

Google Data Analytics Certificate Capstone Project

Daniel Geda

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

Case Study 1: How Does a Bike-Share Navigate Speedy Success?

This project was part of my Google Data Analytics Certificate capstone project.

Here's the scenario: We are working with a fictional company, Cyclistic, which is 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, our team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, the marketing team will design a new marketing strategy to convert casual riders into annual members.

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

Data source: Cyclistic's historical trip data The data has been made available by Motivate International Inc. under this license (<https://divvybikes.com/data-license-agreement>). I am using the trip data from the 4 quarters in 2019.

Load R libraries

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.4      v dplyr  1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   2.0.1      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(readr)
library(lubridate)
```

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

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

```
library(ggplot2)
```

Load data into R

```
Q1_2019 <- read_csv("Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 Columns: 12
```

```
## -- Column specification -----
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## dtm  (2): start_time, end_time

##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
Q2_2019 <- read_csv("Divvy_Trips_2019_Q2.csv")
```

```
## Rows: 1108163 Columns: 12
```

```
## -- Column specification -----
## Delimiter: ","
## chr  (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...
## dbl  (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...
## dtm  (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local En...

##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
Q3_2019 <- read_csv("Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
```

```
## -- Column specification -----
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## dtm  (2): start_time, end_time

##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
Q4_2019 <- read_csv("Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
```

```
## -- Column specification -----  
## Delimiter: ","  
## chr  (4): from_station_name, to_station_name, usertype, gender  
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear  
## dtm  (2): start_time, end_time  
  
##  
## i Use 'spec()' to retrieve the full column specification for this data.  
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Look at the column names for the four quarters to make sure there are no discrepancies

```
colnames(Q1_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"  
## [4] "bikeid"           "tripduration"     "from_station_id"  
## [7] "from_station_name" "to_station_id"    "to_station_name"  
## [10] "usertype"         "gender"           "birthyear"
```

```
colnames(Q2_2019)
```

```
## [1] "01 - Rental Details Rental ID"  
## [2] "01 - Rental Details Local Start Time"  
## [3] "01 - Rental Details Local End Time"  
## [4] "01 - Rental Details Bike ID"  
## [5] "01 - Rental Details Duration In Seconds Uncapped"  
## [6] "03 - Rental Start Station ID"  
## [7] "03 - Rental Start Station Name"  
## [8] "02 - Rental End Station ID"  
## [9] "02 - Rental End Station Name"  
## [10] "User Type"  
## [11] "Member Gender"  
## [12] "05 - Member Details Member Birthday Year"
```

```
colnames(Q3_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"  
## [4] "bikeid"           "tripduration"     "from_station_id"  
## [7] "from_station_name" "to_station_id"    "to_station_name"  
## [10] "usertype"         "gender"           "birthyear"
```

```
colnames(Q4_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"
## [4] "bikeid"           "tripduration"     "from_station_id"
## [7] "from_station_name" "to_station_id"    "to_station_name"
## [10] "usertype"         "gender"           "birthyear"
```

Rename Q2_2019 columns to make them consistent with the rest of the quarters

```
Q2_2019 <- rename(Q2_2019
  ,trip_id = "01 - Rental Details Rental ID"
  ,start_time = "01 - Rental Details Local Start Time"
  ,end_time = "01 - Rental Details Local End Time"
  ,bikeid = "01 - Rental Details Bike ID"
  ,tripduration = "01 - Rental Details Duration In Seconds Uncapped"
  ,from_station_id = "03 - Rental Start Station ID"
  ,from_station_name = "03 - Rental Start Station Name"
  ,to_station_id = "02 - Rental End Station ID"
  ,to_station_name = "02 - Rental End Station Name"
  ,usertype = "User Type"
  ,gender = "Member Gender"
  ,birthyear = "05 - Member Details Member Birthday Year")
```

