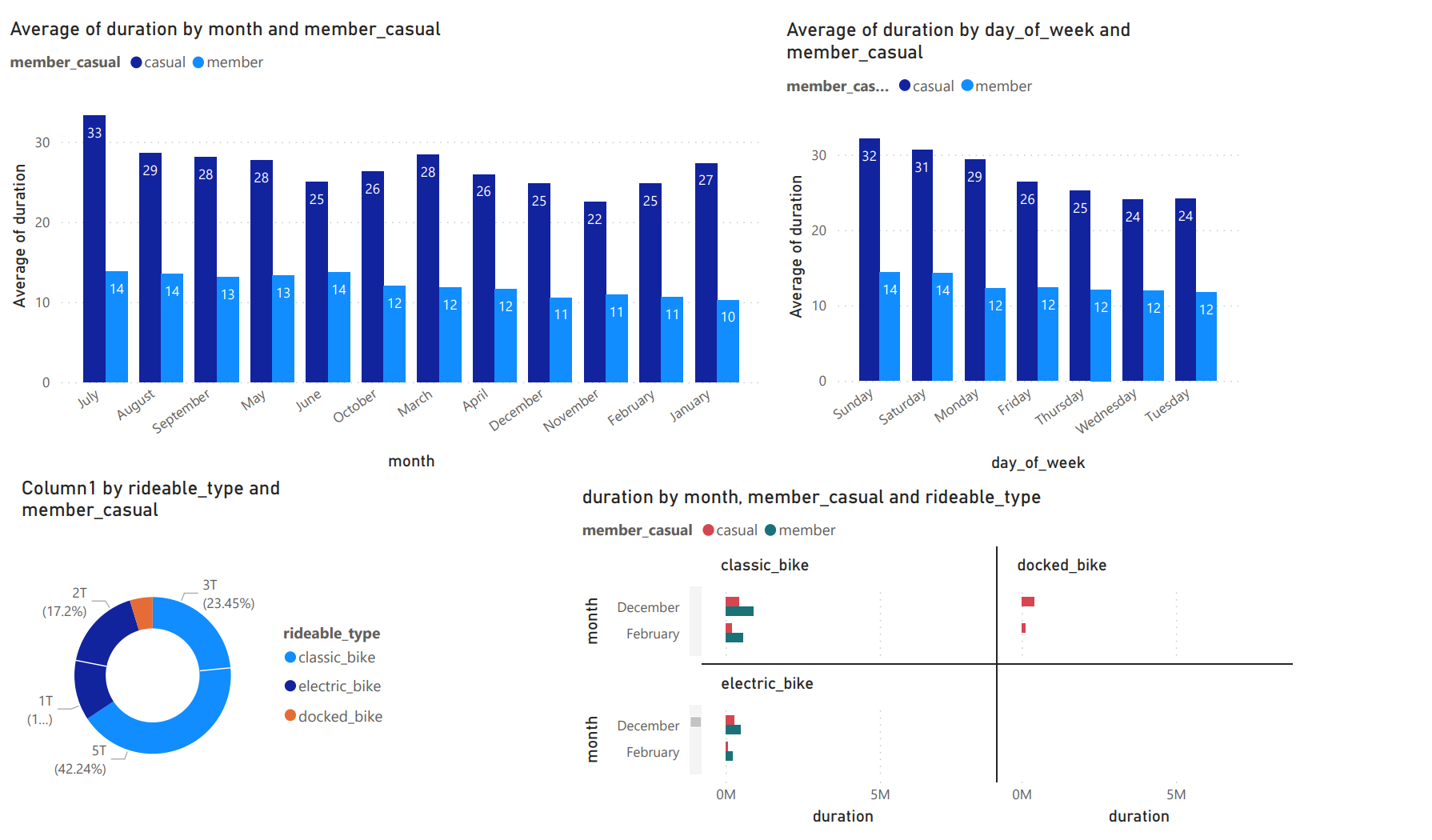


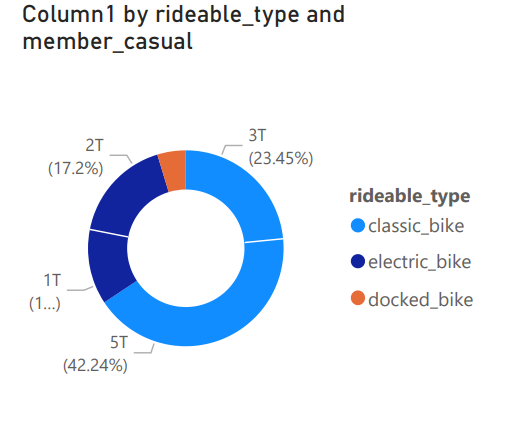
2.0

The first question we need to answer is how annual members and casual riders use Cyclistic bikes differently? And from the data chart (2.0) we can see that most paid member use the service more during the weekdays, and the number starts to decrease at weekend



