How Does a Bike-Share Navigate Speedy Success

Okyere Adubofuor

2026-01-05

Install and load the tools for the project by install the tidyverse package

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.2
## v purrr     1.2.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(readr)
```

Implement the appropriate conflicts resolution

```r
library(conflicted)

conflicts_prefer(dplyr::filter)
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

#———————————————————————- # STEP 1: GATHER THE DATA FOR THE PROJECT AND ASSIGN TO VARIABLES #———————————————————————- ## Upload divvy data

```r
q1_19 <- read_csv("Data/Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 41948 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): start_time, end_time, from_station_name, to_station_name, usertype,...
## dbl (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num (1): tripduration
##
## 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.
```

```
q1_20 <- read_csv("Data/Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 166208 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, 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.
```

STEP 2: COMPARE AND MATCH COLUMNS NAMES AND THEN COMBINE THE DATA

---

have a general view of the data

```
View(q1_19)
View(q1_20)
```

Inspect the structure of the data for column names and their type

```
str(q1_19)
```

```
## spc_tbl_ [41,948 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:41948] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : chr [1:41948] "1/1/2019 0:04" "1/1/2019 0:08" "1/1/2019 0:13" "1/1/2019 0:13"
##  $ end_time         : chr [1:41948] "1/1/2019 0:11" "1/1/2019 0:15" "1/1/2019 0:27" "1/1/2019 0:43"
##  $ bikeid           : num [1:41948] 2167 4386 1524 252 1170 ...
##  $ tripduration     : num [1:41948] 390 441 829 1783 364 ...
##  $ from_station_id  : num [1:41948] 199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr [1:41948] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave & 
##  $ to_station_id    : num [1:41948] 84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr [1:41948] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "We
##  $ usertype         : chr [1:41948] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr [1:41948] "Male" "Female" "Female" "Male" ...
##  $ birthyear        : num [1:41948] 1989 1990 1994 1993 1994 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_character(),
##   ..   end_time = col_character(),
##   ..   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(q1_20)
```

```
## spc_tbl_ [166,208 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:166208] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A38
##  $ rideable_type     : chr [1:166208] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr [1:166208] "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:2
##  $ ended_at          : chr [1:166208] "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:1
##  $ start_station_name: chr [1:166208] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway
##  $ start_station_id  : num [1:166208] 239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr [1:166208] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilto
##  $ end_station_id    : num [1:166208] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num [1:166208] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:166208] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:166208] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:166208] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:166208] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

check the column names

```
colnames(q1_19)
```

```
##  [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(q1_20)
```

```
##  [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"
```

Match column names by changing some to the chosen names

change some column names in q1_19 to the appropriate names in q1_20

```
q1_19 <- q1_19 %>%
  rename(
    start_station_id = from_station_id,
    start_station_name = from_station_name,
    end_station_id = to_station_id,
    end_station_name = to_station_name
  )
```

change some column names in q1_20 to the appropriate names in q1_19

```
q1_20 <- q1_20 %>%
  rename(
    trip_id = ride_id,
    start_time = started_at,
    end_time = ended_at,
    bike_id = rideable_type,
    usertype = member_casual
  )
```

Inspect the data frames after the changes

```
View(q1_19)
View(q1_20)
```

Create "tripduration" column in q1_20 to match the column in q1_19

Load the lubridate to use the datetime values directly

```
library(lubridate)
```

calculate the values for the "tripduration" column

```
q1_20 <- mutate(q1_20,
    end_time = ymd_hms(end_time),
    start_time = ymd_hms(start_time),
    tripduration = as.numeric(difftime(end_time, start_time, units = "secs"))
)

View(q1_20)
```

Set trip_id in q1_19 to character to match the trip_id in q1_20

```
q1_19 <- mutate(q1_19, trip_id = as.character(trip_id))
```

Inspect the dataframes for discrepancies in column names and types