Check if renaming was successful and do a double check using str() and colnames()

```
str(Q1_2019)
```

```
## spec_tbl_df [365,069 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ trip_id      : num [1:365069] 21742443 21742444 21742445 21742446 21742447 ...
## $ start_time   : POSIXct[1:365069], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
## $ end_time     : POSIXct[1:365069], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
## $ bikeid       : num [1:365069] 2167 4386 1524 252 1170 ...
## $ tripduration : num [1:365069] 390 441 829 1783 364 ...
## $ from_station_id : num [1:365069] 199 44 15 123 173 98 98 211 150 268 ...
## $ from_station_name: chr [1:365069] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
## $ to_station_id   : num [1:365069] 84 624 644 176 35 49 49 142 148 141 ...
## $ to_station_name : chr [1:365069] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
## $ usertype       : chr [1:365069] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
## $ gender         : chr [1:365069] "Male" "Female" "Female" "Male" ...
## $ birthyear      : num [1:365069] 1989 1990 1994 1993 1994 ...
## - attr(*, "spec")=
## .. cols(
## ..   trip_id = col_double(),
## ..   start_time = col_datetime(format = ""),
```

```
## .. end_time = col_datetime(format = ""),
## .. bikeid = col_double(),
## .. tripduration = col_number(),
## .. from_station_id = col_double(),
## .. from_station_name = col_character(),
## .. to_station_id = col_double(),
## .. to_station_name = col_character(),
## .. usertype = col_character(),
## .. gender = col_character(),
## .. birthyear = col_double()
## .. )
## - attr(*, "problems")=<externalptr>
```

```
str(Q2_2019)
```

```
## spec_tbl_df [1,108,163 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ trip_id : num [1:1108163] 22178529 22178530 22178531 22178532 22178533 ...
## $ start_time : POSIXct[1:1108163], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...
## $ end_time : POSIXct[1:1108163], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...
## $ bikeid : num [1:1108163] 6251 6226 5649 4151 3270 ...
## $ tripduration : num [1:1108163] 446 1048 252 357 1007 ...
## $ from_station_id : num [1:1108163] 81 317 283 26 202 420 503 260 211 211 ...
## $ from_station_name: chr [1:1108163] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jacks
## $ to_station_id : num [1:1108163] 56 59 174 133 129 426 500 499 211 211 ...
## $ to_station_name : chr [1:1108163] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal
## $ usertype : chr [1:1108163] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
## $ gender : chr [1:1108163] "Male" "Female" "Male" "Male" ...
## $ birthyear : num [1:1108163] 1975 1984 1990 1993 1992 ...
## - attr(*, "spec")=
## .. cols(
## .. '01 - Rental Details Rental ID' = col_double(),
## .. '01 - Rental Details Local Start Time' = col_datetime(format = ""),
## .. '01 - Rental Details Local End Time' = col_datetime(format = ""),
## .. '01 - Rental Details Bike ID' = col_double(),
## .. '01 - Rental Details Duration In Seconds Uncapped' = col_number(),
## .. '03 - Rental Start Station ID' = col_double(),
## .. '03 - Rental Start Station Name' = col_character(),
## .. '02 - Rental End Station ID' = col_double(),
## .. '02 - Rental End Station Name' = col_character(),
## .. 'User Type' = col_character(),
## .. 'Member Gender' = col_character(),
## .. '05 - Member Details Member Birthday Year' = col_double()
## .. )
## - attr(*, "problems")=<externalptr>
```

```
str(Q3_2019)
```

```
## spec_tbl_df [1,640,718 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ trip_id : num [1:1640718] 23479388 23479389 23479390 23479391 23479392 ...
## $ start_time : POSIXct[1:1640718], format: "2019-07-01 00:00:27" "2019-07-01 00:01:16" ...
## $ end_time : POSIXct[1:1640718], format: "2019-07-01 00:20:41" "2019-07-01 00:18:44" ...
## $ bikeid : num [1:1640718] 3591 5353 6180 5540 6014 ...
## $ tripduration : num [1:1640718] 1214 1048 1554 1503 1213 ...
```

```
## $ from_station_id : num [1:1640718] 117 381 313 313 168 300 168 313 43 43 ...
## $ from_station_name: chr [1:1640718] "Wilton Ave & Belmont Ave" "Western Ave & Monroe St" "Lakeview
## $ to_station_id : num [1:1640718] 497 203 144 144 62 232 62 144 195 195 ...
## $ to_station_name : chr [1:1640718] "Kimball Ave & Belmont Ave" "Western Ave & 21st St" "Larrabee S
## $ usertype : chr [1:1640718] "Subscriber" "Customer" "Customer" "Customer" ...
## $ gender : chr [1:1640718] "Male" NA NA NA ...
## $ birthyear : num [1:1640718] 1992 NA NA NA NA ...
## - attr(*, "spec")=
## .. cols(
## .. trip_id = col_double(),
## .. start_time = col_datetime(format = ""),
## .. end_time = col_datetime(format = ""),
## .. bikeid = col_double(),
## .. tripduration = col_number(),
## .. from_station_id = col_double(),
## .. from_station_name = col_character(),
## .. to_station_id = col_double(),
## .. to_station_name = col_character(),
## .. usertype = col_character(),
## .. gender = col_character(),
## .. birthyear = col_double()
## .. )
## - attr(*, "problems")=<externalptr>
```

```
str(Q4_2019)
```

```
## spec_tbl_df [704,054 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ trip_id : num [1:704054] 25223640 25223641 25223642 25223643 25223644 ...
## $ start_time : POSIXct[1:704054], format: "2019-10-01 00:01:39" "2019-10-01 00:02:16" ...
## $ end_time : POSIXct[1:704054], format: "2019-10-01 00:17:20" "2019-10-01 00:06:34" ...
## $ bikeid : num [1:704054] 2215 6328 3003 3275 5294 ...
## $ tripduration : num [1:704054] 940 258 850 2350 1867 ...
## $ from_station_id : num [1:704054] 20 19 84 313 210 156 84 156 156 336 ...
## $ from_station_name: chr [1:704054] "Sheffield Ave & Kingsbury St" "Throop (Loomis) St & Taylor St"
## $ to_station_id : num [1:704054] 309 241 199 290 382 226 142 463 463 336 ...
## $ to_station_name : chr [1:704054] "Leavitt St & Armitage Ave" "Morgan St & Polk St" "Wabash Ave &
## $ usertype : chr [1:704054] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
## $ gender : chr [1:704054] "Male" "Male" "Female" "Male" ...
## $ birthyear : num [1:704054] 1987 1998 1991 1990 1987 ...
## - attr(*, "spec")=
## .. cols(
## .. trip_id = col_double(),
## .. start_time = col_datetime(format = ""),
## .. end_time = col_datetime(format = ""),
## .. bikeid = col_double(),
## .. tripduration = col_number(),
## .. from_station_id = col_double(),
## .. from_station_name = col_character(),
## .. to_station_id = col_double(),
## .. to_station_name = col_character(),
## .. usertype = col_character(),
## .. gender = col_character(),
## .. birthyear = col_double()
## .. )
```

```
## - attr(*, "problems")=<externalptr>
```

```
colnames(Q1_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"
## [4] "bikeid"           "tripduration"     "from_station_id"
## [7] "from_station_name" "to_station_id"    "to_station_name"
## [10] "usertype"         "gender"           "birthyear"
```

```
colnames(Q2_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"
## [4] "bikeid"           "tripduration"     "from_station_id"
## [7] "from_station_name" "to_station_id"    "to_station_name"
## [10] "usertype"         "gender"           "birthyear"
```

```
colnames(Q3_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"
## [4] "bikeid"           "tripduration"     "from_station_id"
## [7] "from_station_name" "to_station_id"    "to_station_name"
## [10] "usertype"         "gender"           "birthyear"
```

```
colnames(Q4_2019)
```

```
## [1] "trip_id"          "start_time"       "end_time"
## [4] "bikeid"           "tripduration"     "from_station_id"
## [7] "from_station_name" "to_station_id"    "to_station_name"
## [10] "usertype"         "gender"           "birthyear"
```