1. Why would casual riders buy Cyclistic annual memberships? Because is clear from the data that the causal members stay longer with the bike in contrast with monthly paying members. Since they’re paying on time been used on the product, we could market to them why is more logical to pay monthly membership which will save them money
2. How can Cyclistic use digital media to influence casual riders to become members





Visualized using Power BI

Market more on classic bike since that is what people are using more, find programs that people could do summer time since our highest time spent on average was July. Finally we should also look at the merging market of electric bike since the younger generation are more aware and informed about of the environment; electric bike would could be more fun and environmental friendly way of commute

**Suggestion**

Perhaps more data on why peoples would use this product will give us insight on where the future of rideable type will be for example riding to get fit will suggest to focus more on the classic bike than the dock or electric bike.

**Analysis**

|  |
| --- |
| #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  #Install required packages  install. Packages(“tidyverse”)  install. Packages(“ggplot2”)  install. Packages(“lubridate”)  install. Packages(“tidyr”)  #Loading package  library(tidyverse) #helps wrangle data  library(lubridate) #helps wrangle date attributes  library(ggplot2) #helps visualize data  library (dplyr) # the be able to use the filter method  # |

|  |
| --- |
| d\_packages  > library(tidyverse) #helps wrangle data  ── Attaching packages ─────────────────── tidyverse 1.3.2 ──  ✔ ggplot2 3.3.6 ✔ urr 0.3.4  ✔ tibble 3.1.6 ✔ dplyr 1.0.9  ✔ tidyr 1.2.0 ✔ stringr 1.4.0  ✔ readr 2.1.2 ✔ forcats 0.5.1  ── Conflicts ────────────────────── tidyverse\_conflicts() ──  ✖ dplyr::filter() masks stats::filter()  ✖ dplyr::lag() masks stats::lag()  Warning messages:  1: package ‘tidyverse’ was built under R version 4.2.1  2: package ‘ggplot2’ was built under R version 4.2.1  3: package ‘tidyr’ was built under R version 4.2.1  > library(lubridate) #helps wrangle date attributes  Attaching package: ‘lubridate’  The following objects are masked from ‘package:base’:  date, intersect, setdiff, union  Warning message:  package ‘lubridate’ was built under R version 4.2.1  > library(ggplot2) #helps visualize data  > library (dplyr)  > install.packages(‘plyr’, repos = “http://cran.us.r-project.org”)  WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:  <https://cran>.rstudio.com/bin/windows/Rtools/  Installing package into ‘C:/Users/chris/AppData/Local/R/win-library/4.2’  (as ‘lib’ is unspecified)  also installing the dependency ‘Rcpp’  trying URL ‘http://cran.us.r-project.org/bin/windows/contrib/4.2/Rcpp\_1.0.9.zip’  Content type ‘application/zip’ length 2842372 bytes (2.7 MB)  downloaded 2.7 MB  trying URL ‘http://cran.us.r-project.org/bin/windows/contrib/4.2/plyr\_1.8.7.zip’  Content type ‘application/zip’ length 1154796 bytes (1.1 MB)  downloaded 1.1 MB |
| # #=====================  # # STEP 1: COLLECT DATA  # #=====================  # # Upload Divvy datasets (csv files) here  Jul\_2021 <- read\_csv("202107-divvy-tripdata.csv")  Aug\_2021 <- read\_csv("202108-divvy-tripdata.csv")  Sep\_2021 <- read\_csv("202109-divvy-tripdata.csv")  Oct\_2021 <- read\_csv("202110-divvy-tripdata.csv")  Nov\_2021 <- read\_csv("202111-divvy-tripdata.csv")  Dec\_2021 <- read\_csv("202112-divvy-tripdata.csv")  Jan\_2022 <- read\_csv("202201-divvy-tripdata.csv")  Feb\_2022 <- read\_csv("202202-divvy-tripdata.csv")  Mar\_2022 <- read\_csv("202203-divvy-tripdata.csv")  Apr\_2022 <- read\_csv("202204-divvy-tripdata.csv")  May\_2022 <- read\_csv("202205-divvy-tripdata.csv")  June\_2022 <- read\_csv("202206-divvy-tripdata.csv")  #====================================================  # STEP 2: WRANGLE DATA AND COMBINE INTO A SINGLE FILE  #check for consistence in the column  colnames (Jul\_2021)  colnames (Aug\_2021)  colnames(Sep\_2021)  colnames (Oct\_2021)  colnames (Nov\_2021)  colnames(Dec\_2021)  colnames (Jan\_2022)  colnames (Feb\_2022)  colnames (Mar\_2022)  