```
str(q1_19)
```

```
## tibble [41,948 x 12] (S3: tbl_df/tbl/data.frame)
##  $ trip_id          : chr [1:41948] "21742443" "21742444" "21742445" "21742446" ...
##  $ start_time       : chr [1:41948] "1/1/2019 0:04" "1/1/2019 0:08" "1/1/2019 0:13" "1/1/2019 0:13"
##  $ end_time         : chr [1:41948] "1/1/2019 0:11" "1/1/2019 0:15" "1/1/2019 0:27" "1/1/2019 0:43"
##  $ bikeid           : num [1:41948] 2167 4386 1524 252 1170 ...
##  $ tripduration     : num [1:41948] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:41948] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:41948] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:41948] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:41948] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ usertype         : chr [1:41948] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr [1:41948] "Male" "Female" "Female" "Male" ...
##  $ birthyear        : num [1:41948] 1989 1990 1994 1993 1994 ...
```

```
str(q1_20)
```

```
## tibble [166,208 x 14] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:166208] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A3
##  $ bike_id           : chr [1:166208] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ start_time        : POSIXct[1:166208], format: "2020-01-21 20:06:59" "2020-01-30 14:22:39" ...
##  $ end_time          : POSIXct[1:166208], format: "2020-01-21 20:14:30" "2020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:166208] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway
##  $ start_station_id  : num [1:166208] 239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr [1:166208] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilte
##  $ end_station_id    : num [1:166208] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num [1:166208] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:166208] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:166208] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:166208] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ usertype          : chr [1:166208] "member" "member" "member" "member" ...
##  $ tripduration      : num [1:166208] 451 223 171 529 332 289 289 297 295 203 ...
```

Select the relevant columns for each dataframe before combining them

```
q1_19_select <- q1_19 %>%
  select(trip_id, usertype, tripduration, start_station_id, start_station_name,
         end_station_id, end_station_name, start_time)

q1_20_select <- q1_20 %>%
  select(trip_id, usertype, tripduration, start_station_id, start_station_name,
         end_station_id, end_station_name, start_time) %>%
  mutate(start_time = as.character(start_time))
```

Inspect the dataframes for discrepancies in column names and types

```
str(q1_19_select)
```

```
## tibble [41,948 x 8] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:41948] "21742443" "21742444" "21742445" "21742446" ...
##  $ usertype          : chr [1:41948] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ tripduration      : num [1:41948] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:41948] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:41948] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:41948] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:41948] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "We
##  $ start_time        : chr [1:41948] "1/1/2019 0:04" "1/1/2019 0:08" "1/1/2019 0:13" "1/1/2019 0:13"
```

```
View(q1_19_select)
str(q1_20_select)
```

```
## tibble [166,208 x 8] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:166208] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A38
##  $ usertype          : chr [1:166208] "member" "member" "member" "member" ...
##  $ tripduration      : num [1:166208] 451 223 171 529 332 289 289 297 295 203 ...
##  $ start_station_id  : num [1:166208] 239 234 296 51 66 212 96 96 212 38 ...
##  $ start_station_name: chr [1:166208] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway
##  $ end_station_id    : num [1:166208] 326 318 117 24 212 96 212 212 96 100 ...
##  $ end_station_name  : chr [1:166208] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilte
##  $ start_time        : chr [1:166208] "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:2
```

```
View(q1_20_select)
```

Combine the two dataframes into one dataframes

```
all_trips <- bind_rows(q1_19_select, q1_20_select)
```

Inspect the combined dataframe as you prepare for Analysis

```
str(all_trips)
```

```
## tibble [208,156 x 8] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:208156] "21742443" "21742444" "21742445" "21742446" ...
##  $ usertype          : chr [1:208156] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ tripduration      : num [1:208156] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:208156] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:208156] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:208156] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:208156] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ start_time        : chr [1:208156] "1/1/2019 0:04" "1/1/2019 0:08" "1/1/2019 0:13" "1/1/2019 0:13
```

STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS

Inspect the combined dataframe that has been created

```
colnames(all_trips) # Lists all column names
```

```
## [1] "trip_id"            "usertype"           "tripduration"
## [4] "start_station_id"   "start_station_name" "end_station_id"
## [7] "end_station_name"   "start_time"
```