Combine the quarterly data into one dataset for the entire year

```
all_2019 <- bind_rows(Q1_2019, Q2_2019, Q3_2019, Q4_2019)
```

Use the summary() function to visually inspect the joined dataset

```
summary(all_2019)
```

```
##      trip_id          start_time          end_time
## Min.   :21742443   Min.    :2019-01-01 00:04:37   Min.    :2019-01-01 00:11:07
## 1st Qu.:22873787   1st Qu.:2019-05-29 15:49:26   1st Qu.:2019-05-29 16:09:28
## Median :23962320   Median :2019-07-25 17:50:54   Median :2019-07-25 18:12:23
## Mean   :23915629   Mean    :2019-07-19 21:47:37   Mean    :2019-07-19 22:11:47
## 3rd Qu.:24963703   3rd Qu.:2019-09-15 06:48:05   3rd Qu.:2019-09-15 08:30:13
## Max.   :25962904   Max.    :2019-12-31 23:57:17   Max.    :2020-01-21 13:54:35
```

```
##
##      bikeid      tripduration      from_station_id from_station_name
## Min.   : 1      Min.   : 61      Min.   : 1.0      Length:3818004
## 1st Qu.:1727    1st Qu.: 411    1st Qu.: 77.0     Class :character
## Median :3451    Median : 709    Median :174.0     Mode  :character
## Mean   :3380    Mean   : 1450    Mean   :201.7
## 3rd Qu.:5046    3rd Qu.: 1283    3rd Qu.:289.0
## Max.   :6946    Max.   :10628400  Max.   :673.0
##
## to_station_id  to_station_name      usertype      gender
## Min.   : 1.0      Length:3818004      Length:3818004      Length:3818004
## 1st Qu.: 77.0     Class :character      Class :character      Class :character
## Median :174.0     Mode  :character      Mode  :character      Mode  :character
## Mean   :202.6
## 3rd Qu.:291.0
## Max.   :673.0
##
##      birthyear
## Min.   :1759
## 1st Qu.:1979
## Median :1987
## Mean   :1984
## 3rd Qu.:1992
## Max.   :2014
## NA's   :538751
```

```
dim(all_2019)
```

```
## [1] 3818004      12
```

```
all_2019 <- all_2019 %>%
  mutate(usertype = recode(usertype
                           , "Subscriber" = "member"
                           , "Customer" = "casual"))
```

Use graphics to check the contents in the usertype column

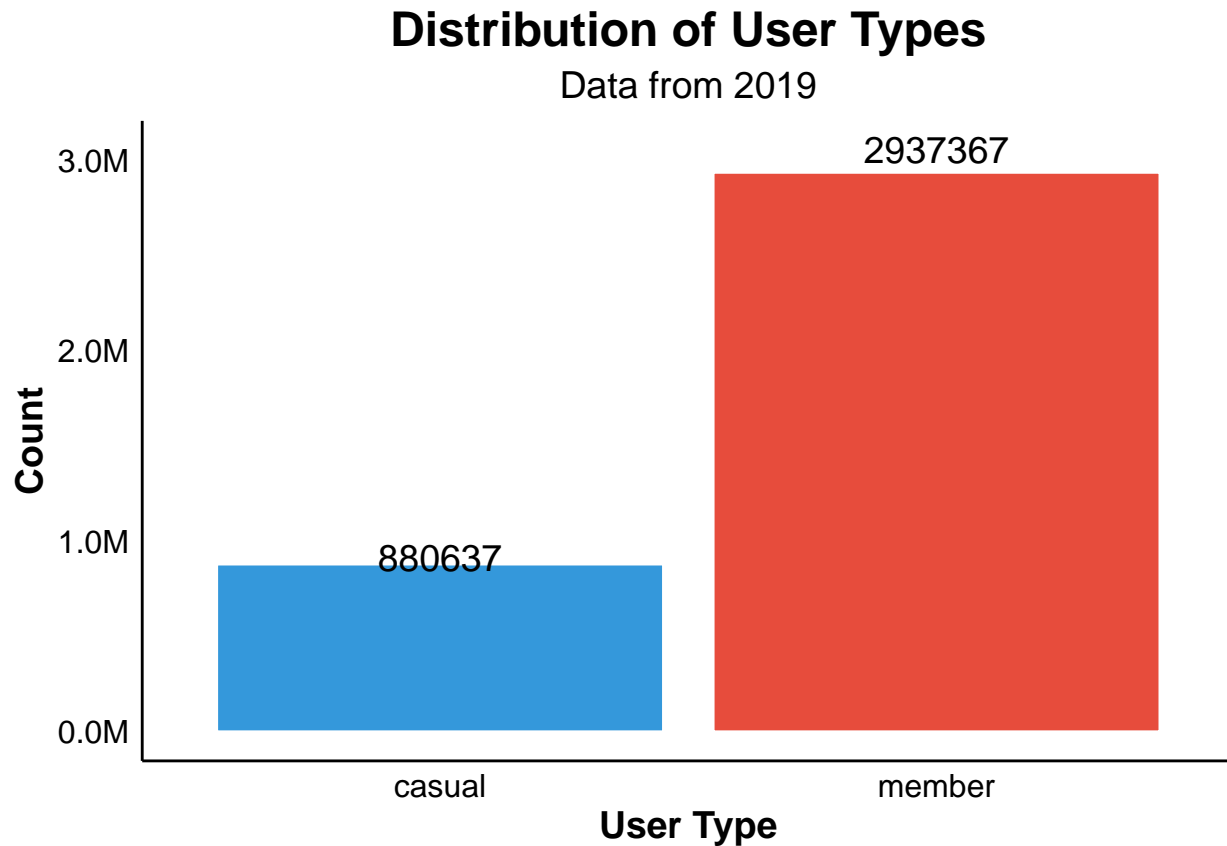
```
ggplot(all_2019, aes(x = usertype, fill = usertype)) +
  geom_bar(color = "white", position = "stack", show.legend = FALSE) +
  geom_text(stat = "count", aes(label = stat(count)),
            position = position_stack(vjust = 1.04), size = 5) +
  labs(title = "Distribution of User Types",
       subtitle = "Data from 2019",
       x = "User Type",
       y = "Count") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
```



```

text = element_text(size = 12),
plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
plot.subtitle = element_text(hjust = 0.5, size = 14),
axis.title = element_text(size = 14, face = "bold"),
axis.text = element_text(size = 12, color = "black")) +
scale_fill_manual(values = c("#3498db", "#e74c3c")) +
scale_y_continuous(labels = scales::number_format(scale = 1e-6, suffix = "M"))