colnames (Apr\_2022)  colnames (May\_2022)  colnames (June\_2022)  # Inspect the dataframes and look for in inconsistency  # so, all the column are of the same data type  str(Jul\_2021)  str(Aug\_2021)  str(q3\_2019)  str(Sep\_2021)  # # Remove "bad" data where rows are NA or duplicate  #==================================  # Enter the data or CSV file you  # will clean and merge below  #====================================  df1<- bind\_rows(  #enter below  Jul\_2021,  Aug\_2021,  Sep\_2021 ,  Oct\_2021,  Nov\_2021,  Dec\_2021,  Jan\_2022,  Feb\_2022,  Mar\_2022,  Apr\_2022,  May\_2022,  June\_2022  )  #change to data frame format  df2<-data.frame(df2)  # Apple na.omit method to remove all NA  df2<-na.omit(df1)  # To reassure that there is no NA value in your data use the  # is.na and any method. If it returns a Boolean TRUE than there are  # still some NA left hence if it returns FALSE, than all NA have been removed  any(is.na(df2))  #Next is not remove duplicate rows  #Apply the unique() or distinct() function for data frame in R  df2<-unique(df2)  #Add columns that list the date, month, day, and year of each ride  #This will allow us to do some aggregation  df2$started\_at <- ymd\_hms(df2$started\_at)  df2$ended\_at <- ymd\_hms(df2$ended\_at)  df2$duration <- as.numeric(difftime(df2$ended\_at, df2$started\_at, units="min"))  df2$month <- format(df2$started\_at, format="%B")  df2$day\_of\_week <- format(df2$started\_at, format="%A")  df2$hour <- format(df2$started\_at, format="%H")  glimpse(df2)  df2 %>%  filter(duration <= 0) %>%  count()  df2<- df2 %>%  filter(duration > 0)  head(df2)  # Compare average, middle minimum, and maximum time  #members and casual users are using the service  aggregate(df2$duration ~ df2$member\_casual, FUN = mean)  aggregate(df2$duration ~ df2$member\_casual, FUN = median)  aggregate (df2$duration ~ df2$member\_casual, FUN = max)  aggregate(df2$duration ~ df2$member\_casual, FUN = min)  summary(df2$duration)  view(max(duration))  # See the average ride time  #by each day for members vs casual users  aggregate(df2$duration ~ df2$member\_casual + df2$day\_of\_week, FUN = mean)  #rearrange to display chronically  df2$day\_of\_week <- ordered(df2$day\_of\_week,  levels=c("Sunday", "Monday", "Tuesday",  "Wednesday", "Thursday", "Friday",  "Saturday"))  aggregate(df2$duration ~ df2$member\_casual + df2$day\_of\_week, FUN = mean)  #==================================  # Visualization  # Using GGplot 2  #==================================  #show what days are both rider type use the service more  # or the number of rides per user type  #=============================  #Weekly  #=============================  df2 %>%  mutate(weekday = wday(started\_at, label = TRUE)) %>%  group\_by(member\_casual, weekday) %>%  summarise(number\_of\_rides = n()  ,average\_duration = mean(duration)) %>%  arrange(member\_casual, weekday) %>% #sorts  ggplot(aes(x = weekday, y = number\_of\_rides, fill = member\_casual)) +  geom\_col(position = "dodge") +  scale\_fill\_manual(values=c('#458B74','8B7355'),  limits = c("member", "casual"))+  guides(fill=guide\_legend(title="Customer Type")) + ggtitle(" Number of rides by Usertype") +  theme(plot.title = element\_text(size = 15, face = "bold", hjust = 0.5) )    =============================  #Monthly  #=============================  # Let's create a visualization for average duration  df2 %>%  mutate(weekday = wday(started\_at, label = TRUE)) %>%  group\_by(member\_casual, weekday) %>%  summarise(number\_of\_rides = n()  ,average\_duration = mean(duration)) %>%  arrange(member\_casual, weekday) %>%  ggplot(aes(x = weekday, y = average\_duration, fill = member\_casual)) +  geom\_col(position = "dodge") + scale\_fill\_manual(values=c('#458B74','8B7355'),  limits = c("member", "casual")) + ggtitle("Average Time spent by User Type") +  theme(plot.title = element\_text(size = 15, face = "bold", hjust = 0.5) )  #export to csv  write.csv(df2,file = "cyclistic\_data.csv") |

Cites

Files used to do this analysis was provided by Coursera/ Google Certification

[Index of bucket "divvy-tripdata"](https://divvy-tripdata.s3.amazonaws.com/index.html)