```
nrow(all_trips) # Displays the number of rows
```

```
## [1] 208156
```

```
dim(all_trips) # Shows the dimension of the dataframe
```

```
## [1] 208156      8
```

```r
head(all_trips) # Displays first 6 rows of the dataframe
```

```
## # A tibble: 6 x 8
##   trip_id  usertype   tripduration start_station_id start_station_name
##   <chr>    <chr>             <dbl>            <dbl> <chr>
## 1 21742443 Subscriber          390              199 Wabash Ave & Grand Ave
## 2 21742444 Subscriber          441               44 State St & Randolph St
## 3 21742445 Subscriber          829               15 Racine Ave & 18th St
## 4 21742446 Subscriber         1783              123 California Ave & Milwaukee ~
## 5 21742447 Subscriber          364              173 Mies van der Rohe Way & Chi~
## 6 21742448 Subscriber          216               98 LaSalle St & Washington St
## # i 3 more variables: end_station_id <dbl>, end_station_name <chr>,
## #   start_time <chr>
```

```r
str(all_trips) # Shows the column names and data types
```

```
## tibble [208,156 x 8] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:208156] "21742443" "21742444" "21742445" "21742446" ...
##  $ usertype          : chr [1:208156] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ tripduration      : num [1:208156] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:208156] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:208156] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:208156] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:208156] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "
##  $ start_time        : chr [1:208156] "1/1/2019 0:04" "1/1/2019 0:08" "1/1/2019 0:13" "1/1/2019 0:13
```

```r
summary(all_trips) #Statistical summary of data
```

```
##    trip_id            usertype          tripduration     start_station_id
##  Length:208156      Length:208156      Min.   :     -1   Min.   :  2.0
##  Class :character   Class :character   1st Qu.:    321   1st Qu.: 77.0
##  Mode  :character   Mode  :character   Median :    517   Median :174.0
##                                        Mean   :   1188   Mean   :201.5
##                                        3rd Qu.:    848   3rd Qu.:289.0
##                                        Max.   :9387024   Max.   :675.0
##  start_station_name end_station_id  end_station_name    start_time
##  Length:208156      Min.   :  2.0   Length:208156      Length:208156
##  Class :character   1st Qu.: 77.0   Class :character   Class :character
##  Mode  :character   Median :172.0   Mode  :character   Mode  :character
##                     Mean   :201.2
##                     3rd Qu.:289.0
##                     Max.   :675.0
```

The usertype columnn shows 4 values representing only 2 categories, therefore they must be made to conform to only 2

This situation arose because the 2 dataframes had different labels for the 2 categories of users

Inspect the data to know the number of values for each label

```r
table(all_trips$usertype)
```

```
##
##     casual   Customer     member Subscriber
##       9829       3408     156379      38540
```

Reassign the values to the desired values

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

List the number of observations for each category to see that the correct changes were made

```r
table(all_trips$usertype)
```

```
##
## casual member
##  13237 194919
```

Extract day, month, and year from the start__time column and create columns for them

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

First convert the start__time column to date format

Since the date are in different datetime formats, parse them using parse__date__time before casting to type date

```r
all_trips$start_time <- parse_date_time(all_trips$start_time,
                                        orders = c("dmy HM", "ymd HMS"))
all_trips$start_time <- as.Date(all_trips$start_time)
```

Extract year, month, and day from start__time and create columns for them

```r
all_trips <- all_trips %>%
  mutate(
    year = year(start_time),
    month = month(start_time, label = TRUE, abbr = FALSE),
    day = day(start_time),
    day_of_week = wday(start_time, label = TRUE, abbr = FALSE, week_start = 1)
  )
```

Remove bad data such as when tripduration is negative or when the ride was initiated by the company

Sometimes the bikes are undocked by the company to check quality issues.

This is done at the HQ so remove all data with "HQ QR" as start_station_name

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR"
                            | all_trips$tripduration < 0),]
```

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

Descriptive analysis on tripduration.

```
tripduration_summary <- summary(all_trips_v2$tripduration)
```

Compare members and casual users

```
members_vs_casual_summary <- aggregate(all_trips_v2$tripduration ~ all_trips_v2$usertype, FUN = summary
```

See the average tripduration for members and casual users on each day of the week

```
members_vs_casual_dotw <- aggregate(all_trips_v2$tripduration ~ all_trips_v2$usertype + all_trips_v2$da
```