```



```

df <- all_2019 %>%
  group_by(usertype) %>%
  count() %>%
  ungroup() %>%
  mutate(perc = `n` / sum(`n`)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc))

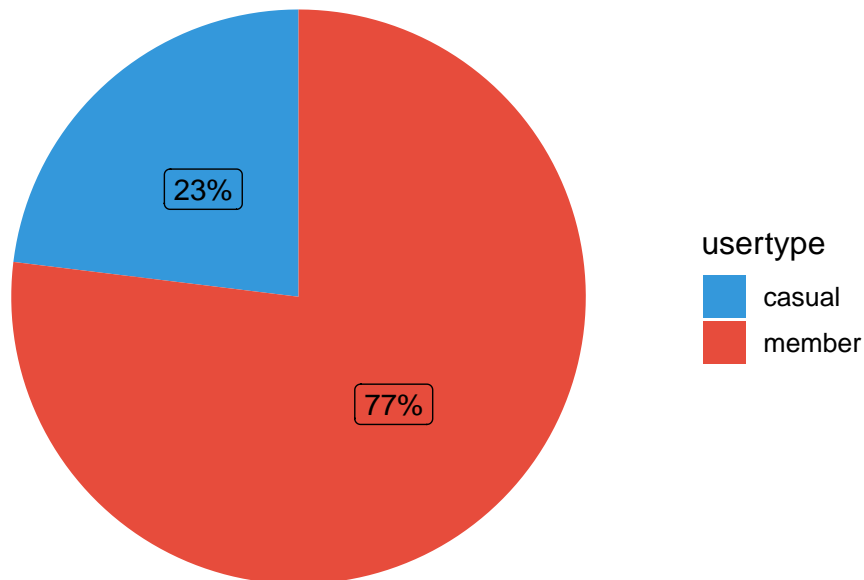
ggplot(df, aes(x = "", y = perc, fill = usertype)) +
  geom_col() +
  geom_label(aes(label = labels),
             position = position_stack(vjust = 0.5),
             show.legend = FALSE) +
  labs(title = "Distribution of User Types",
       subtitle = "Data from 2019",
       x = "",
       y = "") +

```

```
coord_polar(theta = "y")+
theme_minimal() +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank(),
      axis.line = element_blank(),
      text = element_text(size = 12),
      plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
      plot.subtitle = element_text(hjust = 0.5, size = 14),
      axis.title = element_blank(),
      axis.text = element_blank()) +
scale_fill_manual(values = c("#3498db", "#e74c3c"))
```

Distribution of User Types

Data from 2019



Add columns that list the date, month, day, and year of each ride

This will allow us to aggregate ride data for each month, day, or year

```
all_2019$date <- as.Date(all_2019$start_time) #The default format is yyyy-mm-dd
all_2019$month <- format(as.Date(all_2019$date), "%m")
all_2019$day <- format(as.Date(all_2019$date), "%d")
all_2019$year <- format(as.Date(all_2019$date), "%Y")
all_2019$day_of_week <- format(as.Date(all_2019$date), "%A")
```

Descriptive analysis on ride_length (all figures in seconds)

```
summary(all_2019$tripduration)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##       61      411      709     1450     1283 10628400
```

Compare members and casual users

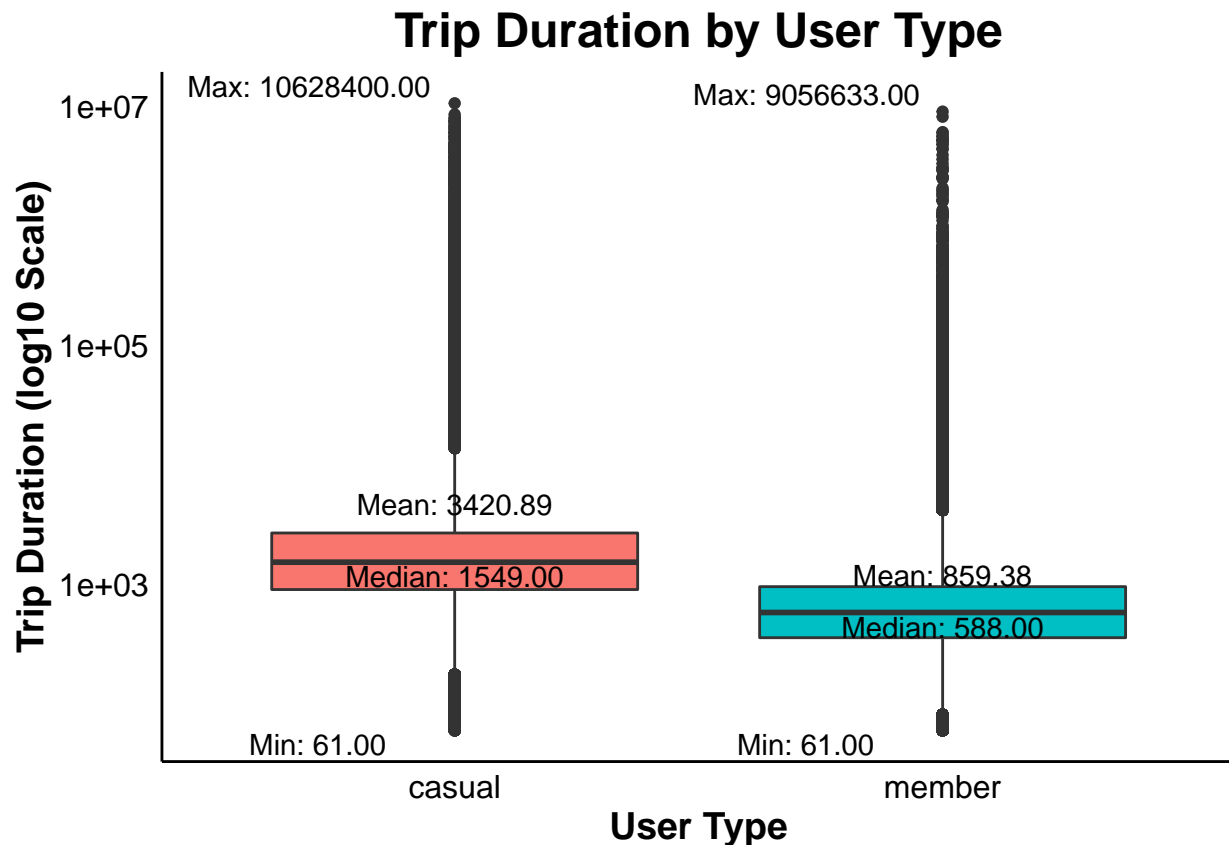
```
summary_stats <- all_2019 %>%
  group_by(usertype) %>%
  summarize(mean_duration = mean(tripduration),
            median_duration = median(tripduration),
            max_duration = max(tripduration),
            min_duration = min(tripduration))
```

```
print(summary_stats)
```

```
## # A tibble: 2 x 5
##   usertype mean_duration median_duration max_duration min_duration
##   <chr>      <dbl>         <dbl>         <dbl>         <dbl>
## 1 casual      3421.           1549       10628400         61
## 2 member       859.           588        9056633         61
```

```
ggplot(all_2019, aes(x = usertype, y = tripduration, fill = usertype)) +
  geom_boxplot(show.legend = FALSE) +
  geom_text(data = summary_stats,
            aes(x = usertype, y = mean_duration, label = sprintf("Mean: %.2f", mean_duration)),
            vjust = -0.3, hjust = 0.5, color = "black") +
  geom_text(data = summary_stats,
            aes(x = usertype, y = median_duration, label = sprintf("Median: %.2f", median_duration)),
            vjust = 1.2, hjust = 0.5, color = "black") +
  geom_text(data = summary_stats,
            aes(x = usertype, y = max_duration, label = sprintf("Max: %.2f", max_duration)),
            vjust = -0.3, hjust = 1.1, color = "black") +
  geom_text(data = summary_stats,
            aes(x = usertype, y = min_duration, label = sprintf("Min: %.2f", min_duration)),
            vjust = 1.2, hjust = 1.5, color = "black") +
  labs(title = "Trip Duration by User Type",
       x = "User Type",
       y = "Trip Duration (log10 Scale)") +
  scale_y_log10() +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
```

```
text = element_text(size = 12),
plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
axis.title = element_text(size = 14, face = "bold"),
axis.text = element_text(size = 12, color = "black"))
```