Fix the other of values in the day_of_week column

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels = c("Sunday", "Monday", "Tuesday",
```

Compare the number of rides and average tripduration on each day for members and casual users

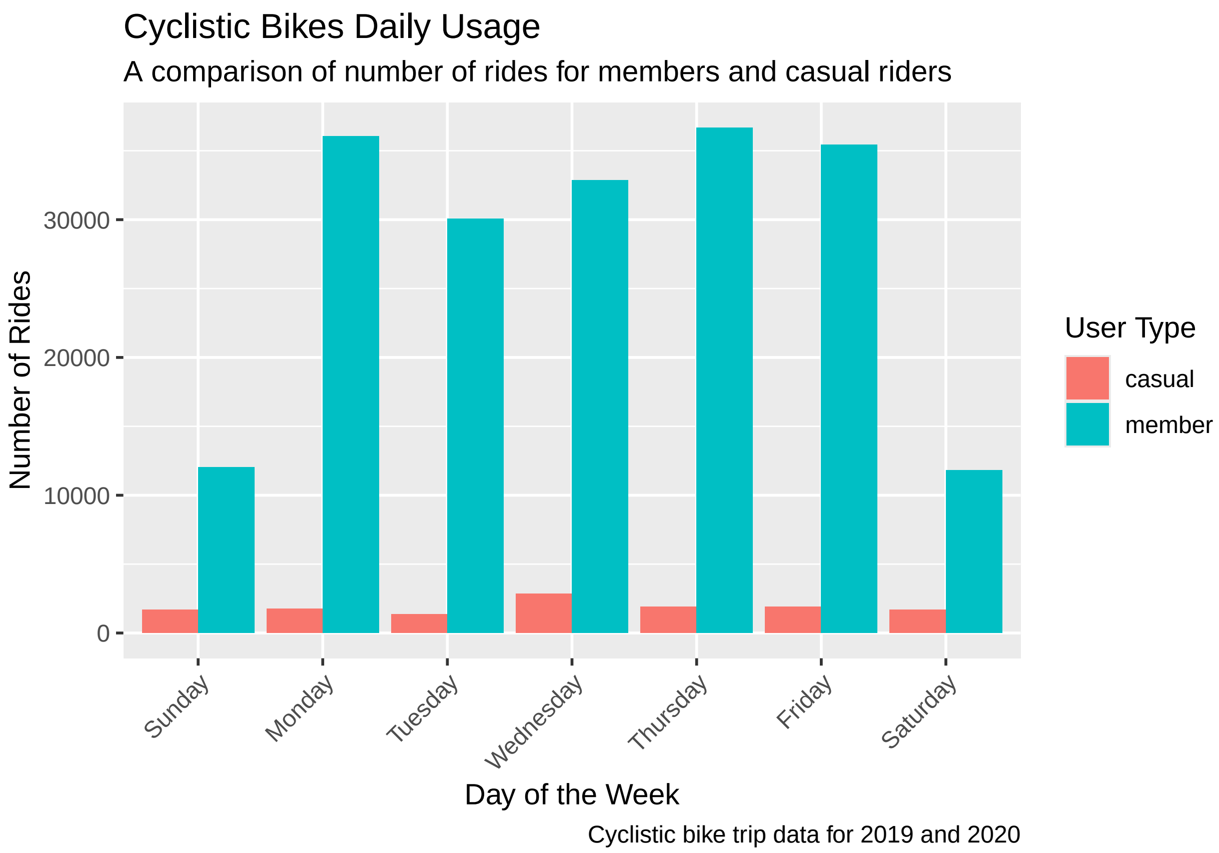
```
members_vs_casual_num_of_rides_vs_duration <- all_trips_v2 %>%
  group_by(usertype, day_of_week) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(tripduration)) %>%
  arrange(usertype, day_of_week)
```

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

Let’s visualize the number of rides by rider type

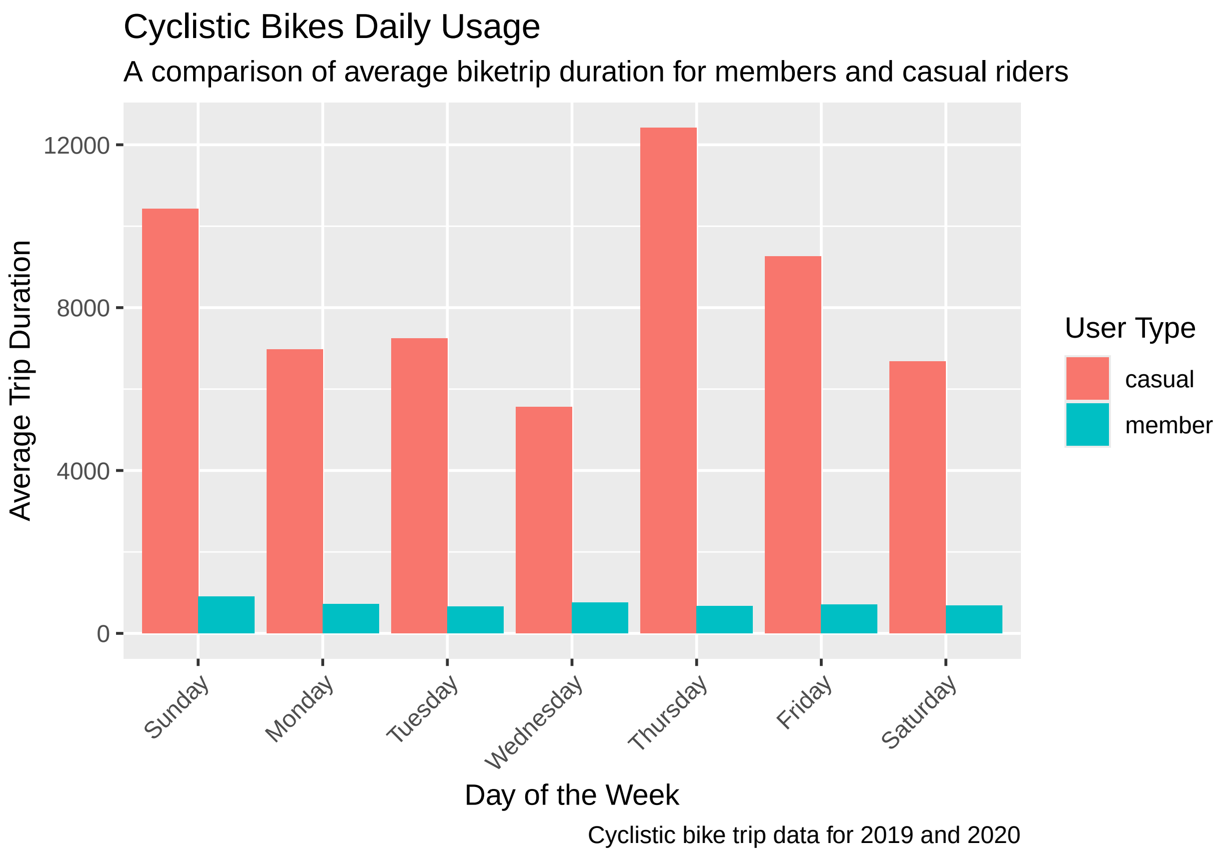
```
all_trips_v2 %>%
  group_by(usertype, day_of_week) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(tripduration)) %>%
  arrange(usertype, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(
    x = "Day of the Week",
    y = "Number of Rides",
    fill = "User Type"
  ) +
  labs(
    title = "Cyclistic Bikes Daily Usage",
    subtitle = "A comparison of number of rides for members and casual riders",
    caption = "Cyclistic bike trip data for 2019 and 2020") +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1)
)
```

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



```
## Let's create a visualization for average tripduration
all_trips_v2 %>%
  group_by(usertype, day_of_week) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(tripduration)) %>%
  arrange(usertype, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(
    x = "Day of the Week",
    y = "Average Trip Duration",
    fill = "User Type"
  ) +
  labs(
    title = "Cyclistic Bikes Daily Usage",
    subtitle = "A comparison of average biketrip duration for members and casual riders",
    caption = "Cyclistic bike trip data for 2019 and 2020") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

```
## ‘summarise()‘ has grouped output by ’usertype’. You can override using the
## ‘.groups‘ argument.
```



Cyclistic Bikes Daily Usage

A comparison of average biketrip duration for members and casual riders

Cyclistic bike trip data for 2019 and 2020

STEP 5: EXPORT SUMMARY FILE FOR FURTHER ANALYSIS

```
write.csv(all_trips_v2, "C:/Users/Okyere Adubofuor/Documents/Data_Analysis/Capstone/csv from R/all_trips

write.csv(members_vs_casual_num_of_rides_vs_duration, "C:/Users/Okyere Adubofuor/Documents/Data_Analysis

write.csv(members_vs_casual_summary, "C:/Users/Okyere Adubofuor/Documents/Data_Analysis/Capstone/csv fr
```