Looking at ridership by month and gender distribution

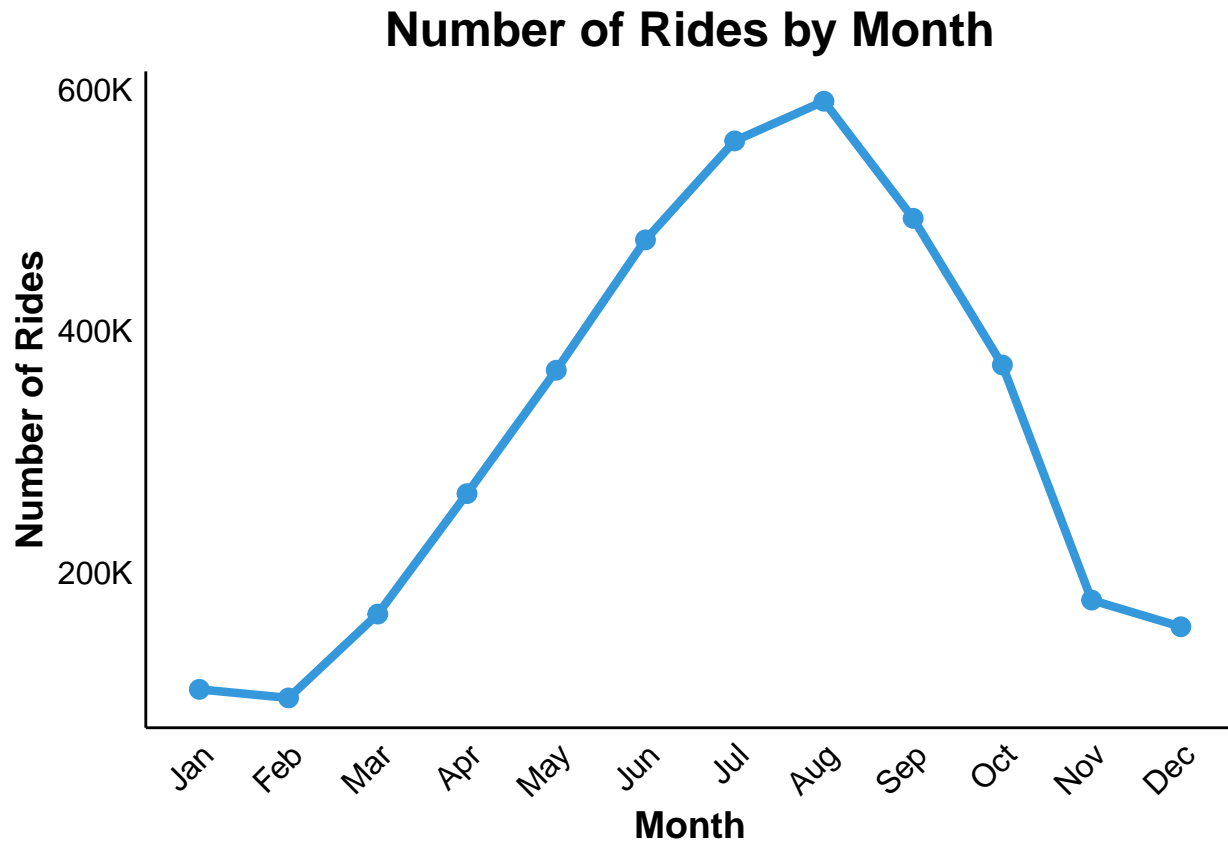
```
rides_by_month <- all_2019 %>%
  group_by(month) %>%
  summarise(number_of_rides = n())

ggplot(rides_by_month, aes(x = month, y = number_of_rides, group = 1)) +
  geom_line(color = "#3498db", size = 1.5) +
  geom_point(color = "#3498db", size = 3) +
  labs(title = "Number of Rides by Month",
       x = "Month",
       y = "Number of Rides") +
  scale_x_discrete(labels = month.abb) +
  theme_minimal() +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
        text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
```

```

plot.subtitle = element_text(hjust = 0.5, size = 14),
axis.title = element_text(size = 14, face = "bold"),
axis.text = element_text(size = 12, color = "black")) +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
scale_y_continuous(labels = scales::number_format(scale = 1e-3, suffix = "K"))

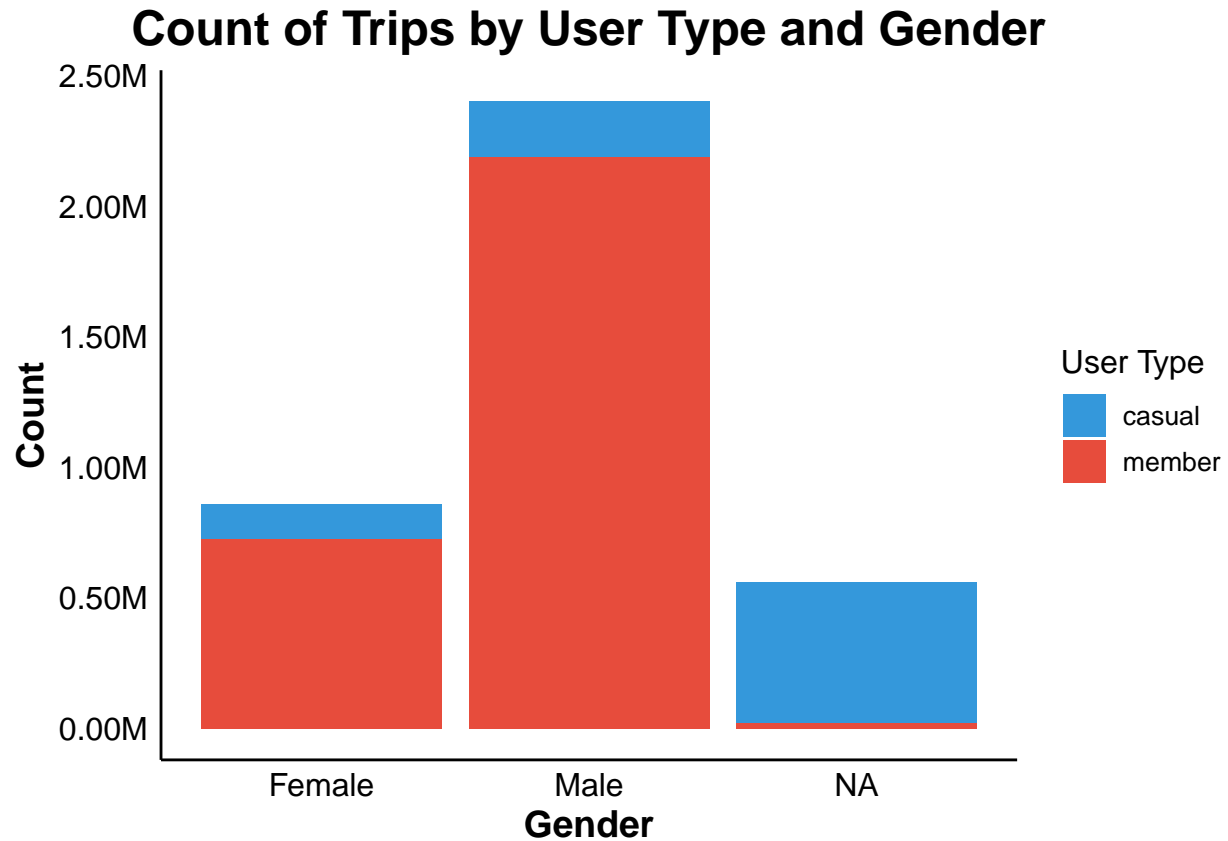
```



```

ggplot(all_2019, aes(x = gender, fill = usertype)) +
  geom_bar() +
  labs(title = "Count of Trips by User Type and Gender", x = "Gender", y = "Count", fill = "User Type")
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
        text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
        axis.title = element_text(size = 14, face = "bold"),
        axis.text = element_text(size = 12, color = "black"),
        legend.position = "right") +
  scale_fill_manual(values = c("#3498db", "#e74c3c")) +
  scale_y_continuous(labels = scales::number_format(scale = 1e-6, suffix = "M"))

```



See the average ride time by each day for members vs casual users

```
aggregate(all_2019$tripduration ~ all_2019$usertype + all_2019$day_of_week, FUN = mean)
```

| ## | all_2019\$usertype | all_2019\$day_of_week | all_2019\$tripduration |
|-------|--------------------|-----------------------|------------------------|
| ## 1 | casual | Friday | 3610.3514 |
| ## 2 | member | Friday | 833.5801 |
| ## 3 | casual | Monday | 3269.7334 |
| ## 4 | member | Monday | 854.6510 |
| ## 5 | casual | Saturday | 3243.5687 |
| ## 6 | member | Saturday | 977.9140 |
| ## 7 | casual | Sunday | 3370.8448 |
| ## 8 | member | Sunday | 923.8375 |
| ## 9 | casual | Thursday | 3596.7864 |
| ## 10 | member | Thursday | 826.5039 |
| ## 11 | casual | Tuesday | 3444.5548 |
| ## 12 | member | Tuesday | 848.8744 |
| ## 13 | casual | Wednesday | 3619.9636 |
| ## 14 | member | Wednesday | 828.3104 |

Ordering days of the week

```
all_2019$day_of_week <- ordered(all_2019$day_of_week, levels=c( "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
aggregate(all_2019$tripduration ~ all_2019$usertype + all_2019$day_of_week, FUN = mean)
```

```
##   all_2019$usertype all_2019$day_of_week all_2019$tripduration
## 1      casual      Monday      3269.7334
## 2      member      Monday      854.6510
## 3      casual      Tuesday     3444.5548
## 4      member      Tuesday      848.8744
## 5      casual     Wednesday     3619.9636
## 6      member     Wednesday      828.3104
## 7      casual      Thursday     3596.7864
## 8      member      Thursday      826.5039
## 9      casual      Friday      3610.3514
## 10     member      Friday      833.5801
## 11     casual      Saturday     3243.5687
## 12     member      Saturday      977.9140
## 13     casual      Sunday      3370.8448
## 14     member      Sunday      923.8375
```

Analyze ridership data by type and weekday

```
all_2019 %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n(), average_duration = mean(tripduration)) %>%
  arrange(usertype, weekday)
```

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

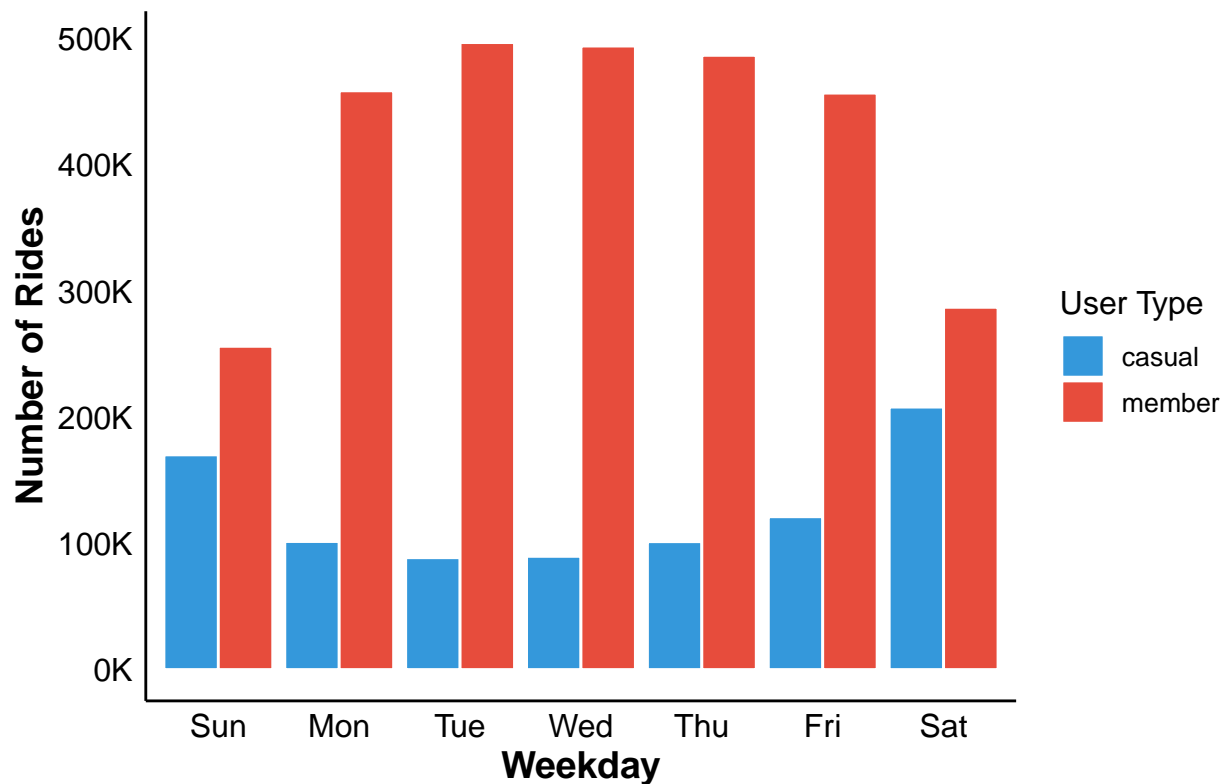
```
## # A tibble: 14 x 4
## # Groups:   usertype [2]
##   usertype weekday number_of_rides average_duration
##   <chr>    <ord>      <int>          <dbl>
## 1 casual  Sun         170179         3371.
## 2 casual  Mon         101489         3270.
## 3 casual  Tue          88655         3445.
## 4 casual  Wed          89745         3620.
## 5 casual  Thu         101372         3597.
## 6 casual  Fri         121141         3610.
## 7 casual  Sat         208056         3244.
## 8 member  Sun         256241          924.
## 9 member  Mon         458780          855.
## 10 member Tue         497025          849.
## 11 member Wed         494277          828.
## 12 member Thu         486915          827.
## 13 member Fri         456966          834.
## 14 member Sat         287163          978.
```

Visualize the number of rides by rider type

```
all_2019 %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n(), average_duration = mean(tripduration)) %>%
  arrange(usertype, weekday) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = usertype)) +
  geom_col(position = "dodge", color = "white", size = 0.5) +
  labs(title = "Number of Rides by User Type and Weekday",
       x = "Weekday",
       y = "Number of Rides",
       fill = "User Type") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
        text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
        axis.title = element_text(size = 14, face = "bold"),
        axis.text = element_text(size = 12, color = "black"),
        legend.position = "right") +
  scale_fill_manual(values = c("#3498db", "#e74c3c")) +
  scale_y_continuous(labels = scales::number_format(scale = 1e-3, suffix = "K"))
```

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

Number of Rides by User Type and Weekday

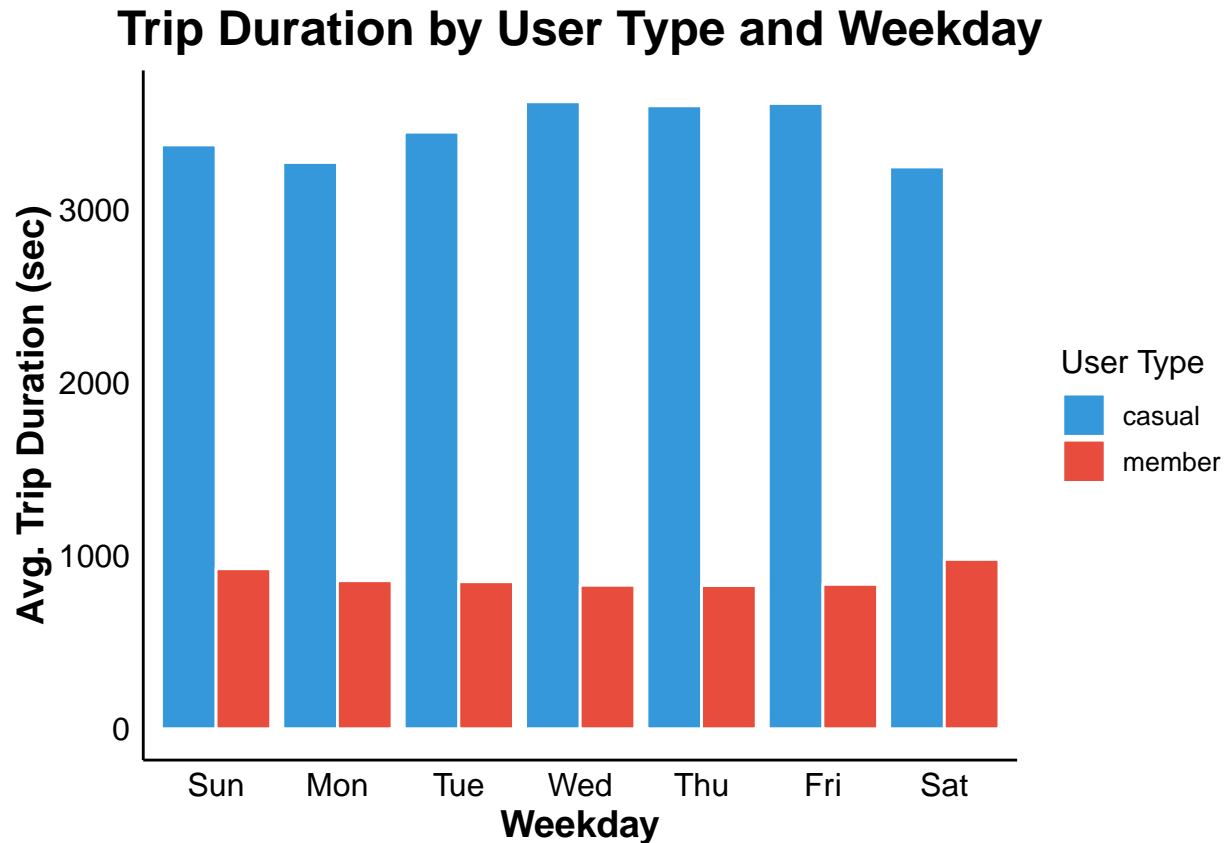


Let's create a visualization for average duration

```
all_2019 %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n(), average_duration = mean(tripduration)) %>%
  arrange(usertype, weekday) %>%
  ggplot(aes(x = weekday, y = average_duration, fill = usertype)) +
  geom_col(position = "dodge", color = "white", size = 0.5) +
  labs(title = "Trip Duration by User Type and Weekday",
       x = "Weekday",
       y = "Avg. Trip Duration (sec)",
       fill = "User Type") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(color = "black"),
        text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
        axis.title = element_text(size = 14, face = "bold"),
        axis.text = element_text(size = 12, color = "black"),
        legend.position = "right") +
```

```
scale_fill_manual(values = c("#3498db", "#e74c3c"))
```

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



Do statistical test to see if the difference in tripduration is significance

```
user_type1_data <- all_2019$tripduration[all_2019$usertype == "member"]
user_type2_data <- all_2019$tripduration[all_2019$usertype == "casual"]

t_test_result <- t.test(user_type1_data, user_type2_data)

print(t_test_result)
```

```
##
## Welch Two Sample t-test
##
## data: user_type1_data and user_type2_data
## t = -42.742, df = 920075, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
## -2678.962 -2444.043
## sample estimates:
## mean of x mean of y
## 859.3833 3420.8857
```

Final observation

Based our analysis we see that while casual customers use bikes for a longer duration, subscribing members average more daily number of rides, especially on weekdays.

Some suggestions:

- Highlight perks such as unlimited rides, exclusive discounts, and priority access to bikes.
- Emphasize the higher daily number of rides that members enjoy, especially on weekdays, showcasing the value of membership for frequent users.
- Offer special promotions or discounted membership rates for casual riders who have a history of longer-duration rides. This can act as an incentive for them to subscribe, considering their extended usage pattern.